

Table of Contents

- 1. Introduction**
- 2. Manuel : Introduction**
- 3. Manuel : Qu'est-ce que FreeCAD ?**
- 4. Manuel : Installation**
 - 4.1 Installation sur windows
 - 4.2 Installation sur linux
 - 4.3 Installation sur mac os
 - 4.4 Désinstallation
 - 4.5 Définir les préférences de base
 - 4.6 Installation de contenu supplémentaire
- 5. Manuel : L'interface FreeCad**
 - 5.1 Les ateliers
 - 5.2 L'interface
 - 5.3 Personnalisation de l'interface
- 6. Manuel : Navigation dans la vue 3D**
 - 6.1 Un mot sur l'espace 3d
 - 6.2 La vue 3d freecad
 - 6.3 Sélection d'objets
- 7. Manuel : Le document FreeCAD**
- 8. Manuel : Objets paramétriques**
- 9. Manuel : Importer et exporter vers d'autres types de fichiers**
- 10. Manuel : Tous les ateliers en un coup d'œil**
- 11. Manuel : Modélisation traditionnelle, la voie CSG**
- 12. Manuel : Dessin 2D traditionnel**
- 13. Manuel: Modélisation pour la conception de produits**
- 14. Manuel : Préparation des modèles pour l'impression 3D**
 - 14.1 Exportation vers des trancheuses
 - 14.2 Conversion d'objets en mailles
 - 14.3 Utilisation de slic3r
 - 14.4 Utilisation du greffon .28addon.29 cura
 - 14.5 Génération du g-code
- 15. Manuel : Génération de dessins 2D**
- 16. Manuel : Modélisation BIM**
- 17. Manuel : Utilisation de feuilles de calcul**
 - 17.1 Lecture de propriétés
 - 17.2 Ecriture de propriétés
- 18. Manuel : Création d'analyses FEM**
- 19. Manuel : Création de rendus réalistes**
- 20. Manuel : petite introduction à Python**
 - 20.1 Ecrire du code python
 - 20.2 Manipuler des objets freecad

20.3 Vecteurs et emplacements

- 21. Manuel : Création et manipulation de la géométrie**
- 22. Manuel : Création d'objets paramétriques**
- 23. Manuel : Création d'outils d'interface**
- 24. Manuel : La communauté**

Introduction

[FreeCAD](#) is an open-source, free parametric 3D computer-aided design (CAD) modelling tool used to create objects ranging from simple designs to complex projects consisting of extensive assemblies of numerous parts. Due to its open-source nature, FreeCAD can be freely downloaded, distributed, modified, and used for both personal and commercial purposes. The source code is openly published under the [LGPL](#) license. This makes FreeCAD an appealing option not only for relatively inexperienced users, such as hobbyists in the 3D printing world, but also for experienced users looking to experiment with a different parametric tool.

The software was initially released in its infancy in 2002. Since then, thanks to the continuous contributions from a devoted community of developers and users, FreeCAD has grown slowly but steadily. The passionate efforts of numerous contributors have not only improved FreeCAD's capabilities as a CAD tool but have also led to the development of additional functionalities in separate domains closely tied to the base FreeCAD suite. Some notable examples include the FEM workbench, which enables the execution of simple Finite Element Analysis, as well as the BIM workbench which offers Building Information Modeling (BIM) capabilities.

This user manual is being written and updated for version [1.0](#) of FreeCAD. Its purpose is to serve as a guide for individuals who wish to start using FreeCAD for their projects. No extensive prior knowledge of any CAD software is required, and the contents of this manual are organized in a straightforward and comprehensive manner, divided into various application areas. The guide includes instructions regarding the installation of FreeCAD, the use of its most prominent workbenches, and extends to more complex topics such as Python scripting. Efforts have been made to present the information in an organized, step-by-step fashion, complete with images. This structure ensures that both novice CAD users and more experienced individuals can benefit from studying this manual. The contents of this manual are published under the [Creative Commons 4.0](#) license, and can be freely used, downloaded, copied, and modified. The FreeCAD Manual is continuously improved through user experiences and contributions. A [Python script](#) is available to convert the latest version of this manual into PDF or EPUB formats for offline reading.

Another valuable resource is the [FreeCAD wiki](#), where numerous contributors have added useful information covering every aspect of the software. Due to the wealth of information available, navigating and using the wiki can be somewhat challenging and overwhelming, especially for new users. For this reason, it is advised to first study the user manual, which offers a more streamlined and smooth introduction, before delving into the more extensive details available in the wiki. Finally, the [FreeCAD Forums](#) are also very helpful and provide additional support and insights.

Manuel : Introduction

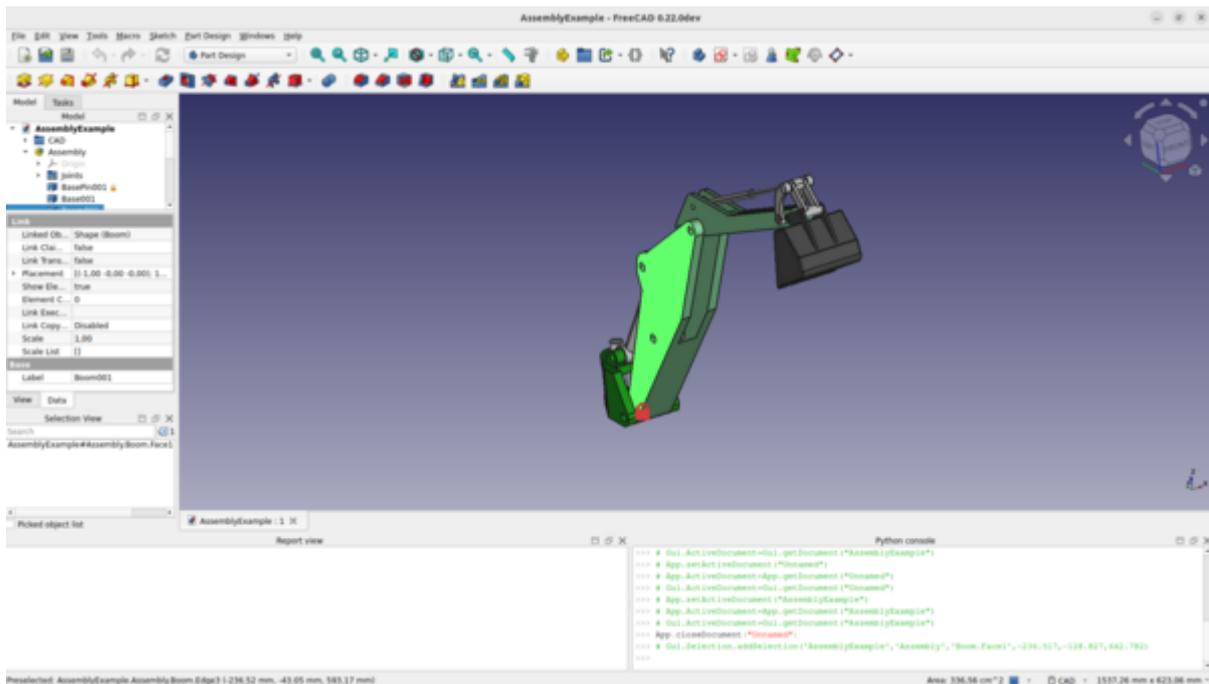
[FreeCAD](#) est un outil de modélisation de conception assistée par ordinateur (CAO) 3D paramétrique, libre et gratuit, utilisé pour créer des objets allant de simples dessins à des projets complexes consistant en de vastes assemblages de nombreuses pièces. En raison de sa nature open-source, FreeCAD peut être librement téléchargé, distribué, modifié et utilisé à des fins personnelles et commerciales. Le code source est publié ouvertement sous la licence [LGPL](#). Cela fait de FreeCAD une option attrayante non seulement pour les utilisateurs relativement inexpérimentés, tels que les amateurs du monde de l'impression 3D, mais aussi pour les utilisateurs expérimentés qui cherchent à expérimenter un outil paramétrique différent.

Le logiciel a été publié pour la première fois en 2002. Depuis, grâce aux contributions continues d'une communauté dévouée de développeurs et d'utilisateurs, FreeCAD s'est développé lentement mais sûrement. Les efforts passionnés de nombreux contributeurs ont non seulement amélioré les capacités de FreeCAD en tant qu'outil de CAO, mais ont également conduit au développement de fonctionnalités supplémentaires dans des domaines distincts étroitement liés à la suite FreeCAD de base. Parmi les exemples notables, on peut citer l'atelier FEM, qui permet l'exécution d'analyses simples par éléments finis, ainsi que l'atelier BIM, qui offre des fonctionnalités de modélisation des informations sur le bâtiment (BIM).

Ce manuel d'utilisation est rédigé et mis à jour pour la version [1.0](#) de FreeCAD. Son but est de servir de guide aux personnes qui souhaitent commencer à utiliser FreeCAD pour leurs projets. Aucune connaissance préalable approfondie d'un logiciel de CAO n'est requise, et le contenu de ce manuel est organisé de manière simple et complète, divisé en différents domaines d'application. Le guide comprend des instructions concernant l'installation de FreeCAD, l'utilisation de ses principaux postes de travail, et s'étend à des sujets plus complexes tels que les scripts Python. Des efforts ont été faits pour présenter les informations de manière organisée, étape par étape, avec des images. Cette structure permet aux utilisateurs novices de CAO comme aux personnes plus expérimentées de tirer profit de l'étude de ce manuel. Le contenu de ce manuel est publié sous la licence [Creative Commons 4.0](#) et peut être librement utilisé, téléchargé, copié et modifié. Le manuel FreeCAD est continuellement amélioré grâce aux expériences et aux contributions des utilisateurs. Un [script en Python](#) est disponible pour convertir la dernière version de ce manuel au format PDF ou EPUB pour une lecture hors ligne.

Une autre ressource précieuse est le [wiki de FreeCAD](#) où de nombreux contributeurs ont ajouté des informations utiles couvrant tous les aspects du logiciel. En raison de la richesse des informations disponibles, la navigation et l'utilisation du wiki peuvent être quelque peu difficiles et accablantes, en particulier pour les nouveaux utilisateurs. Pour cette raison, il est conseillé d'étudier d'abord le manuel de l'utilisateur, qui offre une introduction plus simple et plus fluide, avant de se plonger dans les détails plus détaillés disponibles sur le wiki. Enfin, les [Forums de FreeCAD](#) sont également très utiles et fournissent une assistance et des informations supplémentaires.

Manuel : Qu'est-ce que FreeCAD ?



Comme indiqué dans la section précédente, FreeCAD est une application de modélisation 3D paramétrique de conception assistée par ordinateur (CAO). La [conception paramétrique](#) est une méthode de conception d'objets basée sur la définition de paramètres et de relations entre eux. Dans cette approche, les changements de paramètres se propagent automatiquement dans les éléments associés de la conception, ce qui permet des itérations et des adaptations rapides.

Couramment utilisée dans des domaines tels que l'architecture, l'ingénierie et la conception de produits, la conception paramétrique permet de créer des formes complexes de manière efficace, avec la possibilité d'ajuster et d'affiner divers aspects de la conception par la modification des valeurs d'entrée. Cette approche est particulièrement puissante dans des logiciels tels que FreeCAD, où les utilisateurs peuvent définir ces paramètres de manière dynamique afin d'automatiser et d'optimiser le processus de conception.

FreeCAD est une application multiplateforme qui fonctionne sous Windows, macOS et Linux et est entièrement [open-source](#). Cela signifie que son utilisation, sa modification et sa distribution sont libres. FreeCAD est développé par une communauté enthousiaste plutôt que par une seule entreprise, ce qui permet des améliorations et des mises à jour constantes.

Le logiciel comprend de nombreux composants open-source et peut être intégré dans d'autres applications. Il est connu pour être hautement personnalisable et pour prendre en charge une grande variété de formats de fichiers. Cette flexibilité et le soutien solide de sa communauté font de FreeCAD un outil puissant dans le monde du logiciel libre. Développé collectivement par une communauté mondiale de programmeurs, de passionnés et d'utilisateurs, FreeCAD illustre l'esprit de collaboration des projets open-source. Il offre des avantages significatifs tels que la personnalisation et la possibilité de créer des scripts, mettant à la portée de tout utilisateur intéressé par la modélisation et la conception en 3D des fonctionnalités sophistiquées que l'on trouve généralement dans les logiciels professionnels.

Le site officiel de FreeCAD: [site officiel de FreeCAD](#)

Pour en savoir plus :

- [À propos de FreeCAD](#)
- [Fonctionnalités](#)
- [Captures d'écran et exemples](#)

Manuel : Installation

FreeCAD est sous licence [LGPL](#), ce qui vous permet de le télécharger, de l'installer, de le redistribuer et de l'utiliser à toutes fins, commerciales ou non, sans aucune restriction. Vous conservez l'entière propriété des fichiers que vous créez.

FreeCAD fonctionne de manière identique sur Windows, macOS et Linux, bien que le processus d'installation varie en fonction de la plateforme. Pour les utilisateurs de Windows et de Mac, la communauté FreeCAD propose des installateurs précompilés prêts à l'emploi. Sous Linux, le code source est fourni aux responsables des distributions qui compilent le logiciel pour leurs systèmes spécifiques. En général, les utilisateurs de Linux peuvent installer FreeCAD directement via le gestionnaire de logiciels de leur système.

La page officielle de téléchargement de FreeCAD se trouve à l'adresse la [FreeCAD page de téléchargement](#). Des informations supplémentaires sur le processus d'installation sont disponibles sur le [wiki de téléchargement](#) dédié.

Versions de FreeCAD

Les versions stables officielles de FreeCAD sont disponibles sur la page de téléchargement référencée et dans le gestionnaire de logiciels de votre distribution. Cependant, le rythme de développement de FreeCAD est soutenu, avec de nouvelles fonctionnalités et des corrections de bogues incorporées presque quotidiennement. En raison des périodes prolongées entre les versions stables, vous pouvez vouloir expérimenter avec des versions plus récentes de FreeCAD. Ces versions de développement, ou pré-versions, peuvent être trouvées sur la même page de téléchargement. Pour les utilisateurs d'Ubuntu ou de Fedora, la communauté FreeCAD fournit également des "versions de développement" sur le [PPA](#) (Personal Package Archives) et le [copr](#), régulièrement mis à jour.

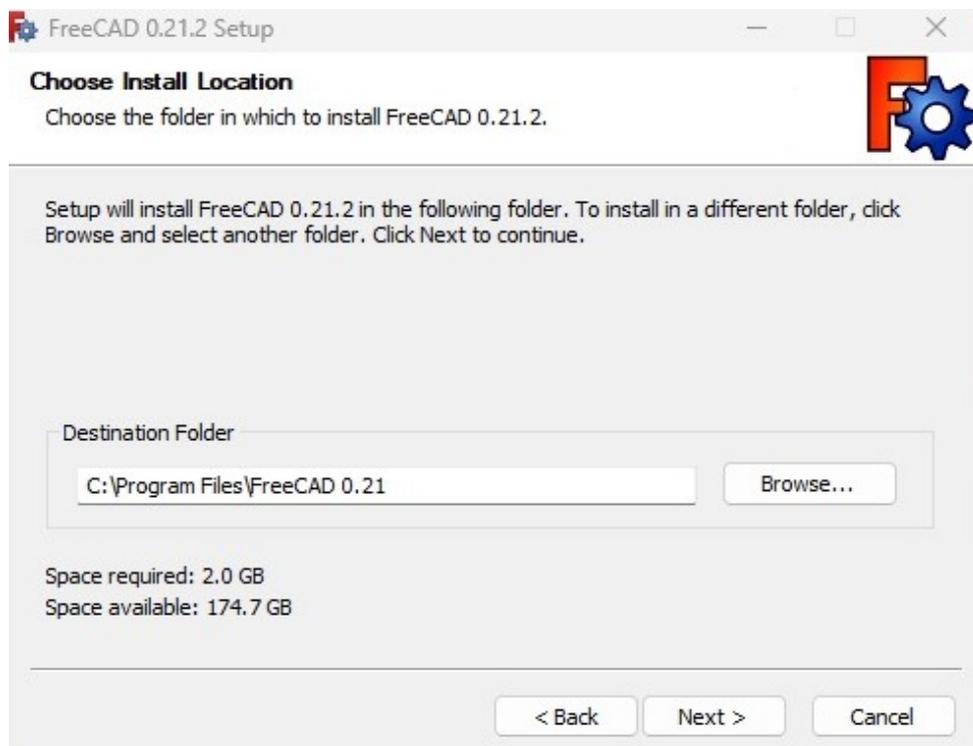
Si vous envisagez d'installer FreeCAD sur une machine virtuelle, sachez que ses performances peuvent être considérablement réduites, voire inutilisables, en raison de la prise en charge limitée d'OpenGL dans de nombreuses machines virtuelles.

Installation sur Windows

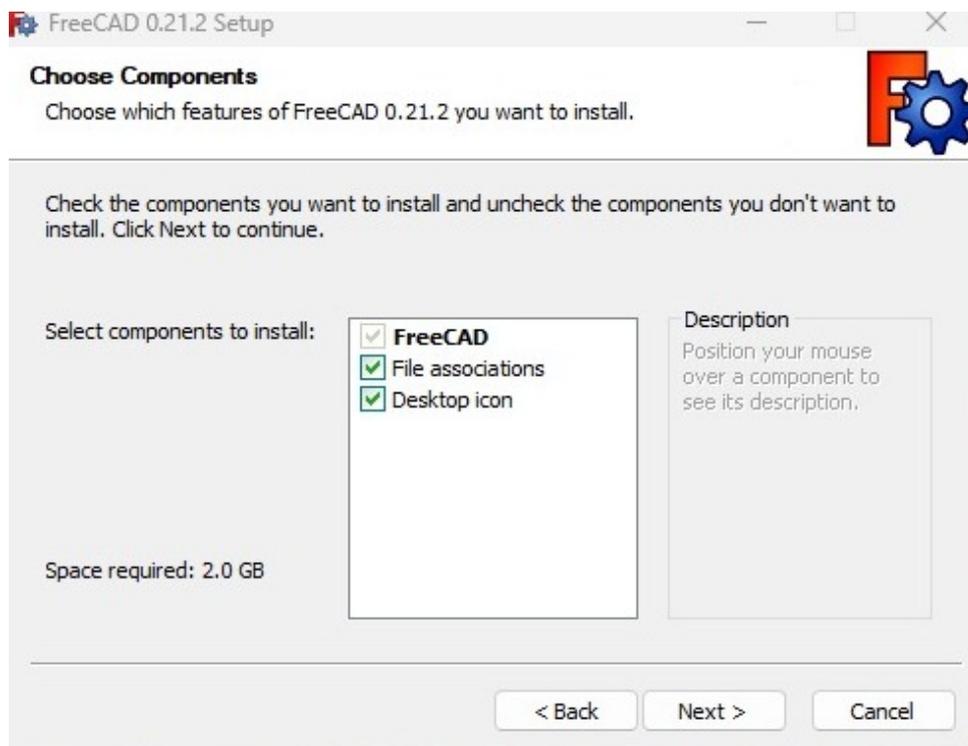
1. Téléchargez un programme d'installation (.exe) à partir de la page de téléchargement. Les programmes d'installation de FreeCAD devraient fonctionner sur n'importe quelle version de Windows à partir de Windows 7.
2. Acceptez les termes de la licence LGPL. C'est l'un des rares cas où vous pouvez vraiment cliquer sur le bouton « accepter » sans lire le texte. Pas de clauses cachées :



3. Vous pouvez laisser le chemin d'accès par défaut ici, ou le changer si vous le souhaitez :



4. Assurez-vous de cocher tous les composants à installer :



5. C'est tout. L'installation est maintenant terminée et vous pouvez commencer à explorer les possibilités de FreeCAD.

Installation d'une version de développement

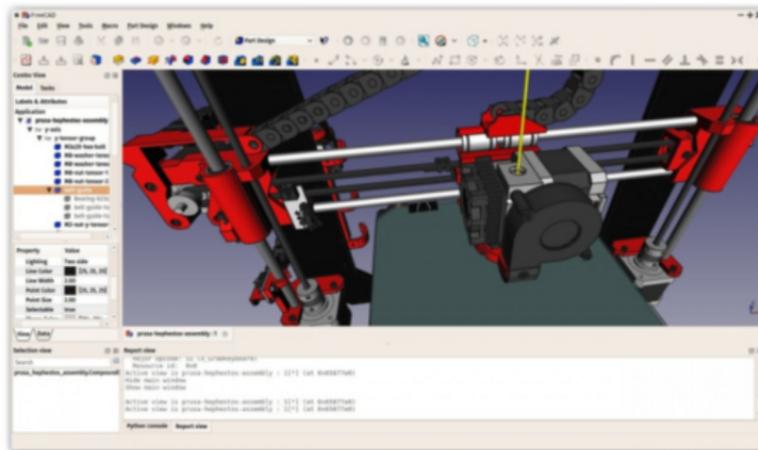
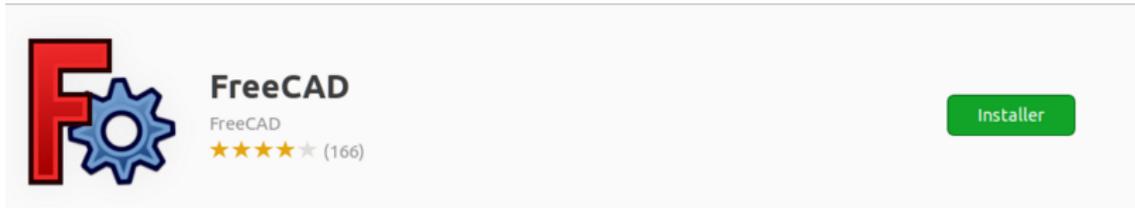
La livraison de FreeCAD et le développement d'un programme d'installation impliquent un investissement considérable en temps et en efforts. C'est pourquoi les versions de développement (également appelées pré-versions) sont généralement livrées sous la forme d'archives .zip ou .7z situées sur la [page de téléchargement de FreeCAD](#). Il n'est pas nécessaire de procéder à une installation formelle avec ces fichiers. Il suffit d'en extraire le contenu et de lancer FreeCAD en double-cliquant sur le fichier FreeCAD.exe qui se trouve à l'intérieur. Cette approche vous permet également de maintenir la version stable et la version « instable » sur le même ordinateur. C'est comme si vous aviez dans votre garage une voiture de tous les jours et un jet pack expérimental !

Installation sur Linux

Pour les utilisateurs des distributions Linux modernes telles qu'Ubuntu, Fedora, openSUSE, Debian, Mint et Elementary, l'installation de FreeCAD est aussi simple qu'un simple clic. Vous pouvez l'installer de manière transparente via l'outil de gestion des logiciels fourni par votre distribution, bien que l'apparence de ces outils puisse différer des images d'illustration puisque chaque distribution utilise sa propre application.

1. Ouvrez le gestionnaire de logiciels et recherchez "freecad" :

2. Cliquez sur le bouton "installer" et c'est tout, FreeCAD s'installe. N'oubliez pas de noter ensuite!



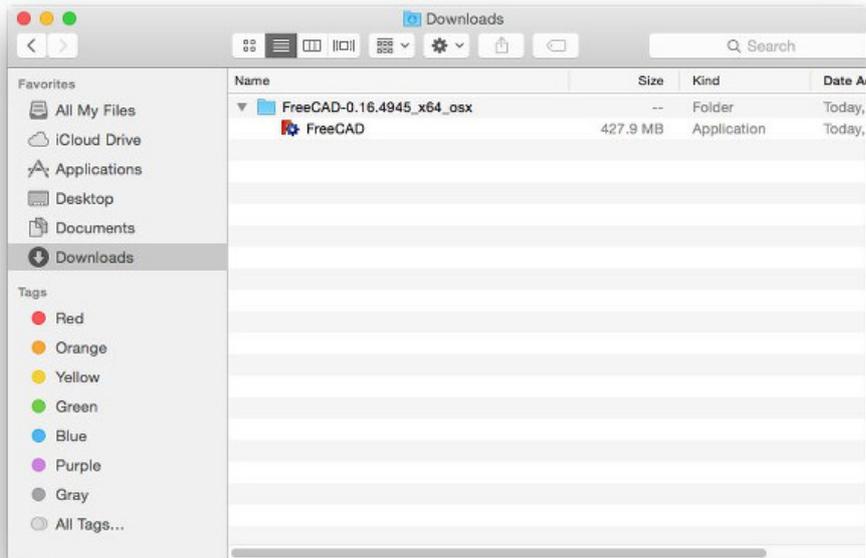
Modes alternatifs

Un des gros avantages de l'utilisation sous Linux ce sont les nombreuses possibilités d'adapter votre logiciel, alors ne vous retenez pas. Pour les utilisateurs d'Ubuntu et de ses dérivés, FreeCAD peut être installé à partir d'un [PPA](#) maintenue par la communauté FreeCAD, qui comprend à la fois des versions stables et des versions de développement. Sur Fedora, vous pouvez accéder aux dernières versions de développement de FreeCAD via [copr](#). De plus, comme FreeCAD est un logiciel libre, vous avez la liberté de [compiler FreeCAD vous-même](#).

Installation sur Mac OS

L'installation de FreeCAD sur Mac OSX est aujourd'hui aussi simple que sur d'autres plates-formes. Cependant, étant donné qu'il y a moins de gens dans la communauté qui possèdent un Mac, les paquets disponibles sont souvent en retard de quelques versions derrière les autres plates-formes.

1. Téléchargez un package compressé correspondant à votre version.
2. Ouvrez le dossier Téléchargements et développez le fichier zip téléchargé:



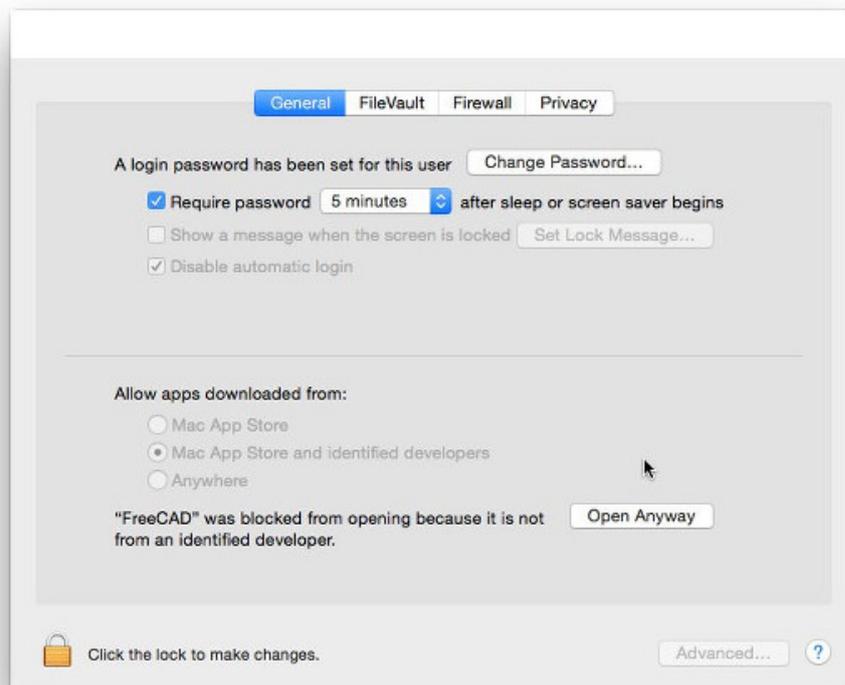
3. Faites glisser l'application FreeCAD de l'intérieur du zip vers le dossier Applications :



4. Ça y est, FreeCAD est installé !



5. Si le système empêche FreeCAD de se lancer, en raison des autorisations restreintes pour les applications ne provenant pas de l'App Store, vous devrez l'activer dans les paramètres du système :



Désinstallation

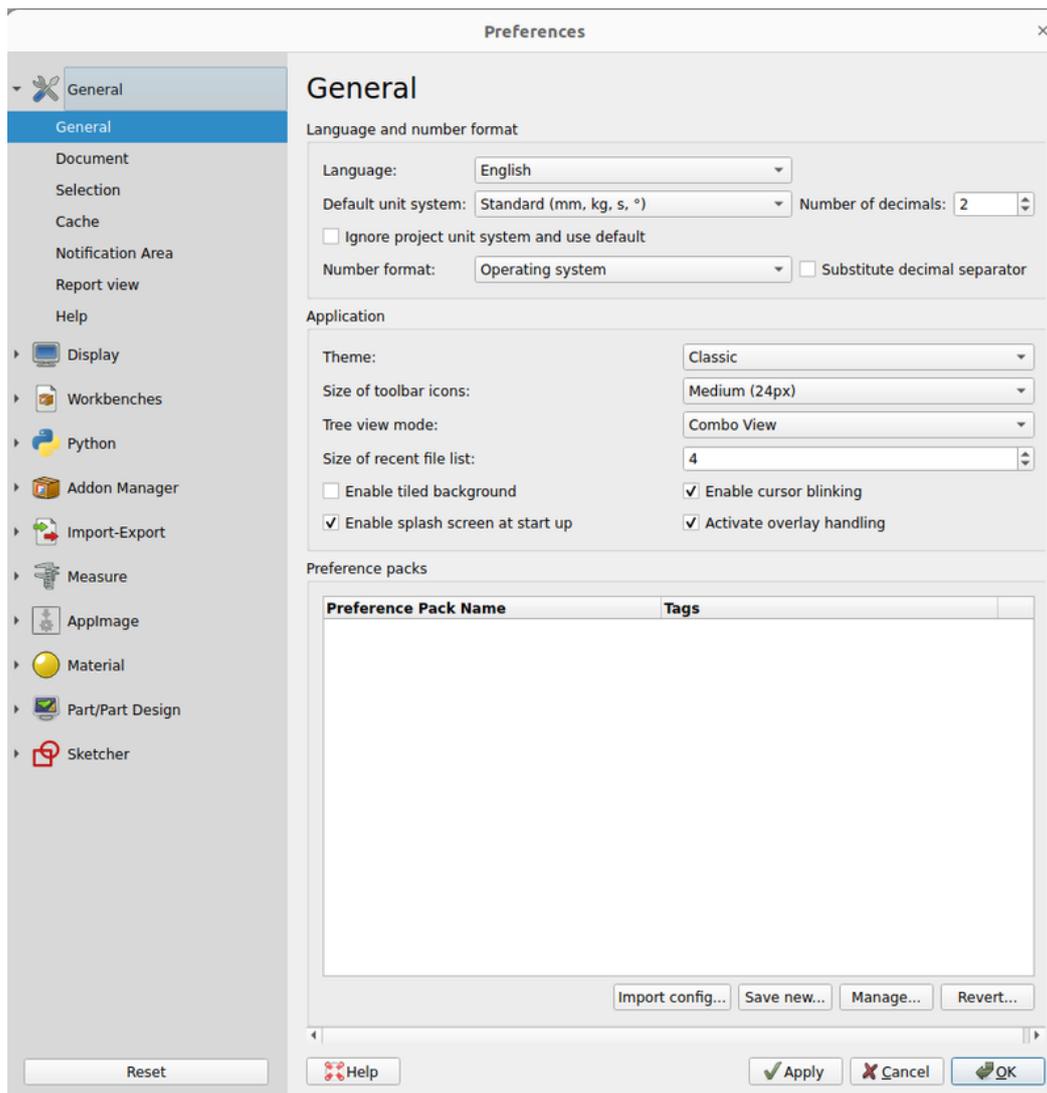
Idéalement, vous ne voudrez jamais vous séparer de FreeCAD, mais si vous devez le désinstaller, sachez que la procédure est simple.

- Sous Windows, utilisez l'option familière « supprimer un logiciel » du panneau de configuration.
- Pour les utilisateurs de Linux, désinstallez le logiciel à l'aide du gestionnaire de logiciels que vous avez utilisé pour l'installer.
- Les utilisateurs de Mac ont la tâche la plus facile : il suffit de faire glisser FreeCAD du dossier Applications vers la corbeille.

Définir les préférences de base

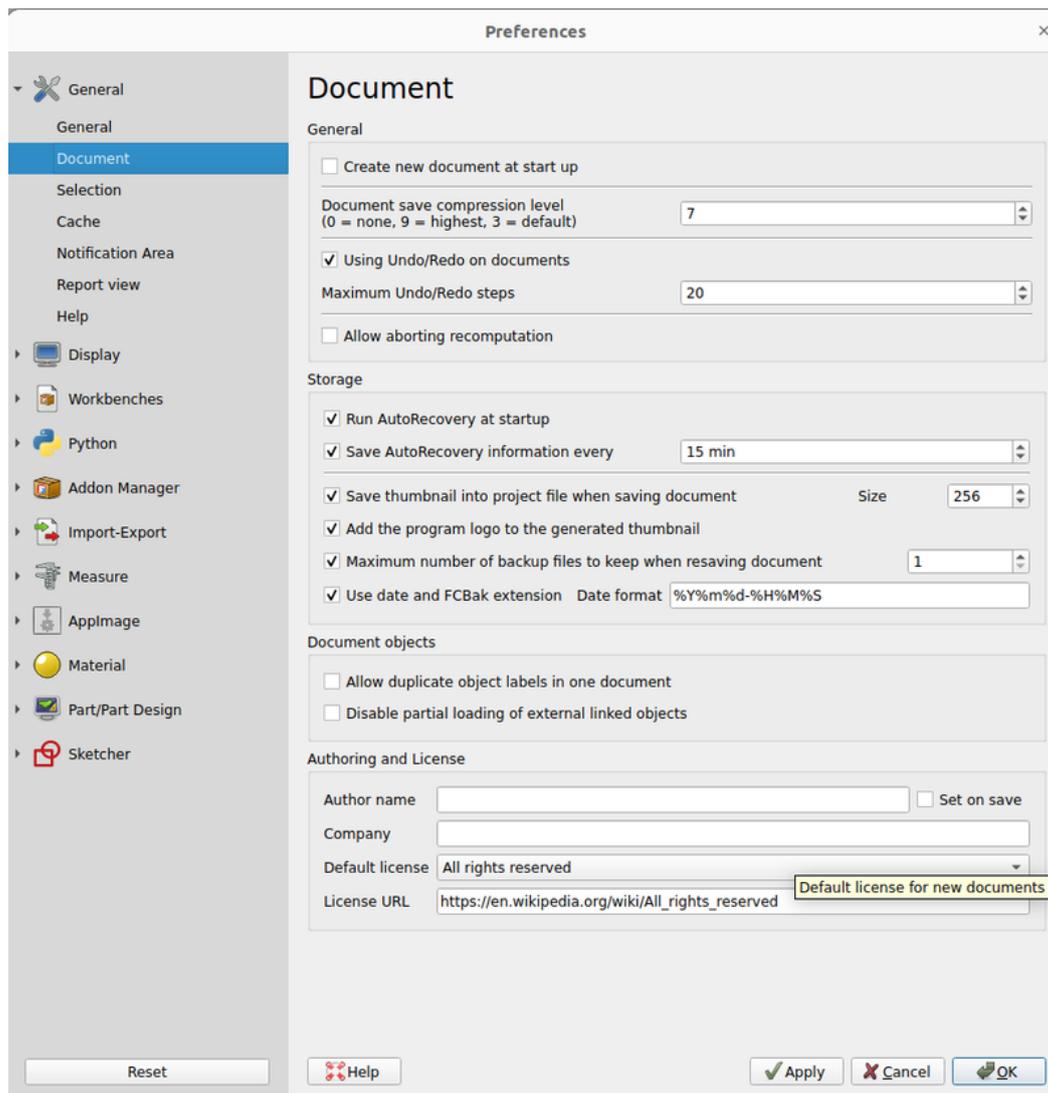
Après avoir installé FreeCAD, vous souhaitez probablement le personnaliser en ajustant certains paramètres. Vous pouvez trouver les paramètres de préférences dans FreeCAD en faisant *Édition* → *Préférences*. Vous trouverez ci-dessous quelques paramètres de base que vous pourriez envisager de modifier immédiatement, mais n'hésitez pas à explorer les différentes pages de préférences pour adapter le logiciel à vos besoins.

Catégorie générale, onglet Général



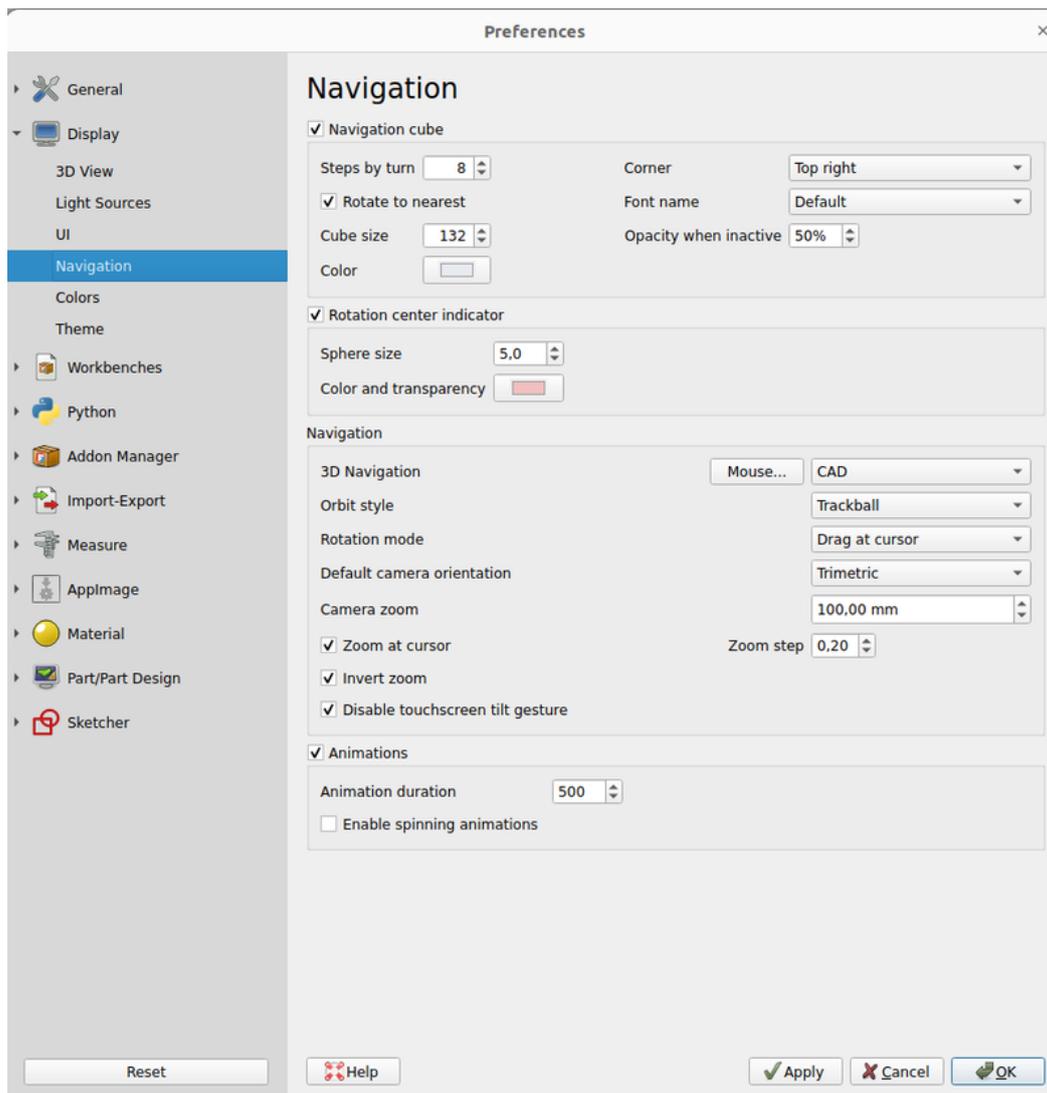
1. **Langue** : par défaut, FreeCAD sélectionnera la langue de votre système d'exploitation, mais vous avez la possibilité de la changer. Grâce à la contribution de nombreux contributeurs, FreeCAD est disponible dans un large éventail de langues.
2. **Unités**: ce paramètre vous permet de choisir le système d'unités par défaut pour vos projets.

Catégorie Général, onglet Document



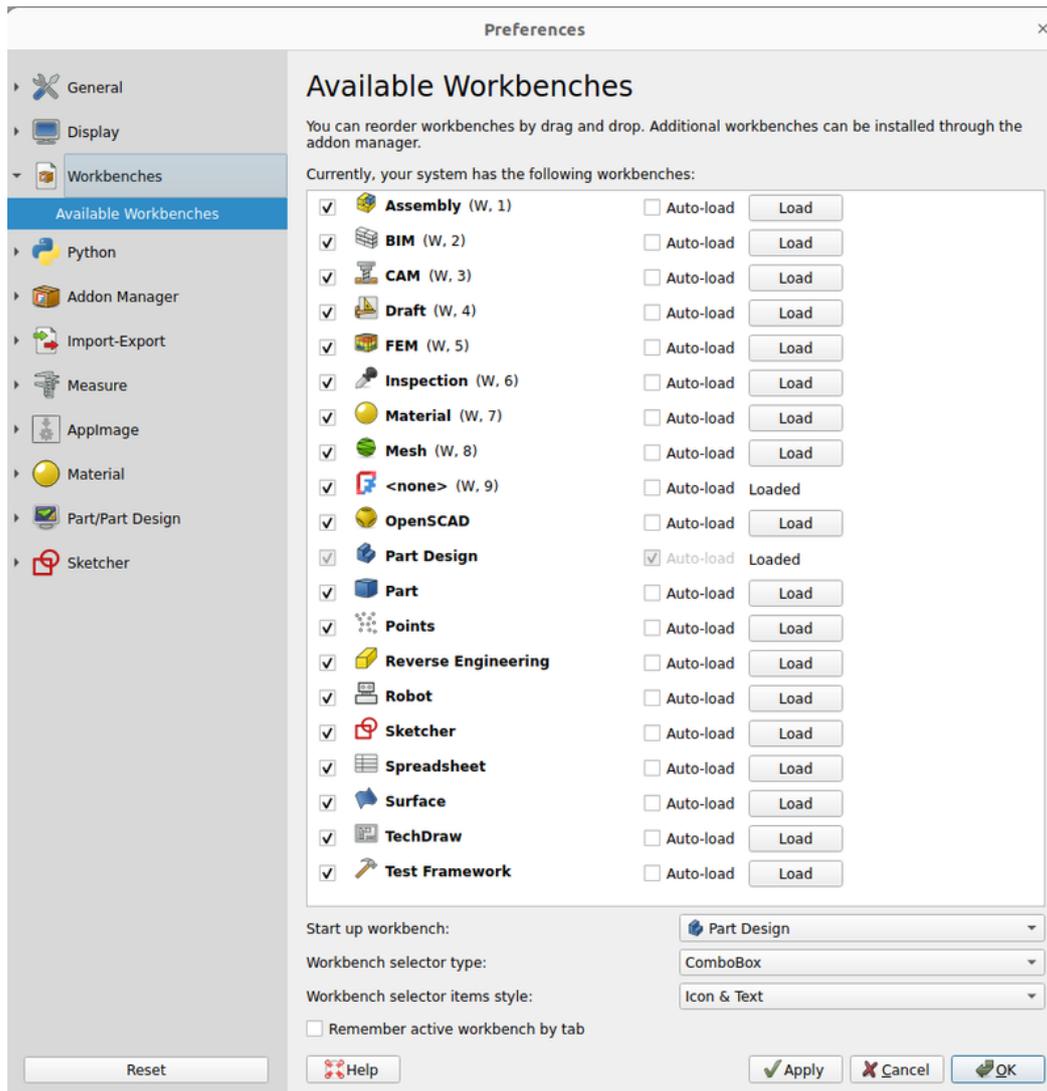
1. **Créer un nouveau document au démarrage** : FreeCAD ouvrira automatiquement un nouveau document à chaque démarrage du programme.
2. **Stockage** : configurez ici les paramètres pour vous aider à récupérer votre travail en cas de crash.
3. **Création et licence** : dans cette zone, vous pouvez déterminer les paramètres pour les nouveaux fichiers. Pour faciliter le partage, vous pouvez commencer par une licence plus permissive, avec une licence copyleft telle que la Creative Commons.

Catégorie Affichage, onglet Navigation



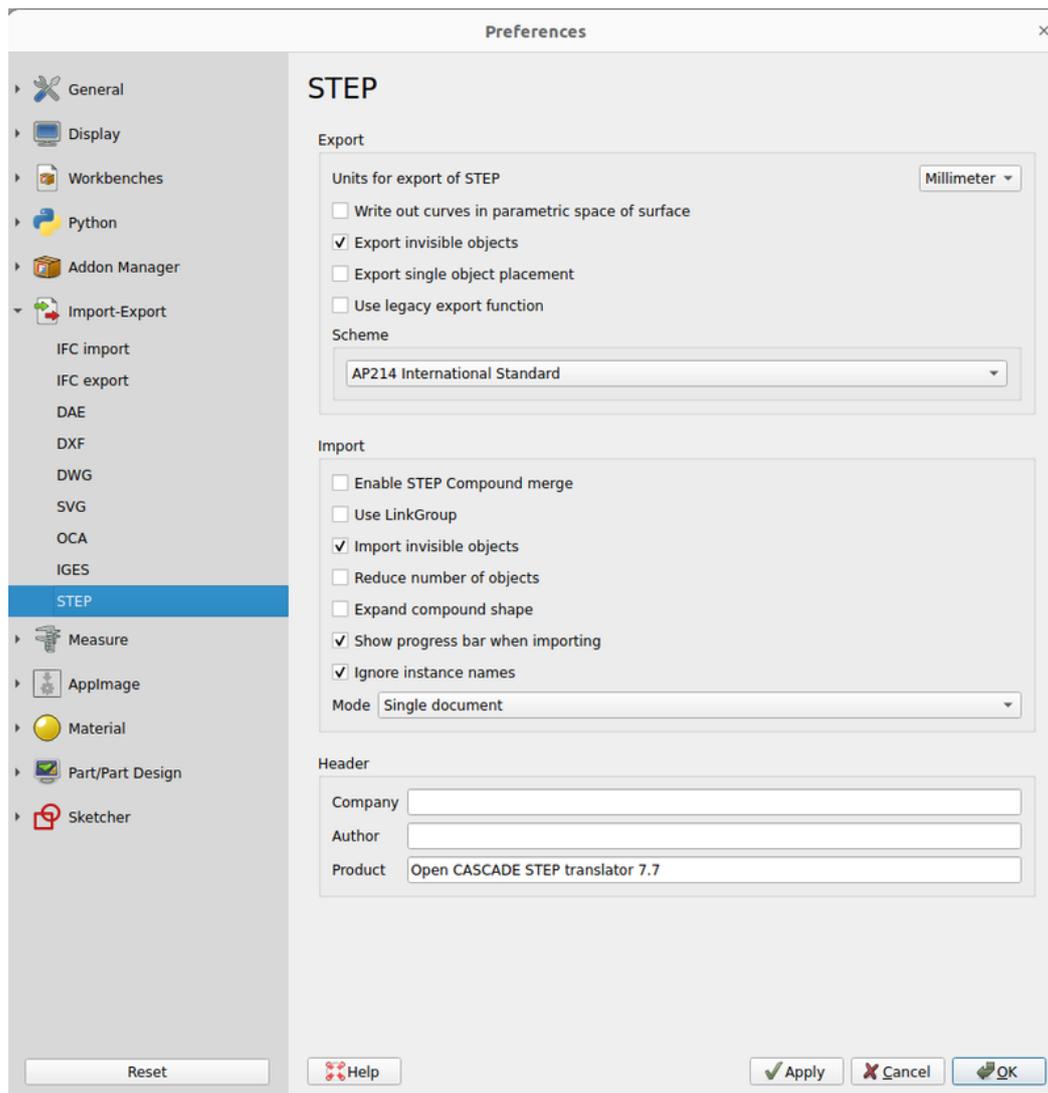
1. **Zoomer sur le curseur** : lorsque cette option est activée, les actions de zoom sont centrées sur le curseur de la souris. Si elle est désactivée, le zoom se concentre sur le centre de la vue.
2. **Inverser le zoom** : cette option inverse la direction du zoom par rapport au mouvement de la souris.

Onglet Ateliers



Bien que FreeCAD s'ouvre généralement sur la page d'accueil, ce paramètre vous permet de l'ignorer. Vous pouvez démarrer directement dans l'atelier de votre choix. En outre, vous pouvez personnaliser les postes de travail affichés dans le menu de sélection.

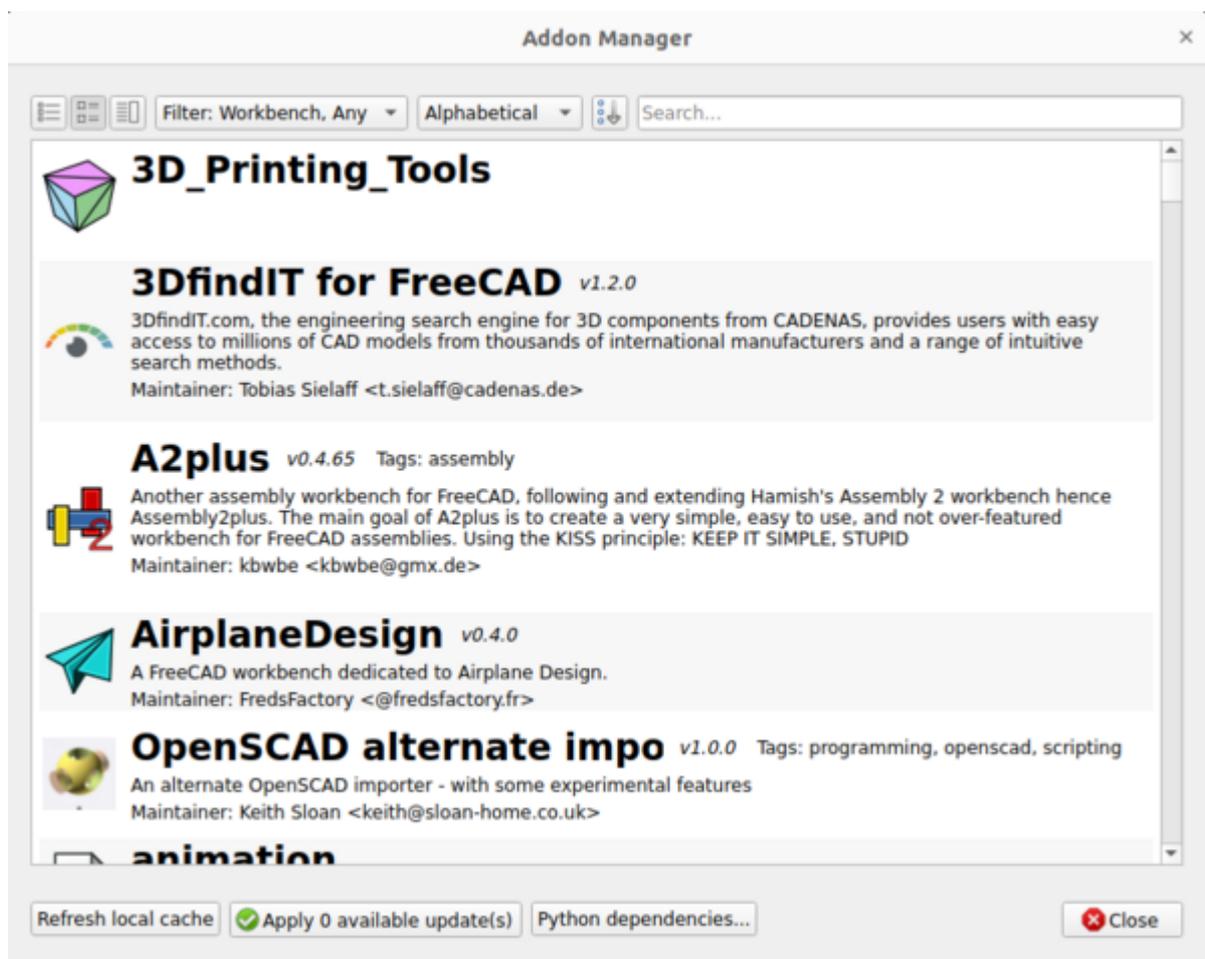
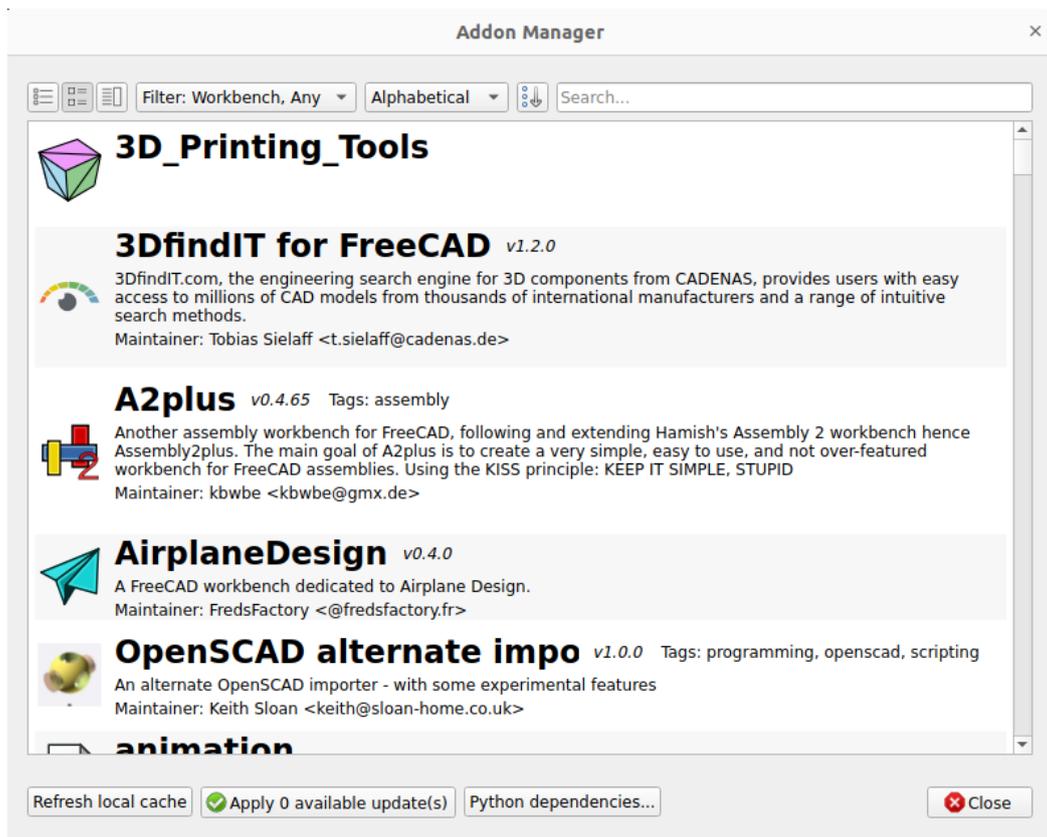
Onglet Importer/Exporter



Cette section permet de définir les paramètres de base pour l'importation et l'exportation dans différents formats.

Installation de contenu supplémentaire

Au fur et à mesure que la communauté FreeCAD s'agrandit et que la facilité de personnalisation augmente, de nombreuses contributions externes et des projets secondaires réalisés par des membres de la communauté commencent à apparaître. Beaucoup de ces projets prennent la forme d'ateliers ou de macros, et vous pouvez facilement les ajouter à votre boîte à outils via le [gestionnaire des extensions](#), accessible depuis le menu Outils. Le gestionnaire des extensions vous ouvre un monde de possibilités, vous permettant d'installer divers composants intéressants, tels que :



1. Une [bibliothèque de pièces](#). Cette bibliothèque est un trésor de modèles utiles ou de fragments de modèles créés par les utilisateurs de FreeCAD. Tous les

éléments de cette bibliothèque sont disponibles gratuitement pour être utilisés dans vos projets et sont accessibles directement dans votre configuration FreeCAD.

2. Des [ateliers en plus](#). Il s'agit d'extensions spécialisées qui améliorent les fonctionnalités de FreeCAD pour des tâches spécifiques. Les exemples d'applications comprennent l'animation de pièces de modèles ou la gestion de processus de construction spécifiques tels que le pliage de tôles ou la modélisation des données du bâtiment (BIM). Des informations détaillées sur chaque atelier, y compris une vue d'ensemble des outils qu'il contient, sont fournies sur la page de chaque extension, accessible par le lien correspondant dans le gestionnaire des extensions.
3. Une large gamme de [macros](#) est également disponible au téléchargement. Celles-ci peuvent considérablement rationaliser votre flux de travail, et une documentation détaillée sur la façon de les utiliser peut être trouvée sur le [wiki de FreeCAD](#).

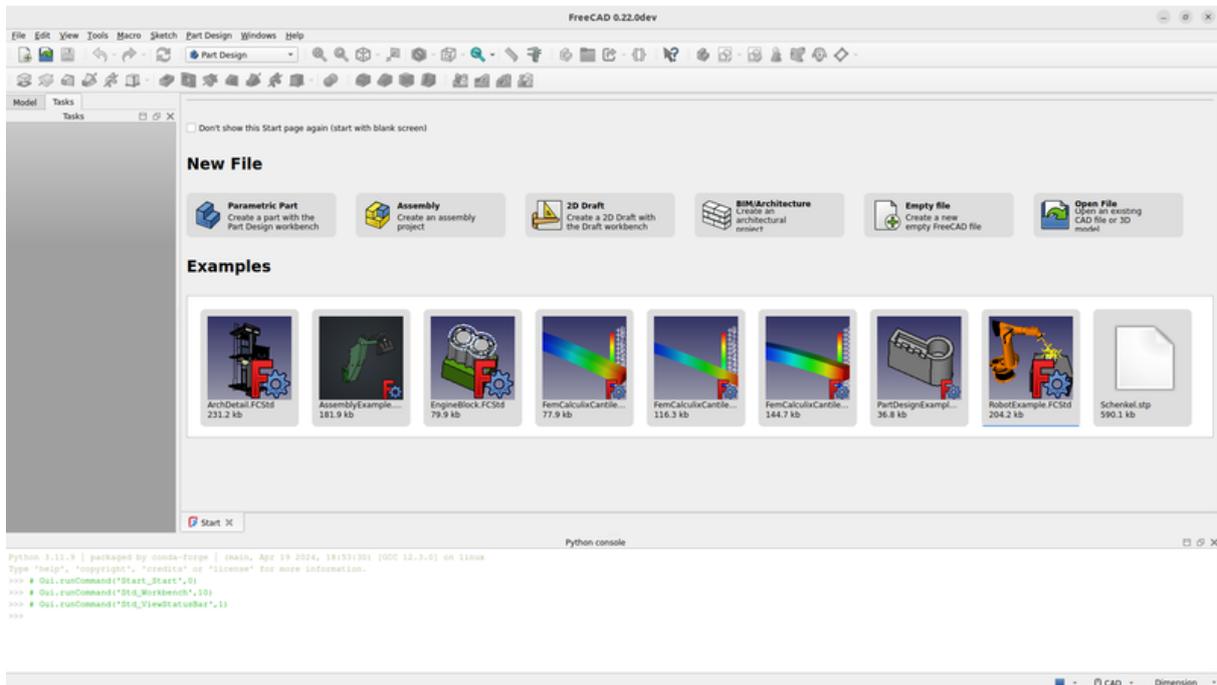
À partir de FreeCAD v0.17.9940, la méthode d'installation recommandée pour tous les outils ci-dessus est le gestionnaire des extensions intégré. Cependant, si pour une raison quelconque cette option n'est pas disponible, une installation manuelle est toujours possible. Pour plus d'informations, consultez la [FreeCAD page des extensions](#).

Lire plus d'informations

- [Plus d'options de téléchargement](#)
- [Dépôt FreeCAD pour Ubuntu](#)
- [Dépôt des extensions de FreeCAD pour Ubuntu](#)
- [Compilez FreeCAD vous-même](#)
- [Traductions FreeCAD](#)
- [Page du github FreeCAD](#)
- [Gestionnaire des extensions de FreeCAD](#)

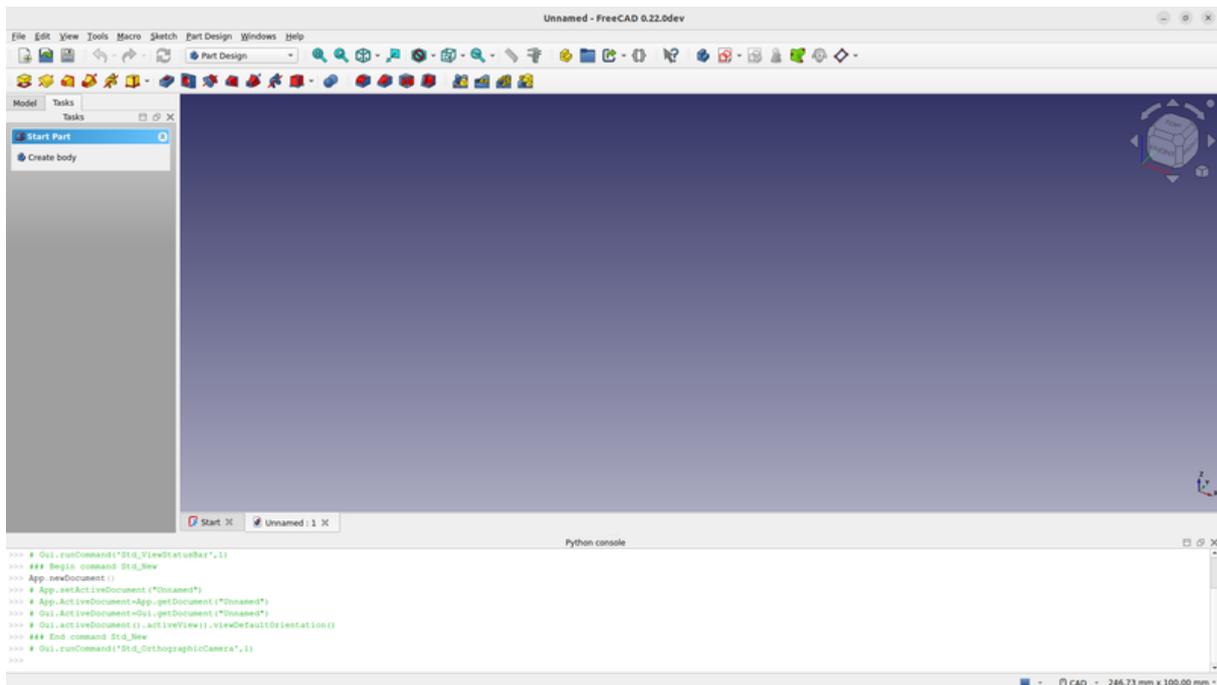
Manuel : L'interface FreeCad

FreeCAD est basé sur le [framework Qt](#) et se caractérise par une interface simple et directe. Les utilisateurs de CAO les plus expérimentés pourront identifier les similitudes avec d'autres logiciels, tandis que les nouveaux utilisateurs trouveront qu'il est facile de naviguer et de découvrir les différentes options offertes par FreeCAD. Voici l'apparence par défaut de FreeCAD :



La page de démarrage sert d'écran d'accueil, conçu pour faciliter l'accès rapide et facile aux principales zones de FreeCAD qu'un utilisateur pourrait souhaiter explorer. Grâce à elle, les utilisateurs peuvent sans effort créer de nouvelles pièces, ouvrir des fichiers récents et lancer des dessins en 2D. En outre, il comporte des raccourcis vers des ressources utiles telles que des tutoriels et des forums d'utilisateurs, qui sont inestimables pour les débutants et les utilisateurs expérimentés à la recherche de conseils ou d'astuces. Les utilisateurs peuvent facilement personnaliser l'apparence de la page de démarrage en fonction de leurs préférences.

Au fur et à mesure que vous vous familiarisez avec FreeCAD, vous pouvez ajuster les paramètres dans les préférences. Vous pouvez ainsi configurer FreeCAD pour qu'il s'ouvre directement dans l'un des ateliers avec un nouveau document prêt à l'emploi lorsque vous le lancez. Vous pouvez également fermer l'onglet Page de démarrage et créer manuellement un nouveau document.



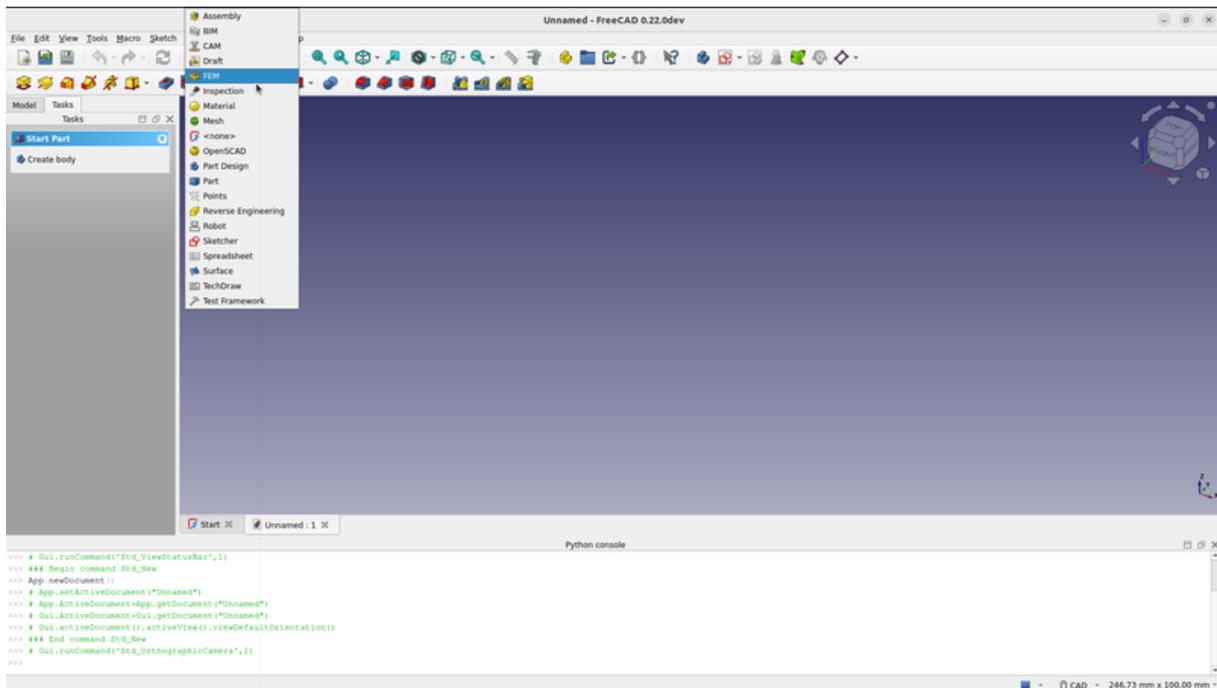
Les ateliers

FreeCAD utilise un système appelé « ateliers », similaire aux cadres conceptuels utilisés dans les logiciels de conception avancés tels que Revit ou CATIA. L'idée d'un atelier est analogue aux stations spécialisées d'un laboratoire scientifique, où différents postes de travail sont équipés pour des types d'expériences distincts. Dans un laboratoire, vous pouvez avoir une zone dédiée à la chimie, une autre à la biologie et une troisième à la physique, chacune équipée des outils spécifiques nécessaires à ces disciplines.

Dans le contexte de FreeCAD, chaque atelier est adapté à un type de tâche particulier, organisant tous les outils nécessaires à cette activité dans une seule interface. Lorsque vous passez d'un atelier à l'autre, l'ensemble des outils et des commandes visibles dans l'interface utilisateur s'adapte pour refléter les besoins de la tâche sélectionnée, bien que le contenu du projet ou la « scène » sur laquelle vous travaillez ne change pas. Cela permet des transitions transparentes dans le flux de travail, par exemple en commençant une conception avec des formes 2D de base dans l'atelier Draft, puis en élaborant ces conceptions avec des outils de modélisation avancés dans l'atelier Part.

Les termes « Atelier » et « Module » sont parfois utilisés de manière interchangeable, mais ils ont des significations distinctes dans FreeCAD. Un module est une extension qui ajoute des fonctionnalités à FreeCAD, alors qu'un atelier est un type spécifique de module équipé de ses propres composants d'interface utilisateur tels que les barres d'outils et les menus, conçus pour faciliter des types de tâches spécifiques. Ainsi, chaque atelier est un module, mais tous les modules ne sont pas des ateliers.

Le contrôle le plus important de l'interface FreeCAD est le sélecteur d'atelier (Workbench) que vous utilisez pour passer d'un atelier à l'autre :



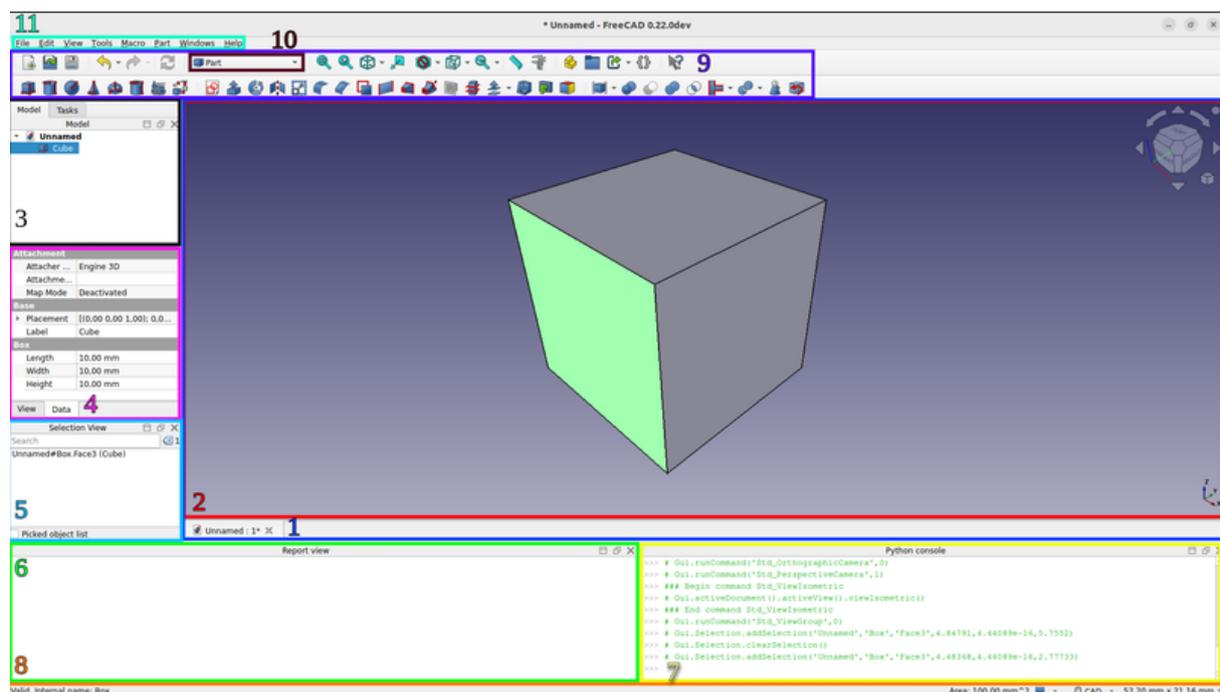
-  L'[atelier Assembly](#) pour la construction et la résolution d'assemblages mécaniques. [introduit dans la version 1.0](#)
-  L'[atelier BIM](#) pour travailler avec des éléments architecturaux et créer des modèles [BIM](#). Il combine l'atelier Arch et l'ancien atelier BIM externe disponible dans la [version 0.21 et précédentes](#).
-  L'[atelier CAM](#) est utilisé pour produire des instructions en G-Code. Cet atelier était appelé "atelier Path" [version 0.21 et précédentes](#).
-  L'[atelier Draft](#) contient des outils 2D et des opérations de CAO 2D et 3D de base.
-  L'[atelier FEM](#) fournit un flux de travail d'analyse par éléments finis (FEA).
-  L'[atelier Inspection](#) est fait pour vous donner des outils spécifiques pour l'examen des formes. Il en est encore aux premiers stades de développement.
-  L'[atelier Material](#) gère le système des matériaux de FreeCAD. [introduit dans la version 1.0](#)
-  L'[atelier Mesh](#) pour travailler avec des maillages triangulés.
-  L'[atelier OpenSCAD](#) pour l'interopérabilité avec OpenSCAD et la réparation de [Géométrie Solide Constructive](#) (CSG).
-  L'[atelier Part](#) pour travailler avec des primitives géométriques et des opérations booléennes.
-  L'[atelier PartDesign](#) pour créer des formes de pièces à partir d'esquisses.
-  L'[atelier Points](#) pour travailler avec des nuages de points.

-  L'[atelier Reverse Engineering](#) est destiné à fournir des outils spécifiques pour convertir des formes/solides/mailles en fonctionnalités paramétriques compatibles avec FreeCAD.
-  L'[atelier Robot](#) pour étudier les mouvements des robots. Non maintenu pour le moment.
-  L'[atelier Sketcher](#) pour travailler avec des esquisses à géométrie contrainte.
-  L'[atelier Spreadsheet](#) pour créer et manipuler des données de feuilles de calcul.
-  L'[atelier Surface](#) fournit des outils pour créer et modifier des surfaces. Il est similaire à l'outil [Part Générateur de formes](#) à partir des arêtes.
-  L'[atelier TechDraw](#) pour produire des dessins techniques à partir de modèles 3D. C'est le successeur de l'[atelier Drawing](#).
-  L'[atelier Test](#) est destiné au débogage de FreeCAD.

Les ateliers déconcertent souvent les nouveaux utilisateurs, car il n'est pas toujours facile de savoir dans quel atelier chercher un outil spécifique. Mais ils s'apprennent rapidement et, au bout d'un certain temps, ils leur paraîtront naturels. Les nouveaux utilisateurs se rendent rapidement compte que les ateliers sont un moyen pratique d'organiser la multitude d'outils offerts par FreeCAD. En outre, les ateliers sont entièrement personnalisables.

L'interface

Regardons mieux les différentes parties de l'interface :



La fenêtre principale de l'application peut être divisée en 11 sections :

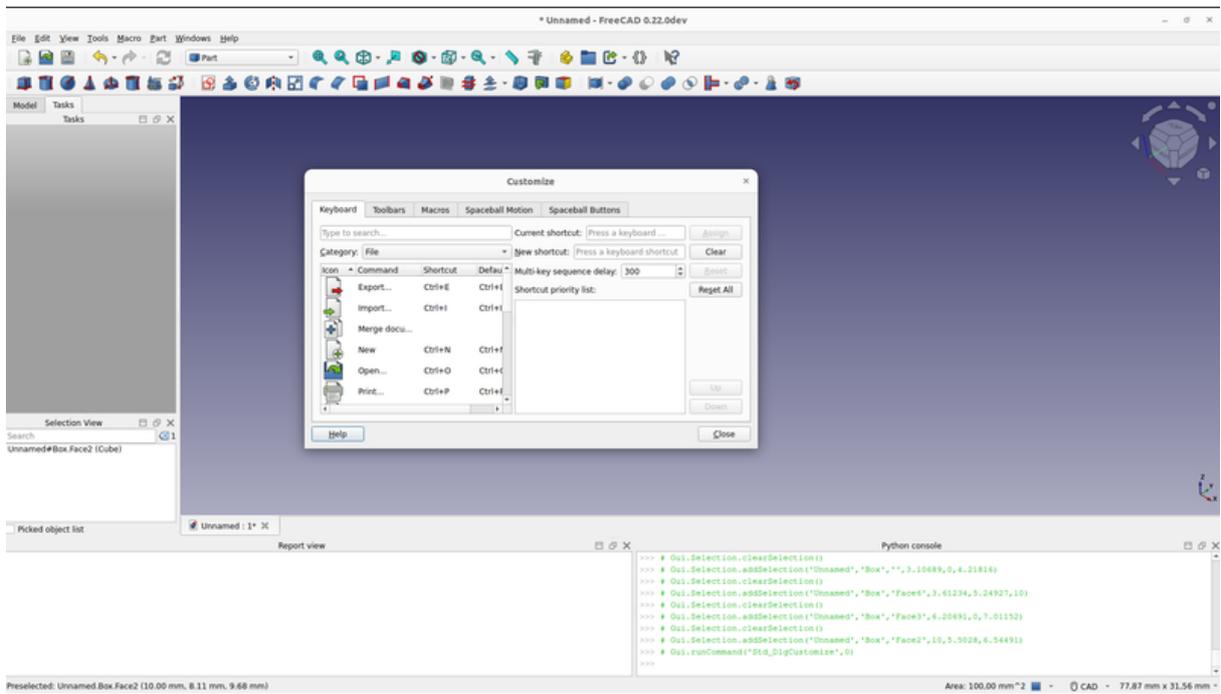
1. La [zone de vue principale](#) qui peut contenir différentes fenêtres à onglets.
2. La [vue 3D](#), normalement intégrée à la [zone de vue principale](#). La vue 3D est l'élément central de l'interface. Elle affiche et permet de manipuler les objets sur lesquels vous travaillez. Il est possible d'avoir plusieurs vues du même document (ou des mêmes objets) ou d'avoir plusieurs documents ouverts simultanément. De plus, chaque vue peut être détachée de la fenêtre principale séparément.
3. L'onglet [Modèle](#) et l'onglet [Tâches](#).
 1. L'onglet [Modèle](#) vous montre le contenu et la structure de votre document.
 2. L'onglet [Tâches](#) est l'endroit où FreeCAD vous demandera des valeurs spécifiques à l'atelier et à l'outil que vous utilisez (par exemple les dimensions d'un objet).
4. L'[éditeur de propriétés](#) qui apparaît lorsque l'onglet [Modèle](#) est actif dans l'interface. Il permet de gérer les propriétés exposées publiquement des objets du document. Il se compose des sections [Data](#) et [View](#), qui présentent respectivement les propriétés de visualisation et les propriétés paramétriques des objets.
5. La [fenêtre de sélection](#) qui indique les objets ou les sous-éléments d'objets (sommets, arêtes, faces) qui sont sélectionnés.
6. La [vue rapport](#) où sont affichés les messages, les avertissements et les erreurs.
7. La [console Python](#) où toutes les commandes exécutées sont imprimées et où vous pouvez entrer du code Python.
8. La [barre d'état](#) où apparaissent certains messages et infobulles.
9. La zone de la barre d'outils, où les barres d'outils sont ancrées.
10. Le [sélecteur d'atelier](#) où vous sélectionnez l'[atelier](#) actif.
11. Le [menu standard](#) qui contient les opérations de base du programme.

La plupart des panneaux ci-dessus peuvent être cachés ou révélés par **Affichage** → **Panneaux**.

Personnalisation de l'interface

L'interface de FreeCAD est conçue pour une personnalisation poussée. Toutes les barres d'outils et tous les panneaux peuvent être déplacés, empilés ou même ancrés dans diverses configurations selon les préférences de l'utilisateur. En outre, ils peuvent être fermés puis rouverts selon les besoins. Au-delà de ces capacités, les utilisateurs disposent de nombreuses autres options, notamment la possibilité de créer des barres d'outils personnalisées avec des outils sélectionnés dans l'un des ateliers disponibles, et d'attribuer ou de modifier des raccourcis clavier pour rationaliser le flux de travail. Cette flexibilité permet aux utilisateurs d'adapter l'environnement à leurs besoins et préférences spécifiques.

Ces options de personnalisation avancées sont disponibles à partir du menu **Outils** → **Personnaliser...** :



Pour en savoir plus :

- [Pour commencer avec FreeCAD](#)
- [Personnalisation de l'interface](#)
- [Ateliers](#)
- [En savoir plus sur Python](#)

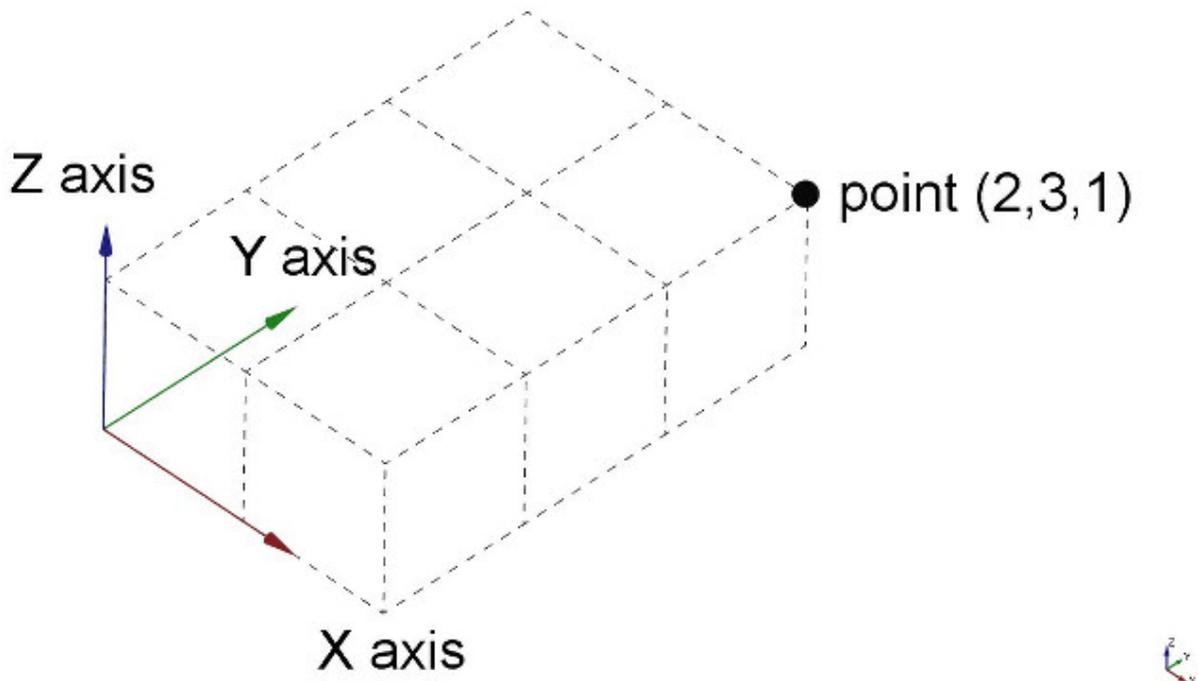
Manuel : Navigation dans la vue 3D

Un mot sur l'espace 3D

Si vous êtes novice en matière d'applications de modélisation 3D, il est essentiel de comprendre certains concepts fondamentaux avant de poursuivre. Si vous avez déjà de l'expérience avec ce type de logiciel, vous pouvez sauter cette introduction.

FreeCAD fonctionne dans un espace tridimensionnel ([Espace Euclidien](#)) centré sur un point d'origine défini par trois axes : X, Y et Z. Typiquement, lorsque l'on regarde FreeCAD depuis le haut, l'axe X s'étend horizontalement vers la droite, l'axe Y s'étend vers l'arrière et l'axe Z se déplace verticalement vers le haut. Le point d'intersection de ces axes, où chaque coordonnée est égale à zéro, est appelé origine.

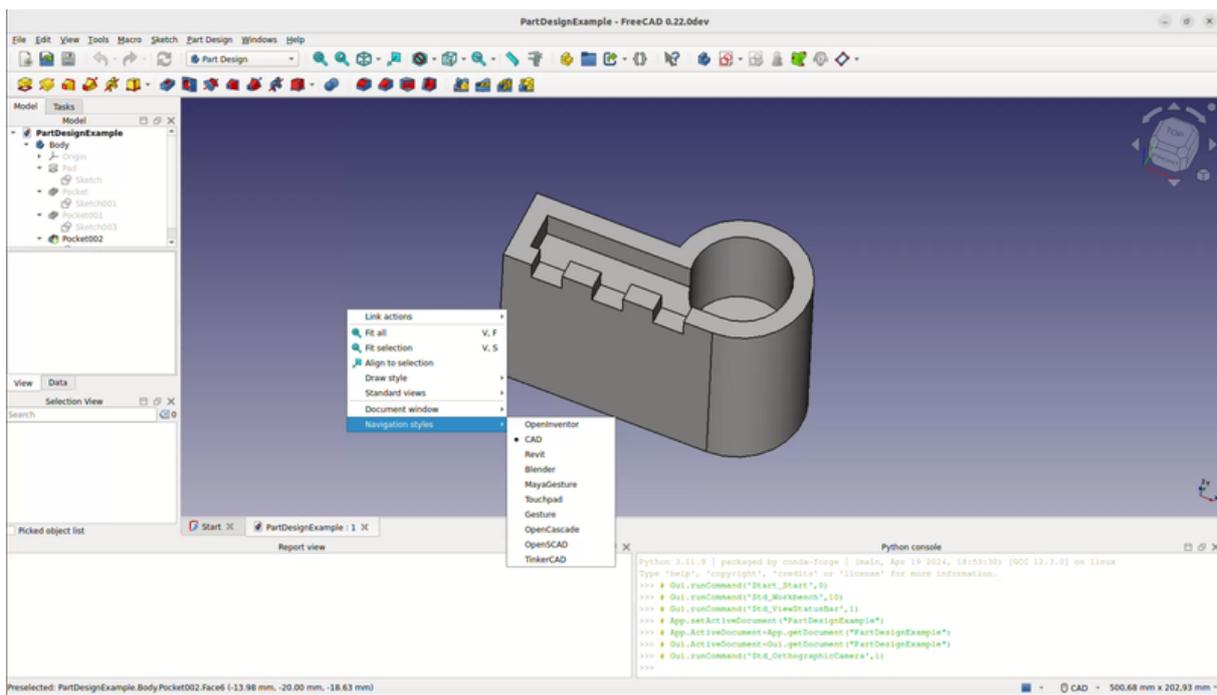
Chaque emplacement dans l'espace de FreeCAD est déterminé par des coordonnées (x, y, z). Par exemple, un point situé aux coordonnées (2,3,1) est positionné à 2 unités le long de l'axe X, à 3 unités le long de l'axe Y et à 1 unité le long de l'axe Z. Naviguer dans cet espace revient à manipuler une caméra. Vous pouvez déplacer la caméra vers la gauche, la droite, le haut ou le bas (panoramique), la faire pivoter autour du point focal (rotation) ou la rapprocher ou l'éloigner des objets (zoom), ce qui vous permet d'explorer votre projet sous différentes perspectives.



La vue 3D FreeCAD

Navigation à la souris

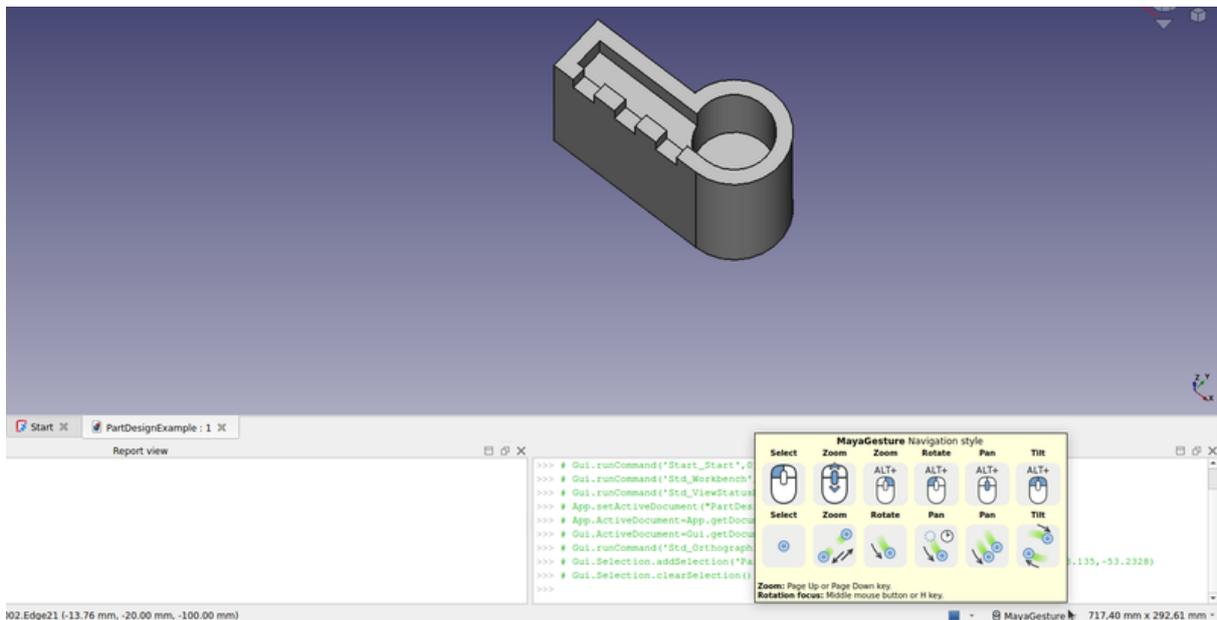
La navigation dans la [vue 3D](#) de FreeCAD peut être réalisée à l'aide de diverses entrées, notamment une souris, un dispositif de navigation spatiale, des raccourcis clavier, un pavé tactile, ou toute combinaison de ces éléments. FreeCAD offre une gamme de [styles de navigation](#) qui définissent la façon dont les trois opérations de visualisation fondamentales (panoramique, rotation et zoom) sont exécutées. En outre, ces styles régissent la manière dont les objets sont sélectionnés dans l'espace de travail. Pour accéder à ces styles de navigation et passer de l'un à l'autre, vous pouvez accéder à l'écran Préférences ou simplement cliquer avec le bouton droit de la souris dans la [vue 3D](#). En outre, il existe une troisième option, plus immédiate, pour modifier le style de navigation directement à partir de l'interface utilisateur située dans la partie inférieure droite de l'écran.



Chacun de ces styles attribue différentes actions aux boutons de la souris ou boutons de souris + combinaison de touches du clavier ou des fonctions à la souris à ces quatre opérations. Le tableau suivant montre les principaux styles disponibles. Les touches du clavier et les boutons de la souris indiqués en magenta doivent être maintenus enfoncés.

Style	Sélection	Zoom	Rotation	Panoramique
Blender				 ou Maj + 
CAD (par défaut)		 ou Ctrl +  + 	 ou  ou Maj + 	 ou Ctrl + 
Gesture				
MayaGesture		 ou Alt + 	Alt + 	Alt + 
OpenCascade		 ou Ctrl + 	 ou Ctrl + 	 ou Ctrl + 
OpenInventou	Maj + 	 ou 		
OpenSCAD		 ou  ou Maj + 	 ou 	
Revit			 ou Maj + 	 ou 
SolidWorks introduced in 1.1		 ou Maj + 		Ctrl + 
TinkerCAD				
Touchpad		Ctrl + Maj + 	Alt +  ou Maj + 	Maj + 

Il convient de noter que lorsqu'un utilisateur survole le menu des styles de navigation situé en bas à droite de l'écran, une infobulle apparaît. Cette infobulle fournit une brève description du style de navigation mis en évidence, offrant ainsi des conseils immédiats sur son utilisation.



Navigation par le clavier

Certaines commandes du clavier sont toujours disponibles et ce, quel que soit le style de navigation :

- **Ctrl** + **+** et **Ctrl** + **-** ou **Pg.Préc** et **Pg.Suiv** pour zoom avant et zoom arrière
- les touches flèches **←** **→** **↑** **↓** pour effectuer un panoramique gauche/droite et haut/bas.
- **Shift** + **←** et **Shift** + **→** pour faire pivoter la vue de 90 degrés.
- le clavier numérique, **0** **1** **2** **3** **4** **5** **6**, pour les sept vues standard :

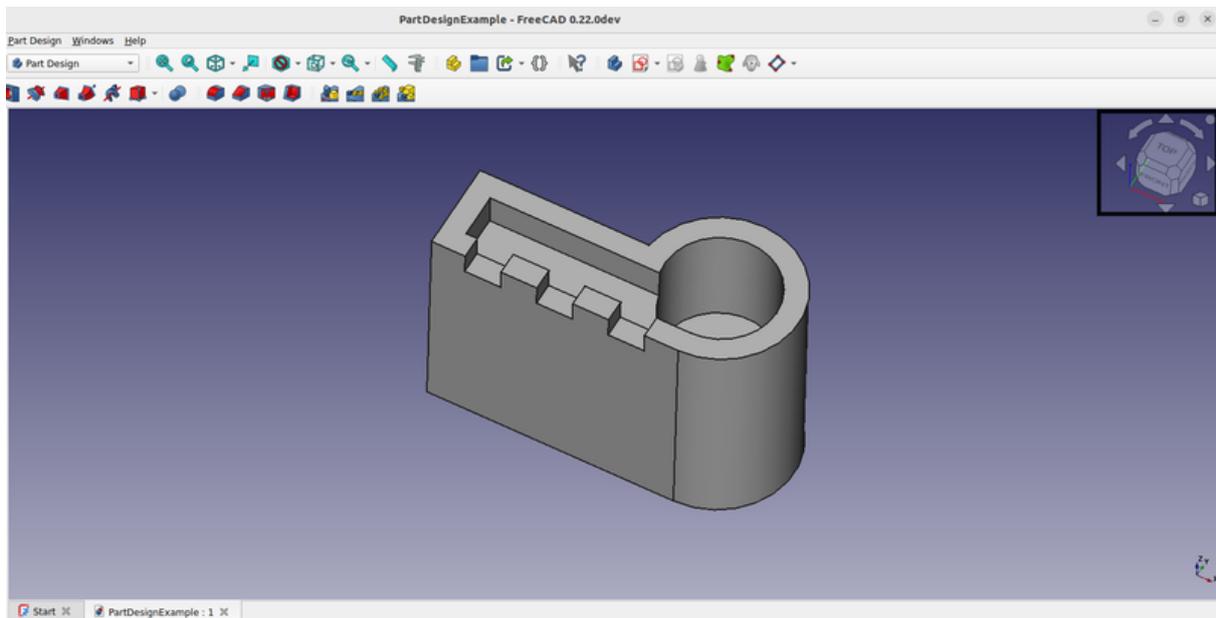
[Isométrique](#), [Avant](#), [Dessus](#), [Droite](#), [Arrière](#), [Dessous](#) et [Gauche](#).

- **V O** mettra la caméra en [mode orthographique](#).
- alors que **V P** mettra la caméra en [mode en perspective](#).
- **Ctrl** vous permettra de sélectionner plus d'un objet ou élément.

Ces contrôles sont également disponibles dans le [menu Affichage](#) et certains dans la barre d'outils Affichage.

Utilisation du cube de navigation

Dans la configuration par défaut, un [cube de navigation](#) se trouve dans le coin supérieur droit de la vue 3D. Il peut être utilisé pour changer de vue.



En cliquant sur une face du cube, la vue passe à cette face. En cliquant sur l'une des quatre flèches triangulaires, la vue pivote de 45 degrés dans la direction indiquée. En cliquant sur l'une des deux flèches courbes, la vue pivote de 45 degrés dans la direction indiquée autour d'une ligne pointant vers vous. En cliquant sur le bouton rond dans le coin supérieur droit du cube, la vue pivote de 180 degrés autour de l'axe vertical de la vue.

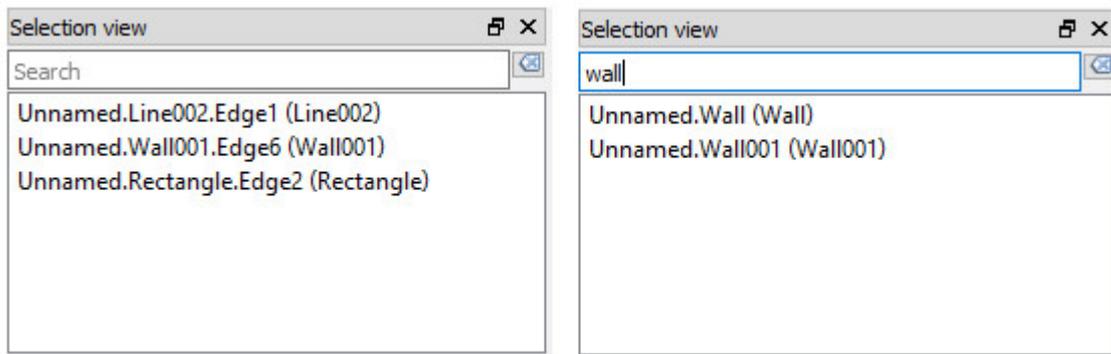
Un mini-cube situé en bas à droite du cube de navigation active un menu déroulant proposant plusieurs options, dont une option permettant de rendre le cube mobile, ce qui permet de déplacer temporairement le cube vers une autre position en le faisant glisser.

Sélection d'objets

Les objets dans la vue 3D peuvent être sélectionnés en cliquant dessus avec la souris (voir détails au-dessus) en fonction du style de navigation.

- Un seul clic sélectionnera l'objet et l'un de ses sous-composants (arête, face, sommet).
- Un double-clic sélectionnera l'objet et tous ses sous-composants.
- Vous pouvez sélectionner plus d'un sous-composant voire différents sous-composants de différents objets en appuyant sur la touche **CTRL**.
- Cliquer avec le bouton de sélection sur une partie vide de la vue 3D désélectionnera tout.

Un panneau appelé "Afficher la Sélection", disponible dans le menu Affichage, peut également être activé. Il vous montrera la sélection en cours :



Vous pouvez également utiliser l'affichage de sélection pour sélectionner des objets en recherchant un objet particulier.

Pour en savoir plus :

- [Les styles de navigation FreeCAD](#)
- [Le cube de navigation](#)

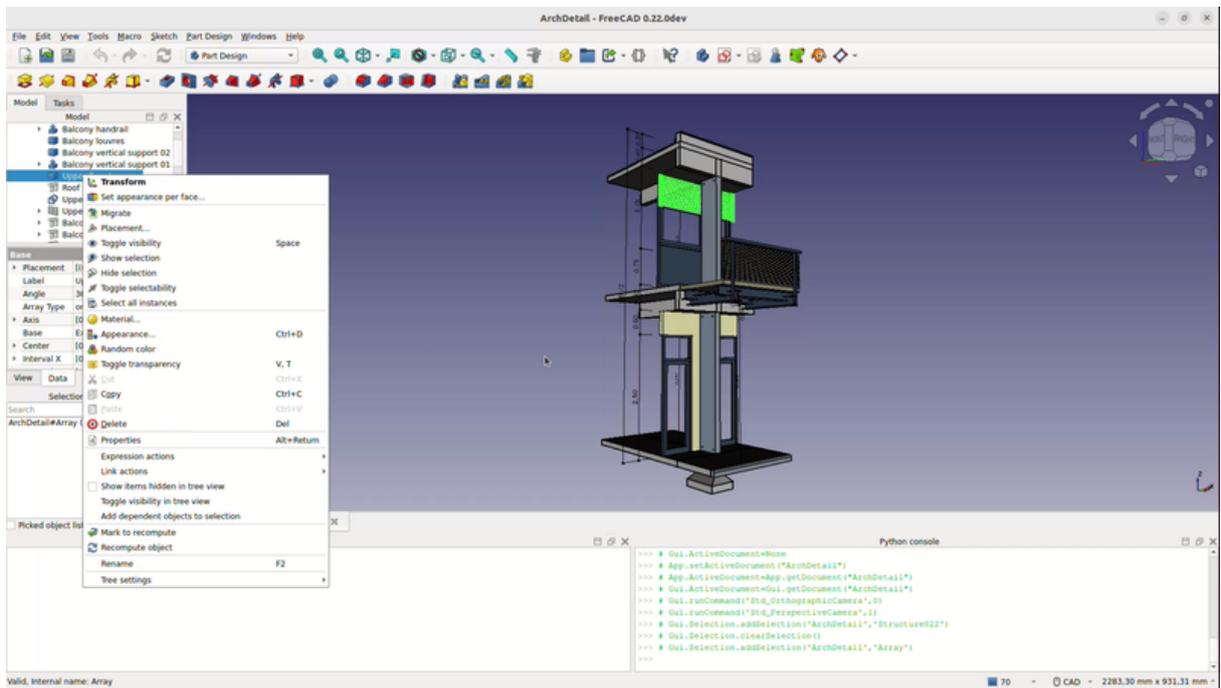
Manuel : Le document FreeCAD

Un document FreeCAD constitue le cœur de votre environnement de conception, encapsulant tous les objets et éléments qui composent votre scène. Il peut contenir une variété d'objets créés dans différents ateliers, ce qui permet une intégration transparente et une grande flexibilité puisque vous pouvez passer d'un atelier à l'autre tout en travaillant dans le même document. Un document est essentiellement le fichier que vous enregistrez sur votre disque et qui contient tout votre travail. FreeCAD vous permet d'ouvrir plusieurs documents simultanément et d'afficher plusieurs vues du même document.

Dans le document, vous pouvez organiser les objets en groupes, chacun portant un nom unique pour en faciliter l'identification et la gestion. Cette gestion des groupes, des objets et de leurs noms s'effectue principalement dans l'arborescence. Les utilisateurs peuvent y créer de nouveaux groupes, y affecter des objets et les supprimer. Des personnalisations plus détaillées, telles que renommer des objets, modifier leurs propriétés visuelles comme la couleur ou ajuster la visibilité, peuvent être effectuées en cliquant avec le bouton droit de la souris sur un objet ou un groupe dans l'arborescence. Des fonctionnalités supplémentaires peuvent également être disponibles en fonction de l'atelier actif.

Les objets dans un document FreeCAD varient considérablement en type, car chaque atelier introduit son propre type d'objets. Par exemple, l'[atelier Mesh](#) est connu pour ajouter des objets de maillage, tandis que l'[atelier Part](#) génère des objets Part. Dans toute session de FreeCAD, si au moins un document est ouvert, ce document est considéré comme actif et est affiché dans la vue 3D courante. Il s'agit du document que vous êtes en train de modifier. Si vous changez d'onglet pour passer à un autre document, l'onglet sélectionné devient le document actif et la plupart des opérations lui seront appliquées.

Les documents FreeCAD sont enregistrés dans le format de fichier .FCStd, qui utilise une structure composée basée sur le format zip, semblable aux formats utilisés par des logiciels tels que LibreOffice. En cas de problème technique, le document peut souvent être décompressé, ce qui permet d'accéder directement à son contenu à des fins de dépannage ou de récupération des données. Cette capacité constitue un filet de sécurité supplémentaire, garantissant que votre travail de conception peut être préservé et récupéré même dans des circonstances inattendues.



Lire plus d'informations

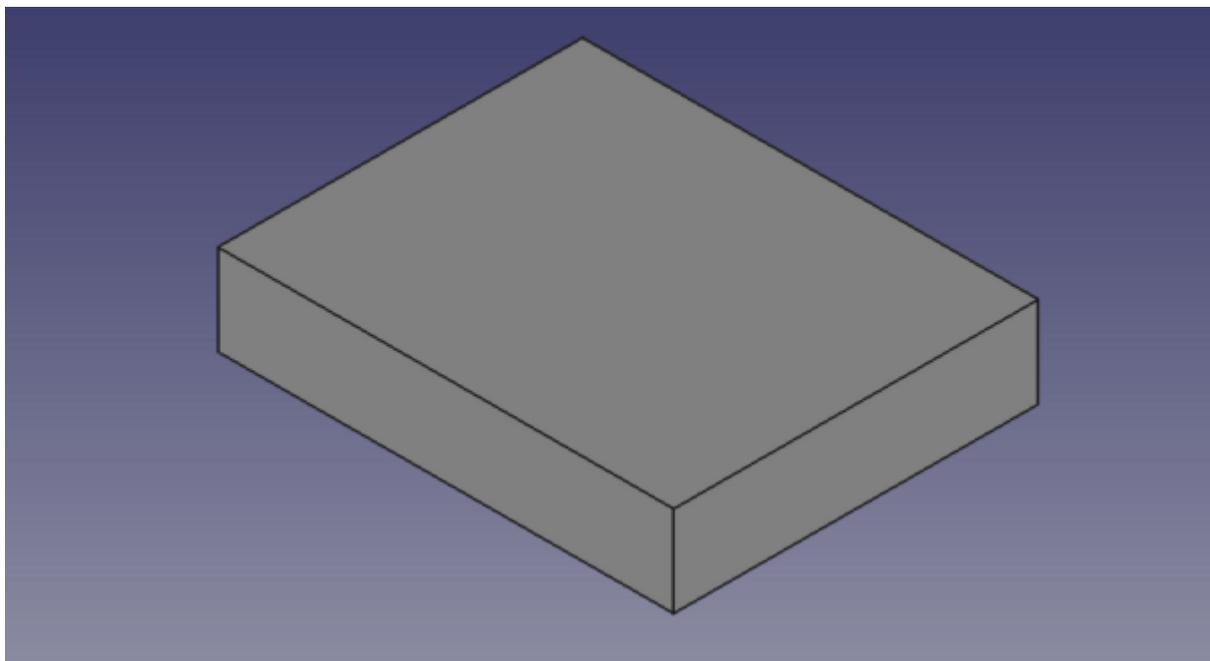
- [Le document FreeCAD](#)
- [Le format de fichier FcStd](#)

Manuel : Objets paramétriques

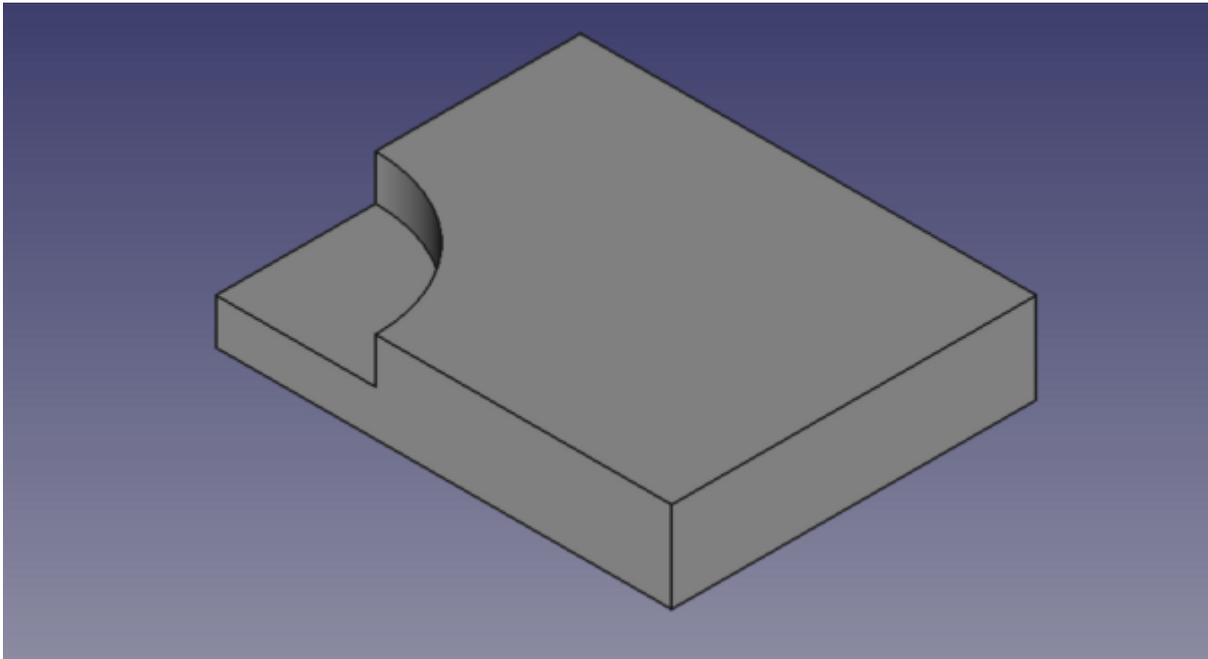
FreeCAD utilise une approche de modélisation paramétrique, où la géométrie des objets est régie par des règles et des paramètres sous-jacents plutôt que d'être sculptée librement. Cela signifie que les dimensions et les caractéristiques de chaque composant sont définies par des paramètres, qui indiquent au programme comment générer la géométrie. Par exemple, pour créer un cylindre, des paramètres tels que le rayon et la hauteur sont spécifiés. Grâce à ces valeurs, FreeCAD génère la forme géométrique précise.

Dans FreeCAD, les objets paramétriques sont essentiellement de petits scripts programmables qui s'exécutent lorsqu'un paramètre est modifié. Ces paramètres peuvent varier considérablement, y compris les nombres entiers et les nombres à virgule flottante, les valeurs dimensionnelles réelles telles que les millimètres, les mètres ou les pieds, les coordonnées (exprimées en x , y , z), les chaînes de texte ou même les références à d'autres objets. Cette polyvalence des paramètres permet de créer des modèles complexes par le biais d'une série d'opérations en chaîne où chaque nouvel objet tire ses caractéristiques d'un objet précédent, tout en introduisant des attributs supplémentaires.

Prenons l'exemple de la création d'un objet cubique solide par modélisation paramétrique. Vous commencez par une forme rectangulaire de base en 2D, appelée « plaque », de longueur l et de largeur w . Cette esquisse définit la base de votre objet cubique. Ensuite, vous définissez une opération d'« extrusion », ou « protrusion », en spécifiant la distance pour pousser ou tirer l'esquisse dans un objet 3D. Il en résulte une forme cubique solide basée sur la forme de l'esquisse et la distance d'extrusion spécifiée.



Sur la face supérieure de la plaque, vous dessinez un cercle d'un diamètre donné d . Vous utilisez ensuite ce cercle pour créer une poche (enlever de la matière) à partir de la plaque d'origine.



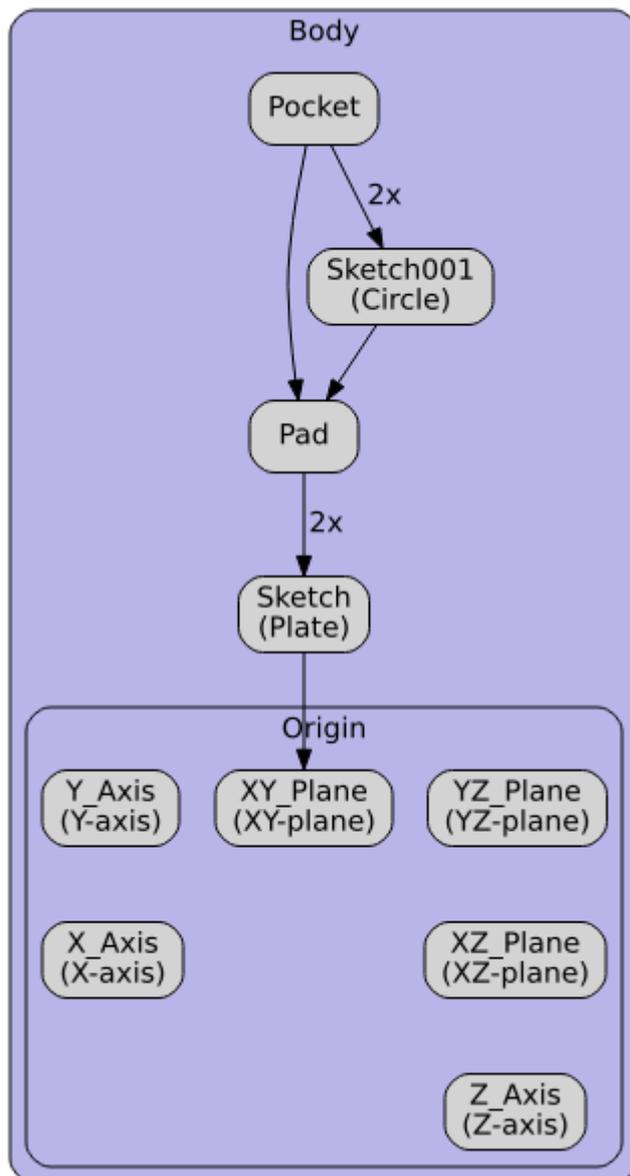
Si vous décidez de modifier l'une des dimensions de la plaque ou du cercle, l'objet final sera également modifié. Grâce à l'utilisation d'une approche de conception paramétrique, il n'est pas nécessaire de refaire l'objet depuis le début.

1. Le recalcul n'est pas toujours automatique. Les opérations lourdes, qui pourraient modifier une grande partie de votre document, et donc prendre un certain temps, ne sont pas exécutées automatiquement. Au lieu de cela, l'objet (et tous les objets qui en dépendent) seront marqués pour le recalcul (une petite icône bleue apparaît sur eux dans l'arborescence). Vous devez alors appuyer sur le bouton de recalcul (ou **Édition** → **Rafraichir**) pour que tous les objets marqués soient recalculés.
2. L'arbre de dépendance doit toujours circuler dans la même direction. Les boucles sont interdites. (Voir [DAG](#), et [vue DAG](#)) Vous pouvez avoir l'objet A qui dépend de l'objet B qui dépend de l'objet C. Mais vous ne pouvez pas avoir l'objet A qui dépend de l'objet B qui dépend de l'objet A. Cela serait une dépendance circulaire. Cependant, vous pouvez avoir beaucoup d'objets qui dépendent du même objet, par exemple, les objets B et C dépendent tous deux de A. Le menu **Outils** → **graphique de dépendance** vous montre un diagramme de dépendance comme sur l'image ci-dessus. Ça peut être utile pour détecter les problèmes.

Dans le processus de modélisation paramétrique de FreeCAD, l'examen de l'arborescence de dépendance d'un objet donne un aperçu clair de la construction séquentielle et des relations au sein d'un modèle. Dans l'exemple ci-dessus, la structure repose sur l'« esquisse de plaque », qui sert de base à la forme initiale du modèle. L'opération « Protrusion » est ensuite appliquée pour ajouter de la matière à cette esquisse fondatrice, créant ainsi une structure tridimensionnelle à partir de la base bidimensionnelle.

Ensuite, une « esquisse de cercle » est dessinée sur la surface formée. Ce cercle constitue la base de l'opération « Cavité » suivante. Cette opération consiste à retirer stratégiquement de la matière de la structure, c'est-à-dire à découper une partie sur la base de l'esquisse du cercle. Ce processus d'ajout puis de retrait de matière permet d'intégrer des géométries et des caractéristiques complexes dans le modèle de base de manière transparente.

Grâce à cette séquence d'opérations - en partant de l'esquisse de base, en ajoutant du volume à l'aide d'une protrusion et en créant des fonctions détaillées à l'aide d'esquisses et de cavités supplémentaires - l'objet final prend forme. Chaque étape de cette chaîne dépend de la précédente, illustrant la nature interconnectée de la conception paramétrique dans FreeCAD.



Tous les objets ne sont pas paramétriques dans FreeCAD. Souvent, les géométries que vous importez d'autres fichiers ne contiennent aucun paramètre et seront des objets simples et non paramétriques. Cependant, ceux-ci peuvent souvent être utilisés comme base ou point de départ pour les objets paramétriques nouvellement créés, en fonction, bien sûr, de ce que requiert l'objet paramétrique, et de la qualité de la géométrie importée.

Tous les objets, cependant, paramétriques ou non, auront quelques paramètres de base, comme un Nom, qui est unique dans le document et ne peut pas être édité, un label, qui est un nom défini par l'utilisateur et qui peut être édité, et un emplacement ([placement](#)) qui définit sa position dans l'espace 3D.

Enfin, il convient de noter que les objets personnalisables paramétriques sont [faciles à programmer en Python](#).

Lire plus d'informations

- [L'éditeur de propriétés](#)
- [Comment programmer des objets paramétriques](#)
- [Positionnement des objets dans FreeCAD](#)
- [Graphe des dépendances](#)

Manuel : Importer et exporter vers d'autres types de fichiers

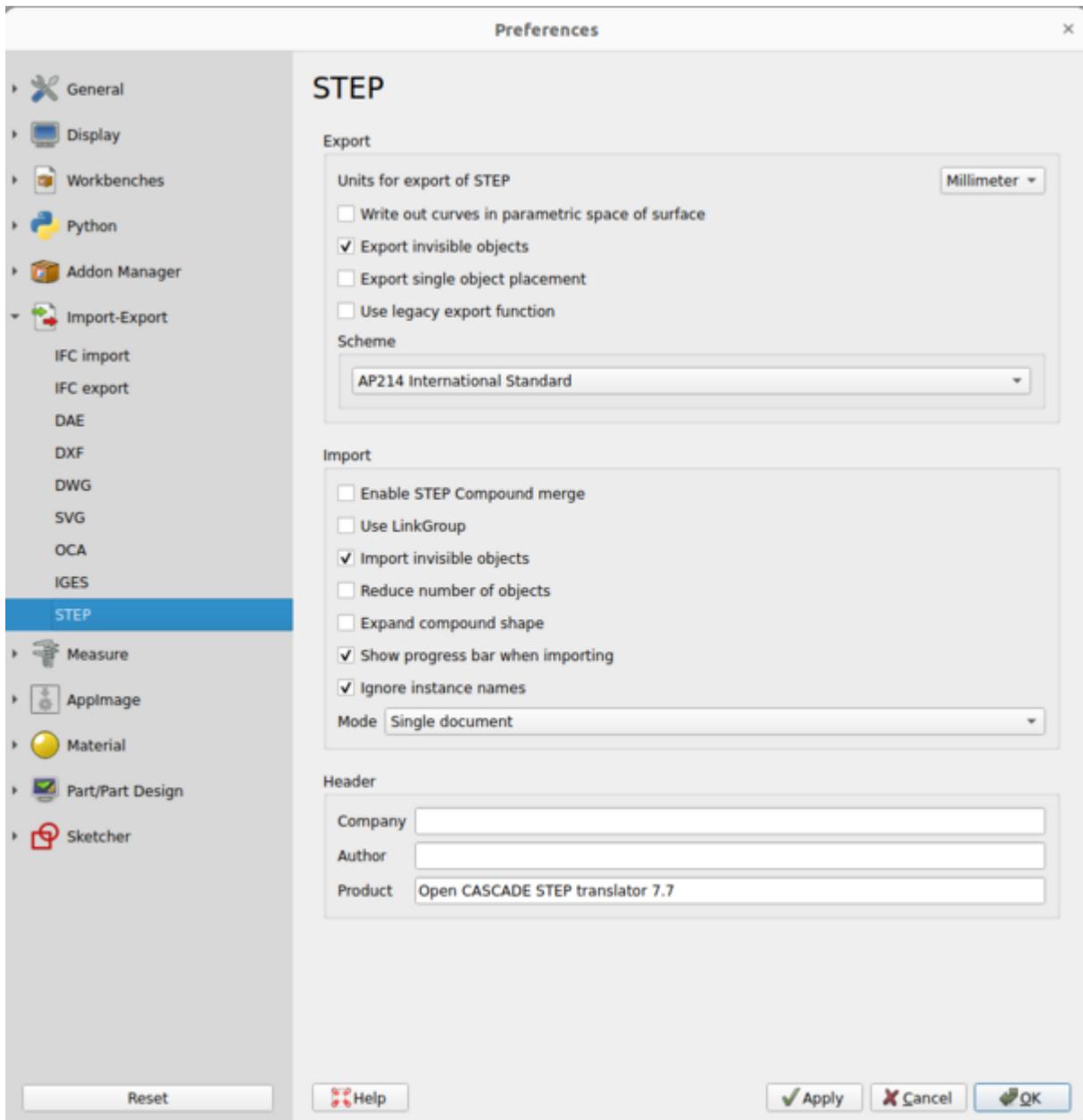
FreeCAD peut importer et exporter vers plusieurs types de fichiers. Voici une liste des plus importants avec une brève description des caractéristiques disponibles :

Format	Import	Export	Notes
STEP	Oui	Oui	C'est le format d'importation/exportation le plus fidèle disponible, car il supporte la géométrie solide et NURBS. Utilisez le chaque fois que cela est possible.
IGES	Oui	Oui	Un format solide ancien, également très bien pris en charge. Certaines anciennes applications ne supportent pas STEP mais supportent IGES.
BREP	Oui	Oui	Le format natif d' OpenCasCade , le noyau de géométrie FreeCAD.
DXF	Oui	Oui	Un format ouvert maintenu par Autodesk. Depuis la 3D les données contenues dans un fichier DXF sont encodées dans un format propriétaire, FreeCAD ne peut importer/exporter que des données 2D à partir de ce format.
DWG	Oui	Oui	La version propriétaire de DXF. Nécessite l'installation de l'utilitaire Teigha File Converter . Ce format souffre des mêmes limitations que DXF.
OBJ	Oui	Oui	Un format basé sur un maillage. Ne peut contenir que des mailles triangulaires. Tous les objets solides et basés sur NURBS de FreeCAD seront convertis en maillage à l'exportation. Un exportateur alternatif est fourni par l'atelier Arch, plus adapté à l'exportation des modèles d'architecture.

DAE	Oui	Oui	Le principal format d'importation/exportation de Sketchup. Ne peut contenir que des maillages triangulés. Tous les objets solides et NURBS de FreeCAD seront convertis en maillage lors de l'exportation.
STL	Oui	Oui	Un format basé sur le maillage, couramment utilisé pour l'impression 3D. Ne peut contenir que des maillages triangulés. Tous les objets solides et NURBS de FreeCAD seront convertis en maillage lors de l'exportation.
PLY	Oui	Oui	Un ancien format de maillage. Peut seulement contenir des mailles triangulaires. Tous les objets solides et basés sur NURBS de FreeCAD seront convertis en maillage à l'exportation.
IFC	Oui	Oui	Classes . Nécessite l'installation de IfcOpenShell-python . Le format IFC et sa compatibilité avec d'autres applications est une affaire complexe, à utiliser avec précaution.
SVG	Oui	Oui	Un excellent format graphique 2D
VRML	Oui	Oui	Un format Web assez ancien.
GCODE	Oui	Oui	FreeCAD peut déjà importer et exporter vers/ depuis plusieurs variantes de GCode, mais seulement un petit nombre de machines sont actuellement prises en charge.

CSG	Oui	Non	Format Géométrie de construction de solides d'OpenSCAD (Constructive Solid Geometry en anglais).
-----	-----	-----	--

Certains de ces formats de fichiers ont des options. Celles-ci peuvent être configurées à partir du menu Édition → Préférences → Importer/Exporter :



Lire plus d'informations

- [Tous les formats de fichiers pris en charge par FreeCAD](#)
- [Travailler avec des fichiers DXF dans FreeCAD](#)
- [Travailler avec des fichiers SVG dans FreeCAD](#)
- [Importation et exportation vers le format IFC](#)
- [OpenCasCade](#)
- [Teigha File Converter](#)

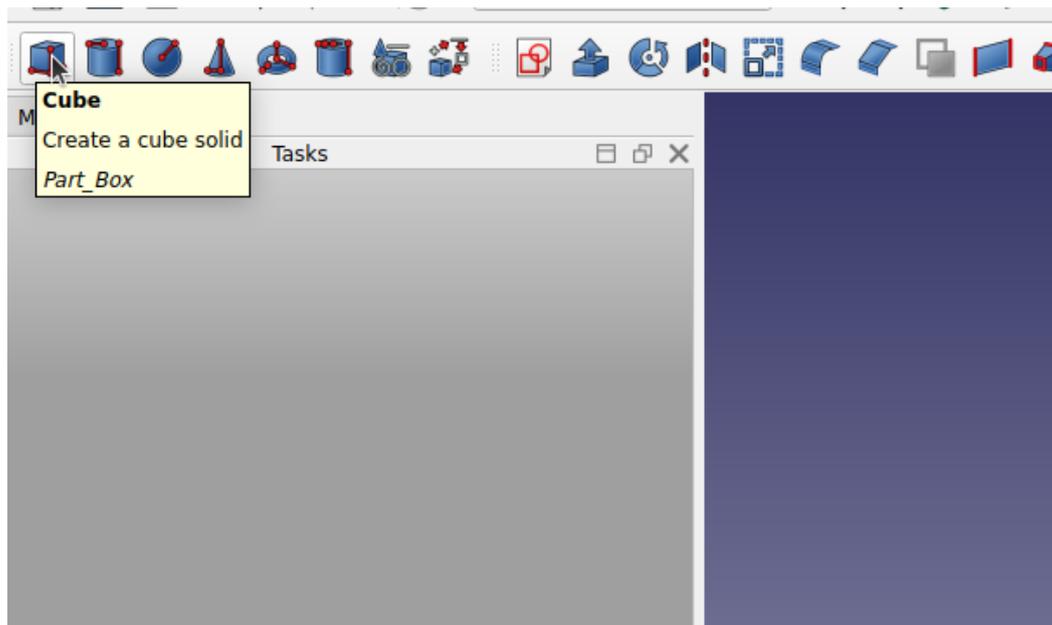
- [Base de données des spécifications IFC](#)
- [IfcOpenShell](#)

Manuel : Tous les ateliers en un coup d'œil

Comme indiqué précédemment, FreeCAD propose plusieurs ateliers, chacun dédié à des applications différentes. Bien que la multitude d'options puisse sembler écrasante au premier abord, chaque atelier est conçu pour répondre à des tâches spécifiques, ce qui rend le flux de travail global plus efficace et adapté aux exigences des différents projets. Par exemple, l'atelier PartDesign est idéal pour créer et modifier des modèles 3D solides, tandis que l'atelier Draft est parfait pour le dessin et la mise en plan 2D. Cette approche modulaire permet aux utilisateurs de personnaliser leur interface et leurs outils en fonction de leurs besoins et préférences spécifiques.

Sur cette page, vous trouverez des informations concernant l'ensemble des ateliers de base et leurs fonctionnalités. Pour plus d'informations, n'hésitez pas à vous référer à chaque page de l'[atelier](#) de la documentation de FreeCAD pour une liste plus complète.

Une fonction intéressante de FreeCAD est la possibilité d'obtenir des informations supplémentaires en passant la souris sur une commande. Cette fonction d'infobulle aide les utilisateurs à comprendre ce que fait chaque commande, en leur fournissant des conseils et en facilitant l'apprentissage et la navigation dans le logiciel.



Quatre ateliers sont également conçus pour fonctionner par paires, l'un étant entièrement intégré à l'autre : BIM contient tous les outils de Draft, et PartDesign comprend tous les outils de Sketcher. Toutefois, par souci de clarté, ils sont séparés ci-dessous.

Atelier Part

L'[atelier Part](#) offre des outils fondamentaux pour travailler avec des pièces solides, y compris des primitives comme les cubes et les sphères, ainsi que des opérations géométriques et booléennes de base. Servant de lien principal avec [OpenCasCade](#),

l'atelier Part constitue la pierre angulaire du système géométrique de FreeCAD, la quasi-totalité des autres ateliers générant une géométrie basée sur les pièces. Ce système de modélisation paramétrique permet de contrôler et de modifier avec précision les modèles 3D grâce à un flux de travail basé sur l'historique. Les utilisateurs peuvent construire et affiner des conceptions complexes en empilant et en ajustant des formes et des opérations plus simples, ce qui garantit un processus de conception robuste et flexible.

Outil	Description	Outil	Description
 Boîte	Crée une boîte	 Cylindre	Crée un cylindre
 Sphère	Crée une sphère	 Cône	Crée un cône
 Tore	Crée un tore (anneau)	 Tube	Crée un tube
 Créer des primitives	Crée diverses autres primitives géométriques paramétriques	 Générateur de formes	Crée des formes plus complexes à partir de primitives
 Extrusion	Extrude les faces planes d'un objet	 Révolution	Crée un solide en faisant tourner un autre objet (non solide) autour d'un axe.
 Objet en miroir	Miroir de l'objet sélectionné sur un plan de miroir donné	 Échelle	Met à l'échelle une forme sélectionnée
 Congé	Crée des congés (arrondis) sur les arêtes d'un objet	 Chanfrein	Chanfreine les arêtes d'un objet
 Créer une face	Crée une face à partir d'une esquisse	 Surface réglée	Créer une surface délimitée entre les courbes sélectionnées
 Lissage	Lisse d'un profil à l'autre	 Balayage	Balaye un ou plusieurs profils le long d'une trajectoire
 Section	Crée une section en intersectant un objet avec un plan de section	 Coupes	Crée plusieurs coupes transversales le long d'un objet
 Décaler en 3D	Crée une copie à l'échelle de l'objet original	 Évider	Attribue une épaisseur aux faces d'une forme
 Projeter sur une surface	Projette sur une face	 Couleur par face	Définit une couleur pour chaque face/surface d'un objet
 Composé	Crée un composé de plusieurs formes	 Éclater le composé	Divise un composé de formes
 Filtre de composé	Filtre les objets composés sur la base d'un paramètre (par exemple, volume, surface, autre objet).	 Opération booléenne	Exécute une opération booléenne sur deux objets sélectionnés
 Soustraction	Coupe (soustrait) un objet d'un autre	 Union	Fusionne (réunit) les objets sélectionnés en une seule pièce

 Intersection	Extrait la partie commune (intersection) de deux objets	 Connecter	Crée un raccord d'objets avec connexion interne
 Intégrer	Intègre un objet creux dans un autre objet creux	 Découper	Crée une découpe dans la paroi d'un objet creux pour ajouter un autre objet creux
 Fragments booléens	Calcule tous les fragments qui peuvent résulter de l'application d'opérations booléennes entre les formes d'entrée	 Séparer/exploser	Divise les formes par intersection avec d'autres formes
 Scinder	Divise les formes par intersection avec d'autres formes	 OU exclusif	Supprime la géométrie partagée par un nombre pair d'objets
 Vérifier la géométrie	Vérifie l'absence d'erreurs dans la géométrie	 Supprimer la fonction	Supprime des fonctions d'une forme

Atelier PartDesign

L'[atelier PartDesign](#) dans FreeCAD est un outil essentiel pour la création et la modification de modèles 3D solides. Il permet aux utilisateurs de concevoir des pièces complexes en esquissant des profils 2D, puis en appliquant diverses opérations telles que des extrusions, des lissages et des révolutions pour générer une géométrie 3D. Cet atelier prend également en charge la création de fonctions telles que les poches, les trous, les congés et les chanfreins, offrant ainsi un ensemble complet d'outils pour la conception détaillée de pièces. En outre, l'atelier PartDesign s'intègre parfaitement à l'atelier Sketcher, ce qui permet aux utilisateurs de définir et de contraindre les esquisses qui servent de base aux modèles 3D. Cette intégration facilite une approche paramétrique de la conception, permettant des ajustements et des mises à jour faciles du modèle tout au long du processus de conception.

Outil	Description	Outil	Description
 Protrusion	Extrude un objet solide à partir d'une esquisse sélectionnée.	 Révolution	Crée un solide en faisant tourner une esquisse autour d'un axe.
 Lissage additif	Crée un solide en effectuant une transition entre deux ou plusieurs esquisses.	 Balayage additif	Crée un solide en balayant une ou plusieurs esquisses le long d'une trajectoire ouverte ou fermée.
 Hélice additive	Crée un solide en balayant une esquisse le long d'une hélice.	 Cube additif	Crée un cube additif.
 Cylindre additif	Crée un cylindre additif.	 Sphère additive	Crée une sphère additive.
 Cône additif	Crée un cône additif.	 Ellipsoïde additif	Crée un ellipsoïde additif.
 Tore additif	Crée un tore additif.	 Prisme additif	Crée un prisme additif.
 Pyramide tronquée additive	Crée une pyramide tronquée additive.	 Cavité	Crée une cavité à partir d'une esquisse sélectionnée.
 Perçage	Crée une fonction de trou à partir d'une esquisse de cercle sélectionnée.	 Rainure	Crée une rainure en faisant tourner une esquisse autour d'un axe.
 Lissage soustractif	Crée une forme solide en effectuant une transition entre deux ou plusieurs esquisses et la soustrait du corps actif.	 Balayage soustractif	Crée une forme solide en balayant une ou plusieurs esquisses le long d'une trajectoire ouverte ou fermée et la soustrait du corps actif.
 Subtractive helix	Creates a solid shape by sweeping a sketch along a helix and subtracts it from the active body	 Subtractive box	Adds a subtractive box to the active body
 Hélice soustractive	Ajoute un cylindre soustractif au corps actif.	 Sphère soustractive	Ajoute une sphère de soustraction au corps actif.
 Cône soustractif	Ajoute un cône de soustraction au corps actif.	 Ellipsoïde soustractif	Ajoute un ellipsoïde soustractif au corps actif.

 Tore soustractif	Ajoute un tore soustractif au corps actif.	 Prisme soustractif	Ajoute un prisme soustractif au corps actif.
 Pyramide tronquée soustractive	Ajoute une pyramide tronquée soustractive au corps actif.	 Opérations booléennes	Effectue des opérations booléennes sur les objets sélectionnés.
 Congé	Ajoute des congés (arrondis) aux arêtes du corps actif.	 Chanfrein	Chanfreine les arêtes du corps actif.
 Dépouille	Applique une dépouille angulaire aux faces d'un objet.	 Évidement	Crée une coque épaisse à partir du corps actif et ouvre la face sélectionnée.
 Symétrie	Crée une symétrie de fonction par rapport à un plan ou une face.	 Répétition linéaire	Crée une répétition linéaire de fonctions.
 Répétition circulaire	Crée une répétition circulaire de fonctions.	 Transformation multiple	Permet de créer une répétition avec n'importe quelle combinaison des autres transformations.

Atelier Draft

L' [atelier Draft](#) dans FreeCAD est conçu pour créer et éditer des dessins en 2D. Il offre une gamme d'outils pour le dessin, y compris des lignes, des arcs, des cercles et du texte. Les utilisateurs peuvent également effectuer des opérations telles que le découpage, l'extension et le décalage. L'atelier de dessin est particulièrement utile pour les dessins d'architecture et peut être utilisé pour créer des géométries 2D complexes qui peuvent ensuite être transformées en modèles 3D. Il s'intègre parfaitement aux autres ateliers et constitue une base polyvalente pour les projets de conception en 2D et en 3D.

Outil	Description	Outil	Description
 Ligne	Dessine un segment de droite entre 2 points.	 Polyligne	Dessine une ligne composée de plusieurs segments de ligne (polyligne).
 Congé	Crée un congé (coin arrondi) ou un chanfrein (arête droite) entre deux lignes.	 Arc	Dessine un segment d'arc à partir du centre, du rayon, de l'angle de départ et de l'angle d'arrivée.
 Arc par 3 points	Crée un arc de cercle à partir de trois points qui définissent sa circonférence.	 Cercle	Dessine un cercle à partir du centre et du rayon.
 Ellipse	Dessine une ellipse à partir de deux points des coins.	 Rectangle	Dessine un rectangle à partir de 2 points opposés.
 Polygone	Dessine un polygone régulier à partir d'un centre et d'un rayon.	 B-spline	Dessine une B-spline à partir d'une série de points.
 Courbe de Bézier	Dessine une courbe de Bézier à partir d'une série de points.	 Courbe de Bézier cubique	Crée une courbe de Bézier composée de segments cubiques.
 Point	Insère un seul point.	 Surface liée	Crée un nouvel objet à partir de faces sélectionnées sur des objets existants.
 Forme à partir d'un texte	Insère une forme composée représentant une chaîne de texte à un point donné du document en cours.	 Hachures	Crée des hachures sur les faces planes d'un objet sélectionné.
 Texte	Crée une annotation sur plusieurs lignes.	 Dimension	Crée une dimension.
 Étiquette	Crée une étiquette.	 Éditer le style des annotations	Permet de définir des styles qui affectent les propriétés visuelles des objets de type annotation.
 Déplacer	Déplace ou copie des objets d'un endroit à un autre.	 Pivoter	Fait pivoter les objets d'un certain angle autour d'un point.
 Échelle	Met à l'échelle des objets par rapport à un point.	 Miroir	Crée des copies miroir par rapport à une ligne.

 Décaler	Décale un objet à une certaine distance.	 Ajouter ou prolonger	Ajuste, étend ou extrude un objet.
 Étirer	Étire les objets en déplaçant les points sélectionnés.	 Cloner	Crée des copies liées d'objets.
 Réseau orthogonal	Crée un réseau orthogonal à partir d'un objet sélectionné.	 Réseau polaire	Crée un réseau à partir d'un objet sélectionné en plaçant des copies le long d'une circonférence.
 Réseau circulaire	Crée un réseau à partir d'un objet sélectionné en plaçant des copies le long de circonférences concentriques.	 Réseau selon une courbe	Crée un réseau à partir d'un objet sélectionné en plaçant des copies le long d'une courbe.
 Réseau lié selon une courbe	Fonctionne de la même manière que Réseau selon une courbe mais crée un réseau de liens au lieu d'un réseau normal.	 Réseau de points	Crée un réseau à partir d'un objet sélectionné en plaçant des copies aux points d'un composé de points.
 Réseau lié selon des points	Fonctionne de la même manière que Réseau selon une courbe mais crée un réseau de liens au lieu d'un réseau normal.	 Réseau torsadé selon une courbe	Crée des copies torsadées le long d'une courbe.
 Réseau lié torsadé selon une courbe	Crée des copies liées torsadées le long d'une courbe.	 Éditer	Modifie le modèle actif.
 Surligner des sous éléments	Met temporairement en évidence les objets sélectionnés ou les objets de base des objets sélectionnés.	 Joindre	Réunit des Draft Lignes et des Draft Polyignes en un seul objet.
 Scinder	Divise une Draft Ligne ou une Draft Polyligne en un point ou un arête spécifié.	 Agréger	Transforme ou joint des objets en un objet de niveau supérieur.
 Désagréger	Transforme ou sépare les objets en objets de niveau inférieur.	 Vue 2D d'une forme	Crée un objet 2D qui est une vue aplatie d'un autre objet.
 Polyligne <=> B-spline	Convertit des Draft Polyignes en Draft B-splines et vice versa.	 Draft <=> Esquisse	Convertit un objet Draft en Esquisse et vice-versa.

 Pente	Incline les Draft Lignes ou les Draft Polyignes sélectionnées.	 Inverser le texte de la dimension	Fait pivoter le texte des dimensions de Draft Dimensions sélectionnées de 180° autour de la ligne de cote.
 Vue 2D d'une forme	Crée des projections 2D à partir d'objets sélectionnés.	 Verrouiller l'aimantation	Active ou désactive l'aimantation globalement.
 Aimantation Extrémité	S'aimante aux extrémités des arêtes.	 Aimantation Milieu	S'aimante au point médian des arêtes.
 Aimantation Centre	S'aimante au centre des faces et des arêtes circulaires.	 Aimantation Angle	S'aimante aux points cardinaux spéciaux des arêtes circulaires, à des multiples de 30° et 45°.
 Aimantation Intersection	S'aimante à l'intersection de deux arêtes.	 Aimantation Perpendiculaire	S'aimante aux points perpendiculaires des faces et des arêtes.
 Aimantation Extension	S'aimante à une ligne imaginaire qui s'étend au-delà des extrémités des arêtes droites.	 Aimantation Parallèle	S'aimante à une ligne imaginaire parallèle aux arêtes droites.
 Aimantation Spécial	S'aimante à des points spéciaux définis par l'objet.	 Aimantation Au plus proche	S'aimante au point le plus proche sur les faces et les arêtes.
 Aimantation Orthogonal	S'aimante aux lignes imaginaires qui traversent le point précédent à des multiples de 45°.	 Aimantation Grille	S'aimante aux intersections des lignes de la grille.
 Aimantation Plan de travail	Projette les points d'aimantation sur le plan de travail en cours.	 Aimantation Dimensions	Affiche les dimensions X et Y provisoires.

Atelier Sketcher

L'[atelier Sketcher](#) contient des outils pour créer et modifier des objets 2D complexes, appelés des esquisses. La géométrie contenue dans ces esquisses peut être positionnée avec précision et liée à l'aide de contraintes. Elles sont principalement conçues pour être les blocs de construction de la géométrie de PartDesign, mais elles sont utiles partout dans FreeCAD.

Outil	Description		
 Point	Dessine un point.	 Polyligne	Dessine une ligne composée de plusieurs segments de ligne.
 Ligne	Dessine un segment de droite à partir de 2 points.	 Arc par centre	Dessine un segment d'arc à partir du centre, du rayon, de l'angle de départ et de l'angle d'arrivée.
 Arc par 3 points	Dessine un segment d'arc à partir de deux points d'extrémité et d'un autre point sur la circonférence.	 Arc d'ellipse	Crée un arc d'ellipse.
 Arc d'hyperbole	Crée un arc d'hyperbole.	 Arc de parabole	Crée un arc de parabole.
 Cercle par centre	Dessine un cercle à partir du centre et du rayon.	 Cercle par 3 points	Dessine un cercle à partir de trois points de la circonférence.
 Ellipse par centre	Dessine une ellipse par le point central, le point de rayon principal et le point de rayon secondaire.	 Ellipse par 3 points	Dessine une ellipse de diamètre majeur (2 points) et de rayon mineur (2 points).
 Rectangle	Dessine un rectangle à partir de 2 points opposés.	 Rectangle centré	Crée un rectangle centré.
 Rectangle arrondi	Crée un rectangle arrondi.	 Triangle équilatéral	Dessine un triangle régulier inscrit dans un cercle de géométrie de construction.
 Carré	Dessine un carré régulier inscrit dans un cercle de géométrie de construction.	 Pentagone	Dessine un pentagone régulier inscrit dans un cercle de géométrie de construction.
 Hexagone	Dessine un hexagone régulier inscrit dans un cercle de géométrie de construction.	 Heptagone	Dessine un heptagone régulier inscrit dans un cercle de géométrie de construction.
 Octogone	Dessine un octogone régulier inscrit dans un cercle de géométrie de construction.	 Polygone régulier	Crée un polygone régulier. Le nombre de côtés peut être spécifié.
 Contour oblong	Crée un contour oblong.	 Rainure en arc	Crée une rainure en arc.

 B-spline simple	Crée une courbe B-spline par des points de contrôle.	 B-spline périodique	Crée une courbe B-spline périodique (fermée) par des points de contrôle.
 B-spline par des nœuds	Crée une courbe B-spline par des points de nœuds.	 B-spline périodique par des nœuds	Crée une courbe B-spline périodique (fermée) par des points de nœuds.
 Géométrie de construction	Bascule d'un élément vers/depuis le mode construction.	 Contrainte de dimension	Sur la base de la sélection en cours, il offre des contraintes dimensionnelles/ géométriques appropriées.
 Contrainte de distance en X	Fixe la distance horizontale entre deux points ou les extrémités d'une ligne.	 Contrainte de distance en Y	Fixe la distance verticale entre deux points ou les extrémités d'une ligne.
 Contrainte de distance	Fixe la longueur d'une ligne, la distance entre deux points, la distance perpendiculaire d'un point à une ligne, la distance entre les bords de deux cercles/arcs ou entre un cercle/arc et une ligne, ou la longueur d'un arc.	 Contrainte auto rayon/diamètre	Fixe le rayon des arcs et des cercles de poids B-spline, ainsi que le diamètre des cercles.
 Contrainte rayon	Fixe le rayon des cercles, des arcs et des cercles de poids B-spline.	 Contrainte diamètre	Fixe le diamètre des cercles et des arcs.
 Contrainte angulaire	Fixe l'angle entre deux arêtes, l'angle d'une ligne avec l'axe horizontal de l'esquisse ou l'angle d'ouverture d'un arc de cercle.	 Contrainte fixe	Contraint l'élément sélectionné en définissant des distances verticales et horizontales par rapport à l'origine, verrouillant ainsi l'emplacement de cet élément.
 Contrainte de coïncidence unifiée	Crée une contrainte de coïncidence entre des points, fixe des points sur des arêtes ou des axes, ou crée une contrainte concentrique..	 Contrainte horizontale/verticale	Contraint les lignes ou les paires de points à être horizontales ou verticales, selon ce qui est le plus proche de l'alignement en cours.

 Contrainte horizontale	Contraint les lignes ou les éléments d'une polyligne sélectionnés à une véritable orientation horizontale. Plusieurs objets peuvent être sélectionnés avant d'appliquer cette contrainte.	 Contrainte verticale	Contraint les lignes ou les éléments d'une polyligne sélectionnés à une véritable orientation verticale. Plusieurs objets peuvent être sélectionnés avant d'appliquer cette contrainte.
 Contrainte parallèle	Contraint les lignes à être parallèles.	 Contrainte perpendiculaire	Contraint deux lignes à être perpendiculaires.
 Contrainte tangente	Crée une contrainte de tangence entre deux entités sélectionnées, ou une contrainte de colinéarité entre deux segments de ligne.	 Contrainte d'égalité	Contraint deux entités sélectionnées à être égales l'une à l'autre. Si elle est utilisée sur des cercles ou des arcs, leurs rayons seront égaux.
 Contrainte de symétrie	Contraint deux points à être symétriques autour d'une ligne ou d'un axe, ou autour d'un troisième point.	 Contrainte de blocage	Bloque les arêtes en place avec une seule contrainte. Il est principalement destiné aux B-splines.
 Congé	Crée un congé entre deux arêtes non parallèles.	 Chanfrein	Crée un chanfrein entre deux arêtes non parallèles.
 Ajuster	Ajuste une arête aux intersections les plus proches avec d'autres arêtes.	 Diviser	Divise une arête en transférant la plupart des contraintes.
 Prolonger	Prolonge ou raccourcit une ligne ou un arc jusqu'à un endroit arbitraire, ou jusqu'à un arête ou un point cible.	 Géométrie externe	Projette sur le plan de l'esquisse des arêtes et/ou des sommets appartenant à des objets situés en dehors de l'esquisse.
 Projection	Projette sur le plan de l'esquisse des arêtes et/ou des sommets appartenant à des objets situés en dehors de l'esquisse.	 Intersection	Crée l'intersection de faces et/ou d'arêtes appartenant à des objets situés en dehors de l'esquisse avec le plan de l'esquisse.
 Copie carbone	Copie toutes les géométries et contraintes d'une autre esquisse dans l'esquisse active.	 Sélectionner l'origine	Sélectionne l'origine de l'esquisse.
 Axe horizontal	Sélectionne l'axe horizontal de l'esquisse.	 Axe vertical	Sélectionne l'axe vertical de l'esquisse.

 Déplacer/dupliquer	Déplace ou crée éventuellement des copies des éléments sélectionnés.	 Pivoter/dupliquer	Fait pivoter ou crée éventuellement des copies pivotées des éléments sélectionnés.
 Mise à l'échelle	Met à l'échelle ou crée éventuellement des copies à l'échelle des éléments sélectionnés.	 Décaler	Crée des arêtes équidistantes autour des arêtes sélectionnées.
 Symétriser	Crée des copies en miroir des éléments sélectionnés.	 Supprimer l'alignement des axes	Supprime l'alignement des axes des arêtes sélectionnées en remplaçant les contraintes horizontales et verticales par les contraintes parallèles et perpendiculaire .

Atelier BIM

L' [atelier BIM](#) (une combinaison des anciens ateliers BIM, Native-IFC et Arch) est conçu pour la conception architecturale et la planification de la construction, offrant des outils pour créer et manipuler des éléments de construction paramétriques tels que des murs, des dalles, des colonnes, des poutres, des toits et des fenêtres. Il prend en charge la norme IFC (Industry Foundation Classes) pour un échange de données transparent avec d'autres logiciels BIM, facilitant ainsi la collaboration. Les utilisateurs peuvent générer des dessins 2D détaillés, des annotations et des documents de construction à partir de modèles 3D, en s'intégrant étroitement à l'atelier Draft pour des tâches telles que la création de plans d'étage, de sections et d'élévations.

Outil	Description	Outil	Description
 Site	Crée un site comprenant les objets sélectionnés	 Bâtiment	Crée un bâtiment comprenant les objets sélectionnés
 Espace	Crée un objet espace dans le document	 Niveau	Crée un niveau incluant les objets sélectionnés
 Mur	Crée un mur à partir de zéro ou en utilisant un objet sélectionné comme base	 Mur-rideau	Crée un mur-rideau à partir de zéro ou en utilisant un objet sélectionné comme base
 Colonne	Crée une colonne à l'emplacement spécifié	 Poutre	Crée une poutre entre deux points
 Dalle	Crée une dalle à partir d'une forme plane	 Porte	Place une porte à un endroit donné
 Fenêtre	Crée une fenêtre en utilisant un objet sélectionné comme base	 Conduite	Crée une conduite
 Raccord	Crée un angle ou une connexion en T entre 2 ou 3 conduites sélectionnées	 Escalier	Crée un objet escalier dans le document
 Toit	Crée un toit à pentes à partir d'une face sélectionnée	 Panneau	Crée un objet panneau à partir d'un objet 2D sélectionné
 Ossature	Crée un objet ossature à partir d'une mise en page sélectionnée	 Clôture	Crée une clôture qui construit une clôture en répétant un poteau et une section de clôture le long d'un tracé donné
 Treillis	Crée un treillis à partir d'une ligne sélectionnée ou à partir de zéro	 Équipement	Crée un objet d'équipement ou de mobilier
 Armature personnalisée	Crée une armature sur la base d'une esquisse ou d'une face sélectionnée.	 Profilé	Crée un profilé 2D paramétrique
 Boîte	Crée une boîte générique	 Bibliothèque	Ouvre la bibliothèque d'objets
 Composant	Ajoute un composant architectural non défini	 Référence externe	Crée un objet de référence externe

 Dimension alignée	Crée une dimension alignée	 Dimension horizontale	Crée une dimension horizontale
 Dimension verticale	Crée une dimension verticale	 Ligne de référence	Crée une polyligne avec une flèche à son extrémité
 Axes	Ajoute un réseau d'axes à 1 direction	 Système d'axes	Ajoute un système d'axes composé de plusieurs axes
 Grille	Ajoute un objet de type grille	 Plan de coupe	Ajoute un objet plan de coupe
 Hachures	Crée des hachures sur la face sélectionnée	 Vue	Crée une vue de dessin technique à partir d'un plan de coupe ou d'un objet 2D
 Matériau	Définit ou crée un matériau pour l'objet sélectionné	 Nomenclature	Créer différents types de nomenclature pour collecter des données à partir du modèle

Autres ateliers intégrés

Bien que ce qui précède résume les outils les plus importants de FreeCAD, beaucoup d'autres ateliers sont disponibles, parmi lesquels :

- L'[atelier TechDraw](#) pour produire des dessins techniques à partir de modèles 3D.
- L'[atelier Mesh](#) permet de travailler avec des maillages polygonaux ([maillages polygonaux](#)). Bien que les maillages ne soient pas le type de géométrie préféré pour travailler dans FreeCAD, en raison de leur manque de précision et de prise en charge des courbes, les maillages ont encore de nombreuses utilisations et sont entièrement pris en charge dans FreeCAD. L'atelier Mesh offre également un certain nombre d'outils de conversion de pièces en maillages et de maillages en pièces.
- L'[atelier Spreadsheet](#) permet la création et la manipulation de données de tableur, qui peuvent être extraites des modèles FreeCAD. Les cellules de feuille de calcul peuvent également être référencées dans de nombreux domaines de FreeCAD, ce qui permet de les utiliser comme structures de données de base.
- L'[atelier FEM](#) traite de l'analyse par éléments finis ([Méthode des éléments finis](#)) et permet d'effectuer des calculs de résistance pré et post-traitement et d'afficher les résultats graphiquement.

Ateliers externes

Un certain nombre d'autres ateliers très utiles produits par les membres de la communauté FreeCAD existent également. Bien qu'ils ne soient pas inclus dans une

installation standard de FreeCAD, ils sont faciles à installer en tant que plug-ins. Ils sont tous référencés dans le dépôt [FreeCAD-addons](#). Parmi les plus développés se trouvent les suivants :

- L'[atelier Drawing Dimensioning](#) offre de nombreux nouveaux outils pour travailler directement sur les feuilles de dessin et vous permet d'ajouter des dimensions, des annotations et d'autres symboles techniques avec un grand contrôle sur leur aspect. **L'extension Drawing Dimensioning n'est plus géré.**
- L'[atelier Fasteners](#) offre une large gamme d'éléments de fixation prêts à insérer comme des vis, des boulons, des tiges, des rondelles et des écrous. De nombreuses options et paramètres sont disponibles.
- L'[atelier A2plus](#) offre une série d'outils pour réaliser et travailler avec des [assemblages](#).

Lire plus d'informations

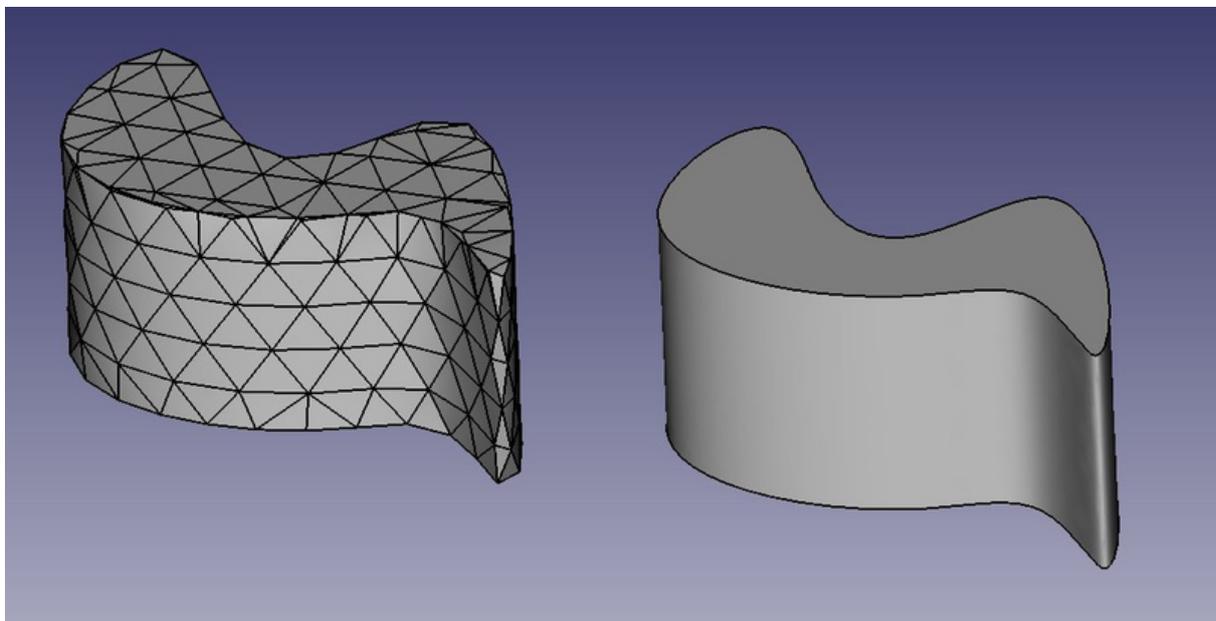
- [Liste complète des ateliers](#)
- [Atelier Part](#)
- [Atelier Draft](#)
- [Atelier PartDesign](#)
- [Atelier BIM](#)
- [Atelier TechDraw](#)
- [Atelier FEM](#)
- [Le dépôt des extensions de FreeCAD](#)

Manuel : Modélisation traditionnelle, la voie CSG

La [géométrie de construction de solides](#) (CSG en anglais : *Constructive Solid Geometry*) décrit la manière fondamentale de travailler avec la géométrie solide en 3D. Elle consiste à créer des objets complexes en ajoutant ou en retirant des pièces de solides à l'aide d'opérations booléennes telles que l'union, la soustraction ou l'intersection.

Comme nous l'avons vu plus haut dans ce manuel, FreeCAD prend en charge différents types de géométrie. Cependant, le type préféré et le plus pratique pour la conception d'objets 3D du monde réel dans FreeCAD est la géométrie solide [BREP](#), principalement gérée par l'atelier Part. BREP définit les objets 3D en spécifiant leurs limites spatiales. Les aspects clés de la BREP comprennent : les faces (éléments de surface de l'objet), les arêtes (lignes de démarcation où deux faces se rencontrent) et les sommets (points où les arêtes convergent).

La technique BREP présente plusieurs avantages. Tout d'abord, elle définit les surfaces à l'aide d'équations mathématiques, ce qui permet une modélisation précise et exacte. Cette précision est cruciale pour les applications d'ingénierie où des dimensions exactes sont requises. En outre, la technique BREP fournit des surfaces lisses et détaillées, contrairement au [maillage polygonal](#) qui approxime les surfaces courbes avec des facettes. Ceci est similaire à la différence entre les images vectorielles, qui évoluent sans perte de qualité, et les images bitmap, qui peuvent apparaître pixellisées lorsqu'elles sont agrandies. BREP conserve des informations topologiques complètes sur l'objet, y compris les relations entre les faces, les arêtes et les sommets, ce qui est essentiel pour les opérations complexes telles que les calculs booléens et le filtrage.



À gauche une représentation du maillage et à droite une représentation du BREP

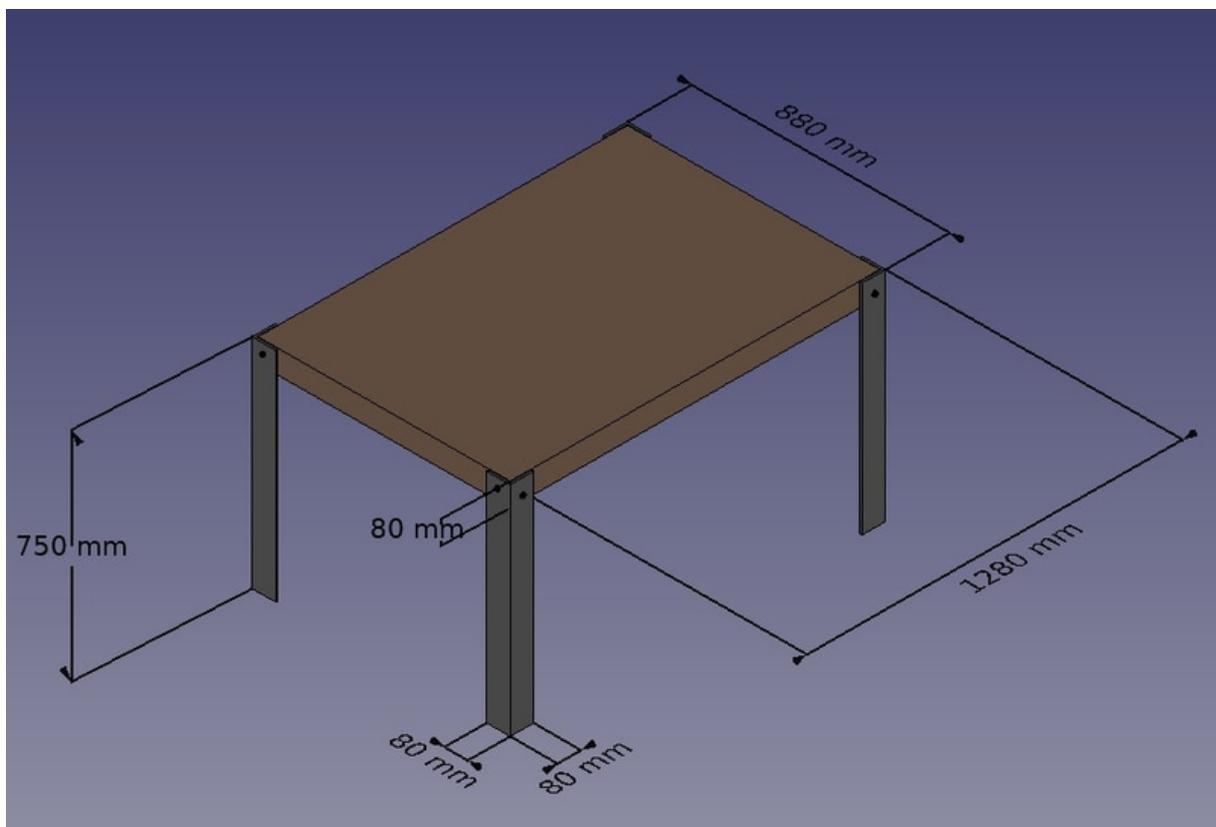
Les maillages polygonaux sont constitués de sommets, d'arêtes et de faces formant des triangles ou des quadrilatères. Les maillages sont plus simples et plus rapides pour le rendu, mais manquent de précision. Lorsqu'elles sont agrandies ou imprimées à grande échelle, les mailles présentent des surfaces à facettes plutôt

que des courbes lisses. En revanche, la BREP utilise des courbes et des surfaces définies mathématiquement, ce qui lui confère une précision et une douceur supérieures. Les modèles BREP sont préférables pour les applications de CAO où la précision est demandée.

Dans FreeCAD, la géométrie basée sur la technique BREP est gérée par [OpenCasCade](#), une bibliothèque de logiciels à code ouvert. L'interface principale entre FreeCAD et le noyau OpenCasCade est l'atelier Part, qui sert de base à la plupart des autres ateliers, en fournissant des outils essentiels pour créer et manipuler des objets BREP. L'atelier Part comprend des outils permettant de créer des primitives, telles que des formes de base comme des boîtes, des cylindres et des sphères. Il comporte également des outils permettant d'effectuer des opérations booléennes telles que la fusion, l'intersection et la soustraction de formes, ainsi que des outils permettant de déplacer, de faire pivoter, de mettre à l'échelle et de cloner des objets.

Bien que d'autres ateliers de FreeCAD, tels que les ateliers PartDesign et Surface, offrent des outils plus avancés pour construire et manipuler la géométrie, ils s'appuient sur l'atelier Part sous-jacent. Il est utile de comprendre le fonctionnement interne des objets Part et de maîtriser les outils de base de Part. Souvent, ces outils plus simples permettent de résoudre des problèmes que des outils plus complexes ne peuvent pas traiter efficacement.

Pour illustrer l'utilisation de l'atelier Part, nous modéliserons cette table en utilisant uniquement des opérations CSG (à l'exception des vis, pour lesquelles nous utiliserons l'une des extensions, et les dimensions qui seront vues au chapitre suivant) :



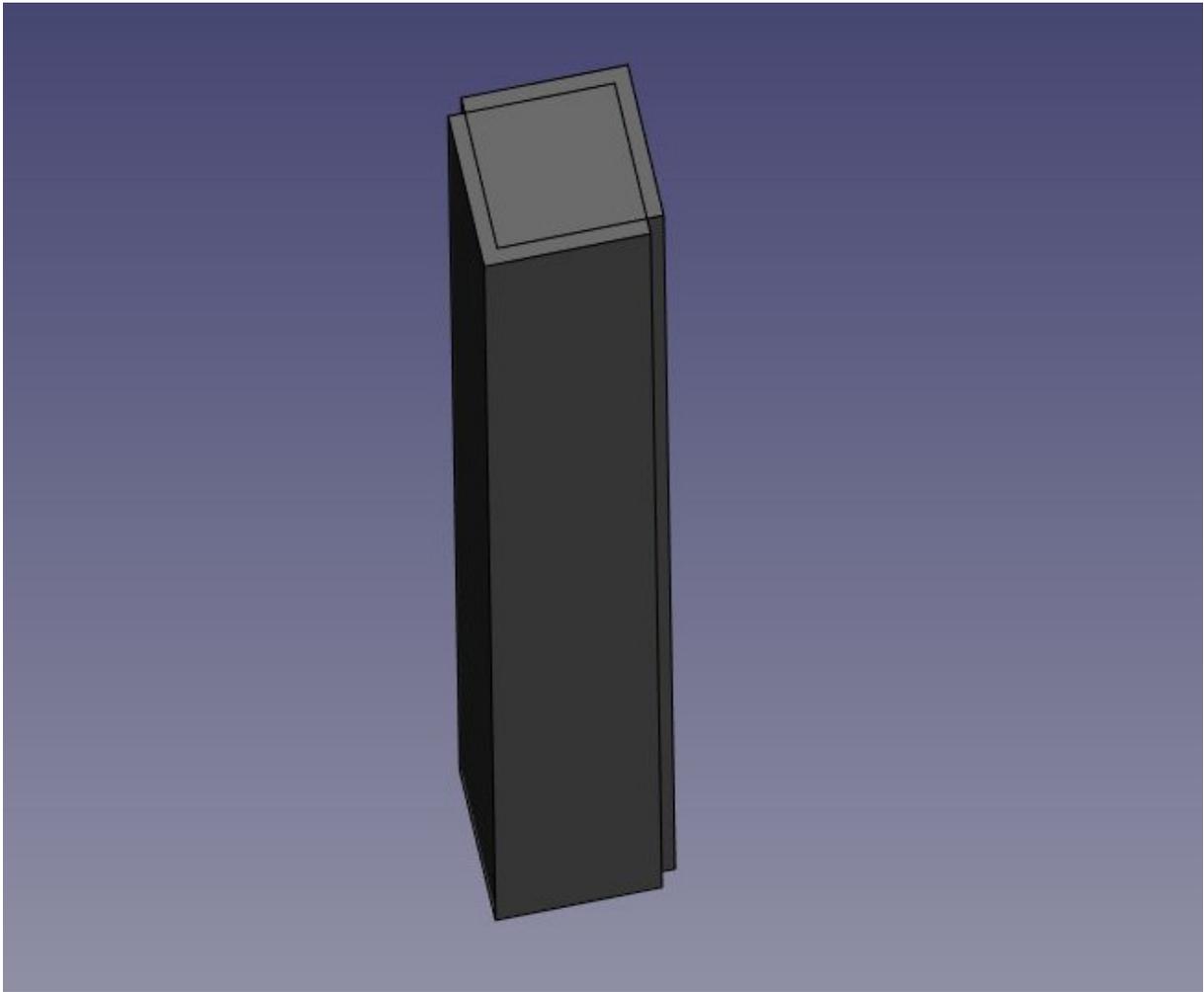
Créons un nouveau document (**Ctrl+N** ou menu **Fichier** → **Nouveau**) pour la conception de notre table. Le document est initialement appelé « Nouveau » dans

l'onglet Modèle du panneau Vue combinée. Si vous enregistrez le document (**Ctrl+Maj+S** ou menu **Fichier** → **Enregistrer sous**) en tant que nouveau document FreeCAD appelé « table.FCStd », le document sera renommé « table », ce qui permet d'identifier plus clairement le projet. Nous utiliserons le millimètre (mm) comme unité de longueur. N'hésitez pas à changer d'unité en utilisant le menu situé dans le coin inférieur droit, selon votre préférence.

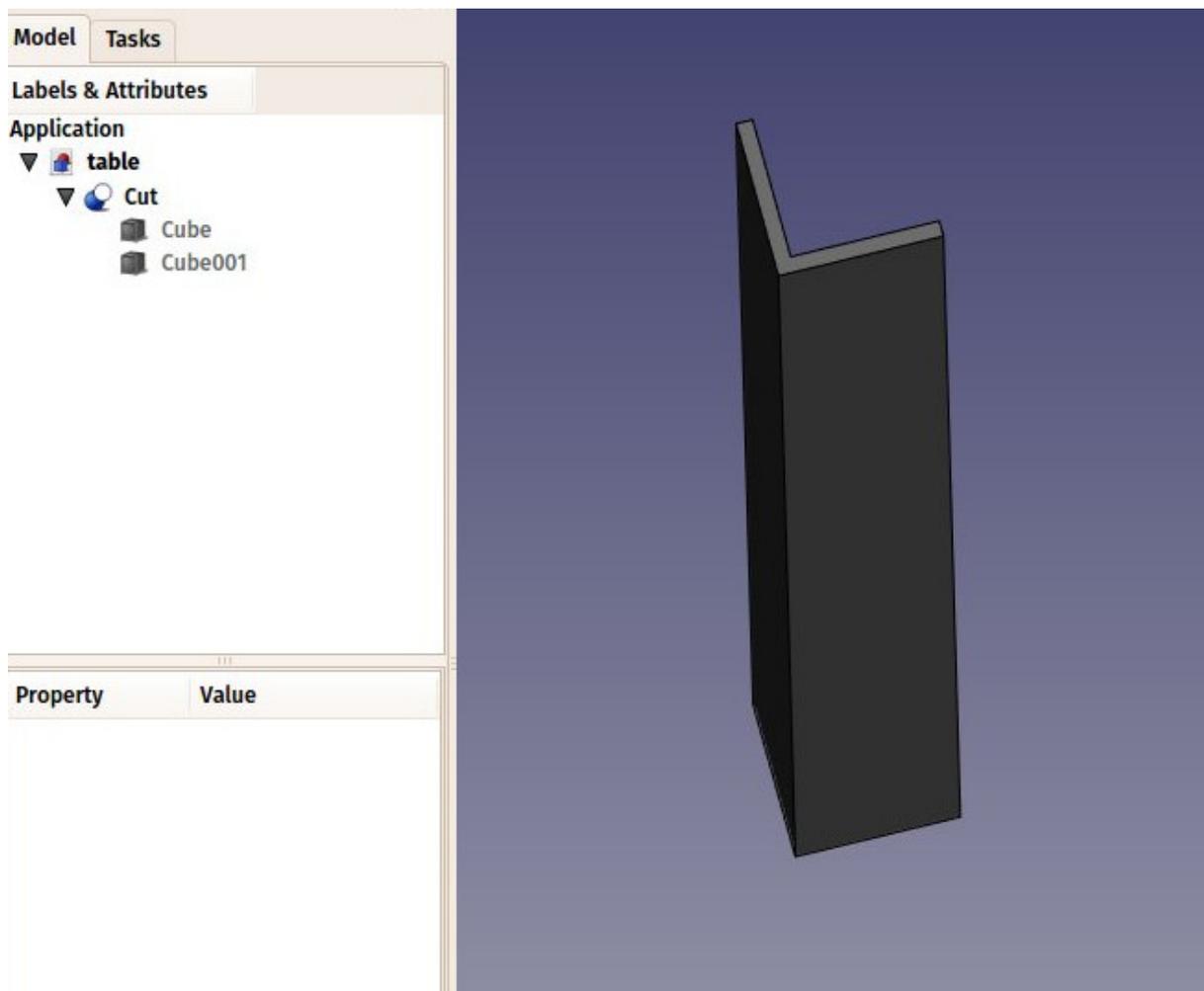
Nous pouvons maintenant passer à l'atelier Part et créer notre premier pied de table.

- Appuyez sur le bouton  **Cube**
- Sélectionnez le cube obtenu, puis définissez les propriétés suivantes (onglet **Données**) :
 - Longueur : 80 mm
 - Largeur : 80 mm
 - Hauteur : 750 mm
- Dupliquez le cube en appuyant sur **Ctrl+C** puis **Ctrl+V** (ou menu **Édition** → **Copier et Coller**). Vous ne verrez aucun changement dans la vue 3D, car le deuxième objet se superpose au premier. Vous pouvez voir que le nouveau cube a été collé parce que l'arborescence montre maintenant un objet appelé « Cube001 ».
- Sélectionnez Cube001 dans l'arborescence
- Modifiez la position de Cube001 en modifiant sa propriété Placement dans l'onglet Données (cliquez sur la flèche située à côté de Position pour la développer) :
 - Position x : 8 mm
 - Position y : 8 mm

Vous devriez maintenant voir deux grands cubes, l'un décalé de 8 mm par rapport à l'autre dans les directions X et Y :



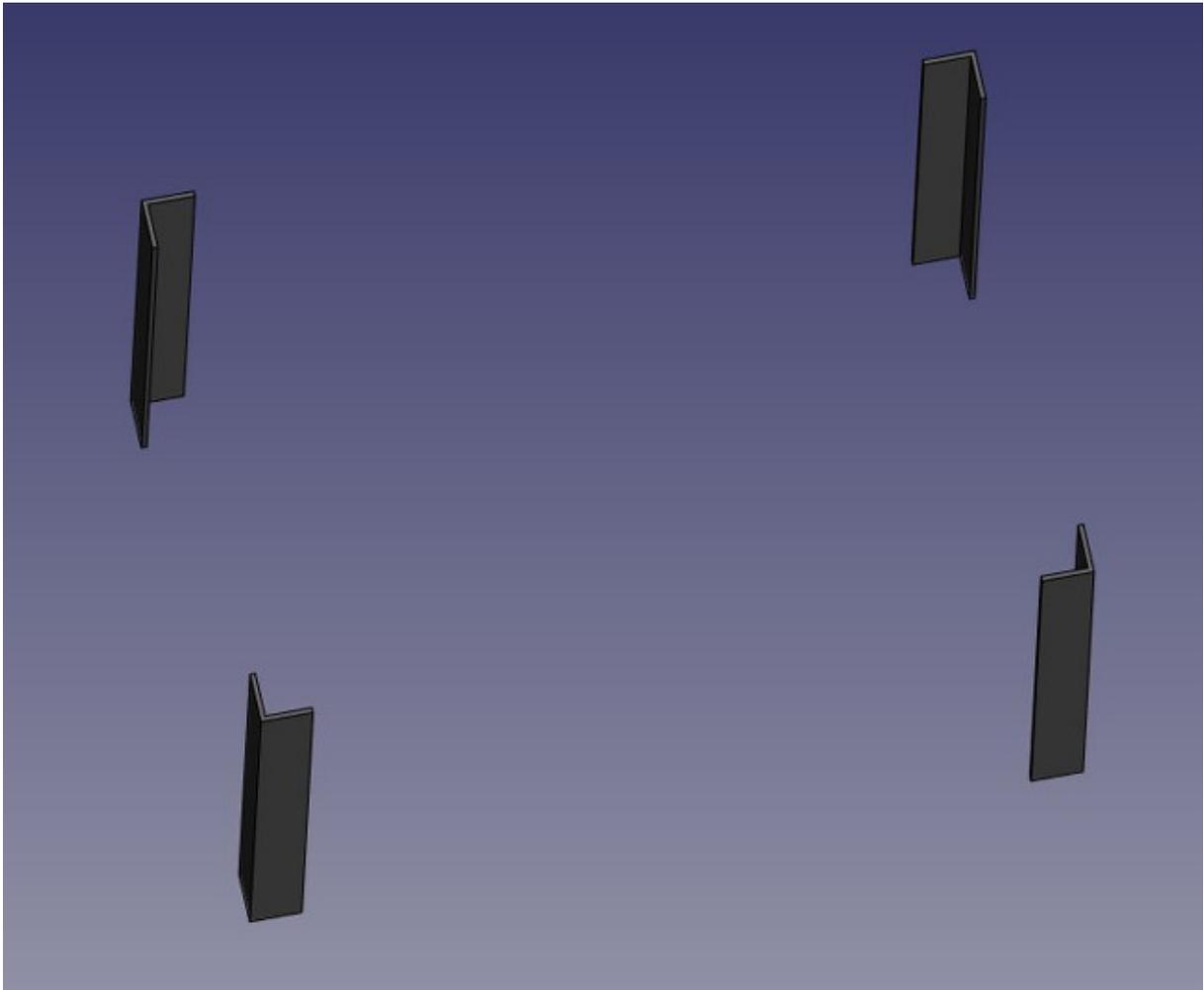
- Nous pouvons maintenant soustraire un cube de l'autre pour obtenir notre pied de table en forme de L : sélectionnez le cube d'origine, dont il restera une partie après l'opération Couper. Ensuite, avec la touche Ctrl enfoncée, sélectionnez Cube001, qui sera soustrait du premier. Notez que l'ordre de sélection détermine le résultat de l'opération Couper. Appuyez sur le bouton  **Soustraction** :



L'objet nouvellement créé, appelé « Cut », contient les deux cubes que nous avons utilisés comme opérandes. En fait, les deux cubes sont toujours dans le document et ont simplement été cachés et regroupés sous l'objet Cut dans l'arborescence. Vous pouvez toujours les sélectionner en développant la flèche située à côté de l'objet Cut. Si vous le souhaitez, vous pouvez les rendre à nouveau visibles en cliquant sur les icônes en forme d'œil situées à côté de leurs étiquettes d'objet, dans leurs menus de clic droit ou en modifiant leurs propriétés.

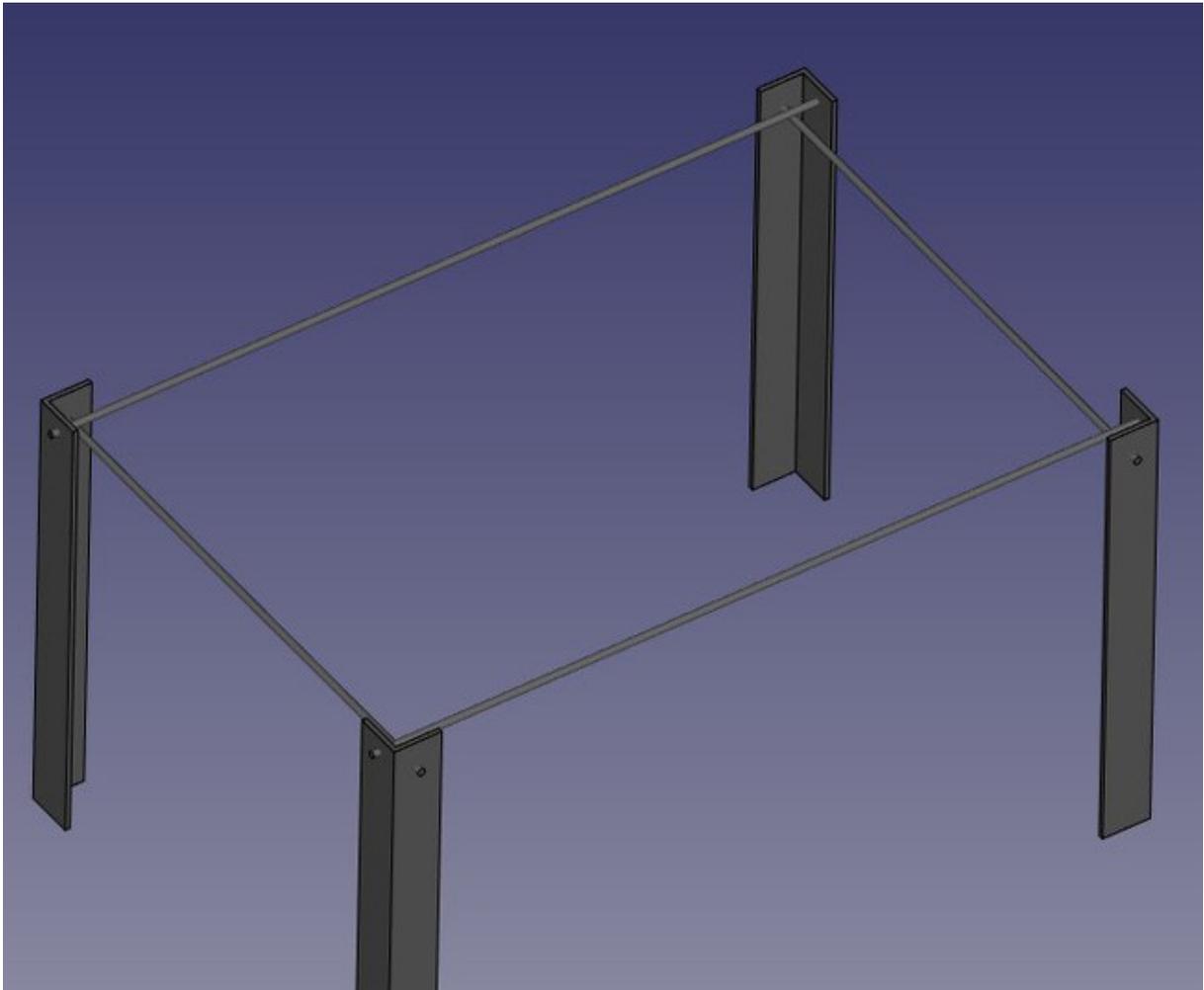
Vous pouvez également effectuer des coupes et d'autres opérations booléennes avec l'outil [Part Opération booléenne](#).

- Maintenant, créons les trois autres pieds de table en faisant six copies supplémentaires de notre cube original. Comme il est toujours copié dans le presse-papiers, vous pouvez simplement le coller (Ctrl+V) 6 fois. Modifiez la position de chaque nouveau cube comme suit :
 - Cube002: x: 0, y: 800 mm
 - Cube003: x: 8 mm, y: 792 mm
 - Cube004: x: 1200 mm, y: 0
 - Cube005: x: 1192 mm, y: 8 mm
 - Cube006: x: 120 mm, y: 800 m
 - Cube007: x: 1192 mm, y: 792 mm
- Effectuons maintenant trois opérations de coupe supplémentaires en sélectionnant d'abord le cube « hôte », puis le cube à couper. Nous avons maintenant quatre objets Cut :



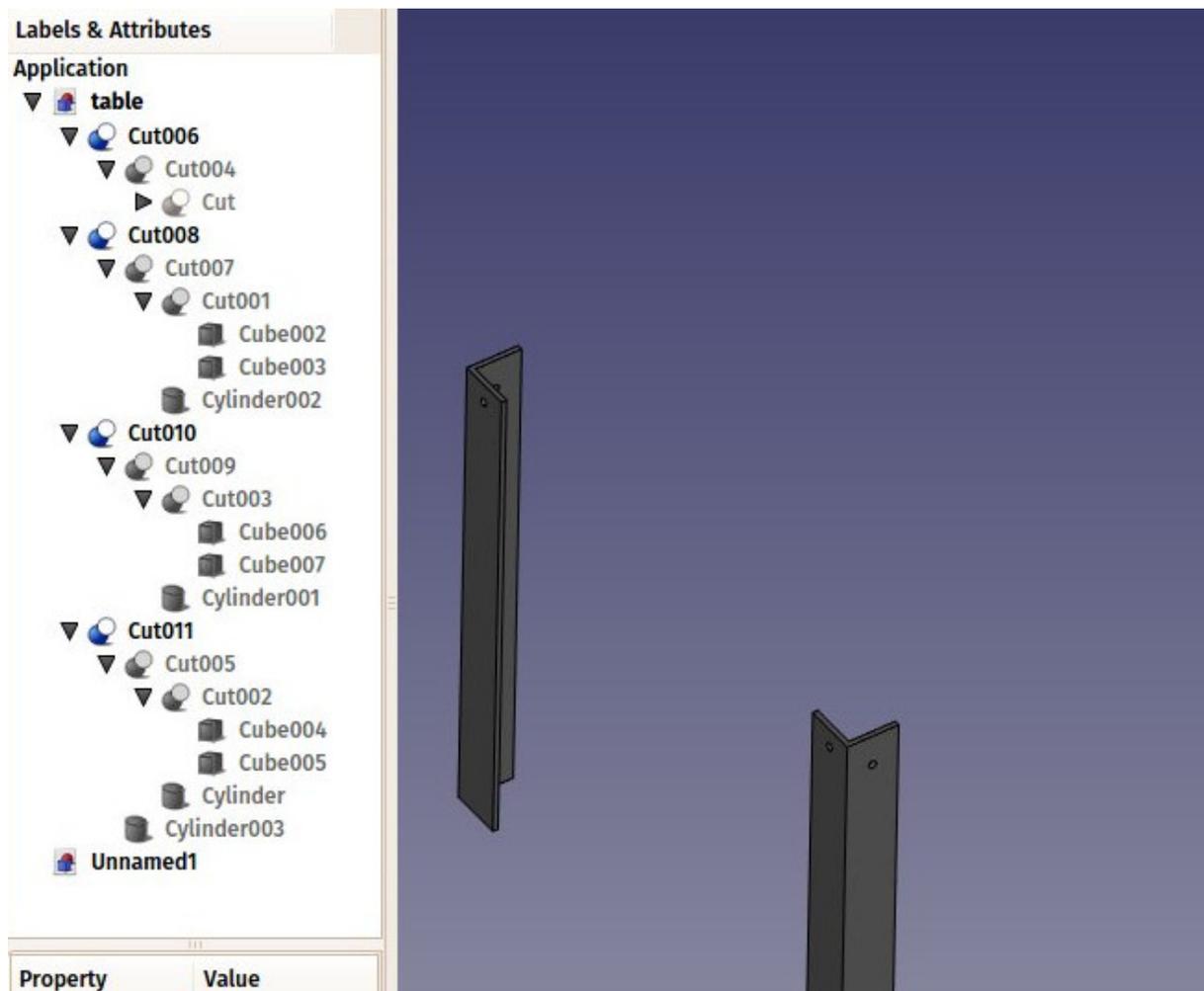
Au lieu de dupliquer le cube de base six fois, nous aurions pu dupliquer le pied complet trois fois en copiant/collant l'objet Cut créé ci-dessus à partir de nos deux premiers cubes et en faisant pivoter chaque pied coupé à sa bonne orientation. Dans FreeCAD, il existe souvent plusieurs façons d'obtenir le même résultat. Il est important de s'en souvenir, car vous pouvez trouver plus facile ou plus efficace d'utiliser des techniques différentes dans des contextes différents.

- Nous allons maintenant faire des trous pour les vis, en utilisant la même opération Couper. Comme nous avons besoin de 8 trous (deux dans chaque pied), nous pourrions créer 8 objets à soustraire. Cependant, explorons une autre voie. Nous pouvons créer 4 cylindres, chacun intersectant une paire de pieds. Créons-les donc avec l'outil  **Cylindre**. Vous pouvez faire un cylindre et le dupliquer trois fois. Donnez à chaque cylindre un rayon de 6 mm. Cette fois-ci, nous devons faire pivoter les cylindres en utilisant la propriété **Placement** sous l'onglet Données (**Remarque** : changez la propriété de l'axe avant de définir l'angle, ou la rotation ne sera pas appliquée) :
 - Cylindre : hauteur : 1300 mm, angle : 90°, axe : x:0, y:1, z:0, position : x : -10 mm, y : 40 mm, z : 720 mm
 - Cylinder001 : hauteur : 1300 mm, angle : 90°, axe : x:0, y:1, z:0, position : x : -10 mm, y : 840 m, z : 720 mm
 - Cylinder002 : hauteur : 900 mm, angle : 90°, axe : x:-1, y:0, z:0, position : x : 40 mm, y : -10 mm, z : 700 m
 - Cylinder003 : hauteur : 900 mm, angle : 90°, axe : x:-1, y:0, z:0, position : x : 1240 mm, y : -10 mm, z : 700 mm



Vous remarquerez que les cylindres dépassent des pieds de la table. En effet, comme dans toutes les applications 3D basées sur les solides, les opérations booléennes dans FreeCAD échouent parfois lorsque les faces des objets sont coplanaires. Nous pouvons éviter les erreurs potentielles en plaçant les extrémités des cylindres au-delà des surfaces des pieds.

- Effectuons maintenant les soustractions pour créer des trous dans les pieds de la table. Sélectionnez le premier pied, puis, en appuyant sur Ctrl, sélectionnez l'un des cylindres qui l'intersectent et appuyez sur le bouton **Soustraction**. Le trou sera créé dans le pied et le cylindre sera caché. Vous pouvez le retrouver dans l'arborescence en développant le nouvel objet coupé du pied.
- Sélectionnez l'autre pied qui est coupé par ce cylindre caché et répétez l'opération. Cette fois, sélectionnez le cylindre dans la vue arborescente, puisqu'il est caché dans la vue 3D. (Vous pouvez également rendre le cylindre visible et le sélectionner dans la vue 3D). Répétez cette opération pour les autres pieds jusqu'à ce que chacun d'entre eux ait deux trous :



Comme vous pouvez le constater, chaque pied est désormais décrit par une série d'opérations multiples imbriquées dans l'arborescence. Toute la géométrie que nous avons créée reste paramétrique et vous pouvez modifier à tout moment n'importe quel paramètre de n'importe laquelle des opérations précédentes. Dans FreeCAD, cette série est appelée « historique de modélisation », car elle enregistre l'historique des opérations effectuées.

Une autre particularité de FreeCAD est que le concept d'objet 3D et le concept d'opération 3D tendent à se confondre. Le terme « coupe » fait référence à une opération, ainsi qu'à l'objet 3D résultant de cette opération. Dans FreeCAD, cet objet est parfois appelé « fonctionnalité », plutôt qu'« objet » ou « opération ».

Modélisons maintenant le plateau de la table. Il s'agira d'un simple prisme rectangulaire. Commençons donc par un autre **Cube** et modifions ses dimensions sous l'onglet Données comme suit :

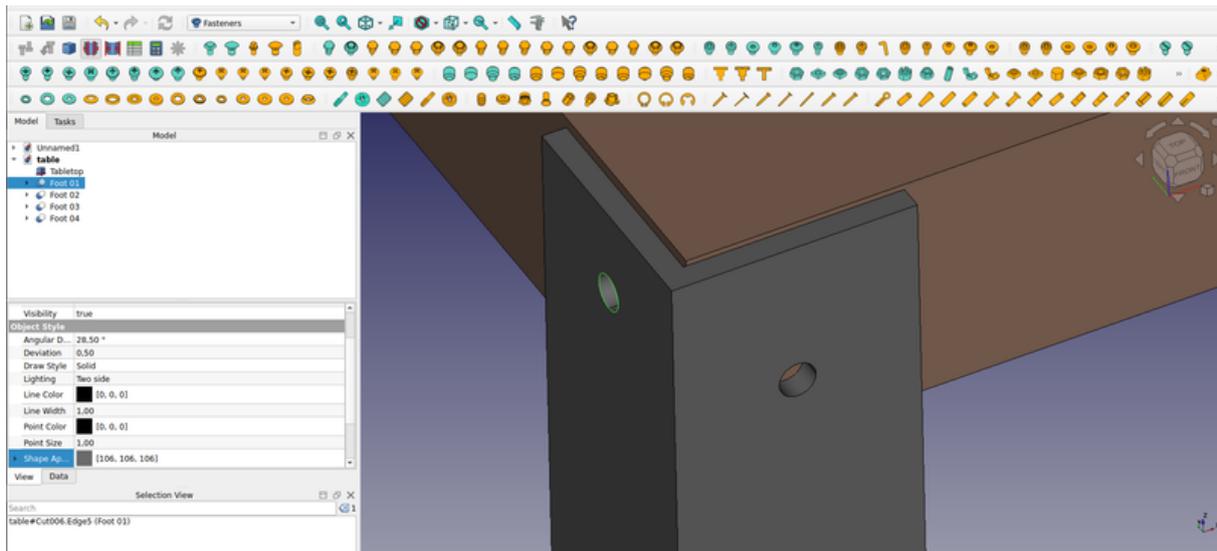
- Boîte : longueur : 1260 mm, largeur : 860 mm, hauteur : 80 mm, position : x : 10 mm, y : 10 mm, z : 670 mm.

Dans l'onglet **Vue**, vous pouvez lui donner une belle couleur brunâtre, semblable à celle du bois, en changeant sa propriété **Shape Appearance** :

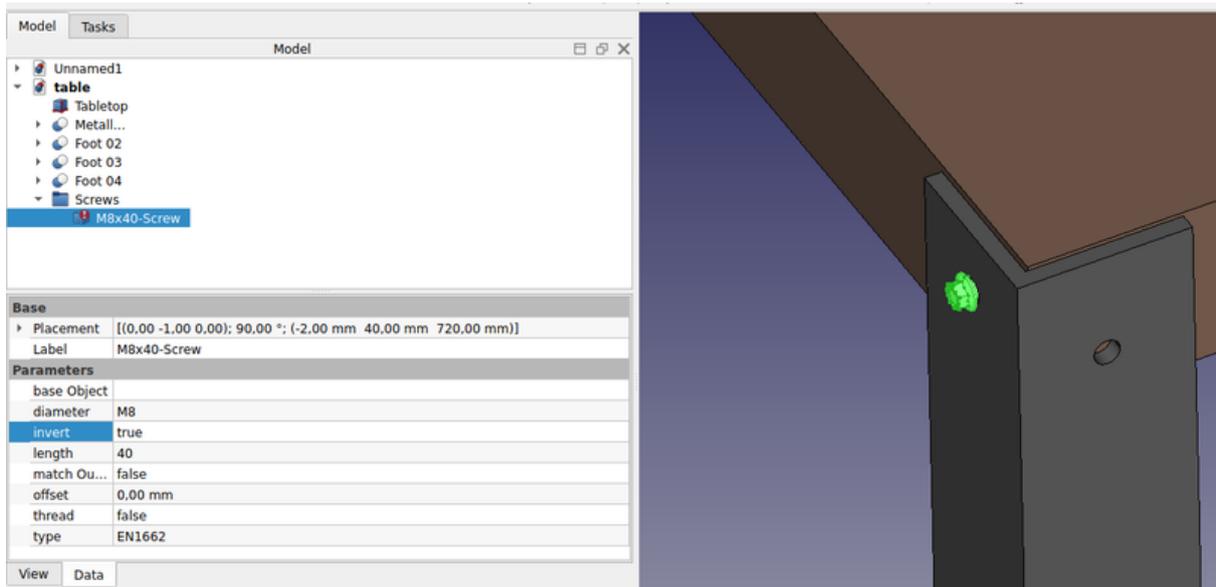
Maintenant que nos cinq pièces sont complètes, il est temps de leur donner des noms plus descriptifs que « Cut015 ». En cliquant avec le bouton droit de la souris sur chaque objet dans l'arborescence (ou en appuyant sur **F2** lorsqu'un objet est

sélectionné), vous pouvez leur donner un nom plus significatif. Le fait de nommer correctement vos objets peut être tout aussi important que la manière dont vous les modélisez.

- Nous allons maintenant insérer des vis à l'aide d'une extension. [Fasteners](#) est une extension extrêmement utile développée par un membre de la communauté FreeCAD. Vous pouvez le trouver sur le dépôt des [extensions de FreeCAD](#). Installer des extensions est facile ! Voir la page [Gestionnaire des extensions](#) pour plus d'informations.
- Une fois que vous avez installé l'atelier Fasteners et redémarré FreeCAD, sélectionnez Fasteners dans la liste déroulante des ateliers. Ajoutons une vis à l'un des trous que nous avons modélisés ci-dessus. Tout d'abord, sélectionnez l'arête circulaire d'un trou dans l'un des pieds de la table :



- Ensuite, sélectionnez l'une des boulons fournies dans l'atelier Fasteners. Pour cet exercice, nous utiliserons le **EN 1665 Boulon à tête hexagonale avec embase, série large**. Le boulon sera placé dans notre trou et aligné sur celui-ci, et son diamètre correspondra automatiquement à la taille de notre trou. Parfois, l'orientation du boulons devra être inversée, en utilisant sa propriété **Invert** :



- Répétez cette opération pour les sept autres trous et notre table est finie !

Comme indiqué précédemment, vous pouvez obtenir le même résultat en suivant des étapes différentes. Pour le démontrer, créons le même tableau en utilisant une méthodologie différente. N'oubliez pas qu'il n'y a pas de bonne ou de mauvaise méthode, mais seulement une créativité individuelle.

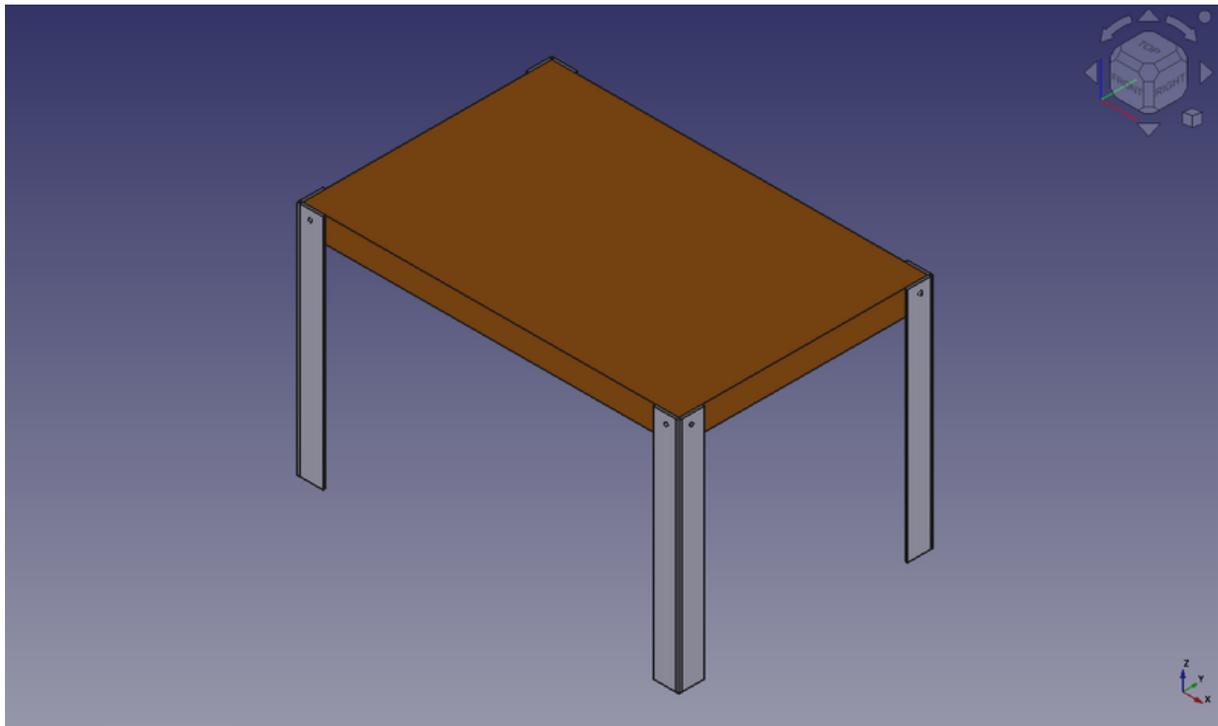
Nous allons commencer de la même manière : en créant un cube avec les dimensions suivantes : longueur 80 mm, largeur 8 mm, et hauteur 750 mm

- Créez un cube en sélectionnant le bouton  **Cube** et définissez les propriétés suivantes (dans l'onglet **Données**) :
 - Longueur : 80 mm
 - Largeur : 8 mm
 - Hauteur : 750 mm
- Ensuite, nous allons créer un  **Cylindre** avec les propriétés suivantes :
 - rayon : 6 mm, hauteur : 100 mm, angle : 90°, axe : x : 1, y : 0, z : 0, position : x : 40 mm, y : 40 mm, z : 720 mm
- Ensuite, nous allons appliquer l'opération Soustraction. Sélectionnez le cube, puis maintenez la touche Ctrl enfoncée et sélectionnez le cylindre. Gardez à l'esprit que l'ordre est important pour définir lequel reste. Ensuite, appuyez sur le bouton  **Soustraction**.
- Nous allons ensuite copier et coller l'objet coupé en appuyant sur **Ctrl+C** puis **Ctrl+V** (ou menu **Édition** → **Copier** et **Coller**) :
 - angle : 90°, axe : x : 0, y : 0, z : 1, position : x : 8 mm
- Sélectionnez les deux objets et appliquez l'outil  **Union**. Les deux objets sont maintenant fusionnés et nous avons un pied.
- Copiez et collez le pied fusionné, en le positionnant à
 - angle : 90°, axe : x : 0, y : 0, z : 1, position y : 800 mm.
- Sélectionnez les deux objets et appliquez l'outil  **Union**. Les deux objets sont maintenant fusionnés et nous avons un pied de table en forme de L.
- Copiez et collez le composé, en le positionnant à :
 - angle : 180°, axe : x:0, y:0, z:1, position x : 1200 mm, y : 800 mm. Nous avons nos pieds.

Créons le plateau de la table.

- Créez un cube et modifiez ses propriétés comme suit :
 - Longueur : 752 mm
 - Largeur : 1184 mm
 - Hauteur : 784 mm
 - Position x : 8 mm, y : 8 mm, z : 670 mm.

Maintenant, continuez à ajouter des boulons depuis l'atelier Fasteners comme précédemment.



La structure interne des objets Part

Comme nous l'avons vu plus haut, FreeCAD permet de sélectionner non seulement des objets entiers, mais aussi des parties d'objets, par exemple l'arrête circulaire de notre trou de boulon. C'est le bon moment pour jeter un coup d'œil rapide à la façon dont les objets Part sont construits en interne. Chaque atelier qui produit une géométrie de pièce sera basé sur ces objets :

- **Sommets** : ce sont les points (généralement les extrémités) sur lesquels tout le reste est construit. Par exemple, une ligne a deux sommets.
- **Arêtes** : les arêtes peuvent prendre la forme de lignes, d'arcs, d'ellipses ou de courbes [NURBS](#). Elles ont généralement deux sommets, mais certains cas particuliers n'en ont qu'un (par exemple, un cercle fermé).
- **Polylignes** : une polyligne est une séquence d'arêtes connectées par leurs extrémités. Une polyligne peut contenir des arêtes de n'importe quel type et peut être fermée ou non.
- **Faces** : les faces peuvent être planes ou courbes. Elles peuvent être définies par une polyligne fermée formant l'arête de la face, ou par plus d'une polyligne si la face a des trous.
- **Coques** : les coques sont des groupes de faces reliées par leurs arêtes. Elles peuvent être ouvertes ou fermées.

- **Solides** : une coque fermée peut être transformée en solide. Les solides comportent la notion d'intérieur et d'extérieur. De nombreux ateliers s'appuient sur cette notion pour s'assurer que les objets qu'ils produisent peuvent être construits dans le monde réel.
- **Composés** : les composés combinent d'autres formes d'un ou de plusieurs types en un seul objet.

Dans la vue 3D, vous pouvez sélectionner individuellement des **sommets**, des **arêtes** ou des **faces**. La sélection de l'un d'entre eux entraîne la sélection de l'ensemble de l'objet auquel il appartient.

Une note sur le design partagé

En regardant la table ci-dessus, vous pourriez penser que sa conception n'est pas très bonne. Les pieds ne semblent pas très bien fixés au plateau de la table ! Vous voudrez peut-être ajouter des pièces de renfort ou vous aurez peut-être d'autres améliorations à l'esprit. C'est là que le partage devient intéressant. Vous pouvez télécharger le fichier créé au cours de cet exercice à partir du lien ci-dessous et le modifier pour l'améliorer. Ensuite, si vous partagez ce fichier amélioré, d'autres personnes pourront peut-être l'améliorer ou utiliser votre table bien conçue dans leurs projets. Votre conception pourrait alors donner d'autres idées à d'autres personnes, et peut-être aurez-vous contribué à un monde meilleur...

Téléchargements

Le fichier produit dans cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/blob/master/files/table.FCStd>

Lire plus d'informations

- [Atelier Part](#)
- [Le dépôt FreeCAD addons](#)
- [L'atelier Fasteners](#)

Manuel : Dessin 2D traditionnel

L'atelier Draft de FreeCAD sert de passerelle entre le dessin 2D et la modélisation 3D, ce qui le rend unique par rapport aux systèmes de CAO 2D traditionnels tels qu'AutoCAD. Alors que FreeCAD fonctionne dans un environnement entièrement 3D, l'atelier Draft est conçu pour fournir aux utilisateurs des outils de dessin 2D familiers tout en offrant la flexibilité d'une transition en douceur entre les esquisses 2D et les objets 3D. Par exemple, vous pouvez définir des plans de travail personnalisés pour dessiner sur des surfaces ou des orientations spécifiques, ce qui permet de construire des modèles paramétriques. FreeCAD étant paramétrique, toute modification apportée aux dimensions est automatiquement répercutée sur l'ensemble du projet.

L'un des points forts de l'atelier Draft est son ensemble d'outils complet, qui comprend à la fois des outils de dessin de base et des outils de modification avancés. Ces outils peuvent être utilisés non seulement pour le dessin en 2D, mais aussi pour la manipulation d'objets dans l'espace 3D. Vous pouvez définir des plans de travail, des grilles et appliquer des contraintes pour vous assurer que les relations géométriques restent intactes dans votre conception. Les objets dessinés peuvent être modifiés et repositionnés à l'aide de modes d'accrochage et d'une variété de contraintes, ce qui facilite grandement la précision du dessin.

Quelques-uns des outils présents dans l'atelier de dessin :

1. Outils de dessin :

-  [Ligne](#),  [Polyligne](#) : ces outils permettent de créer des segments de ligne droite ou des polygones continus, qui peuvent être contraints et convertis en formes 3D.
-  [Cercle](#),  [Ellipse](#), et  [Arc](#) : utilisées pour définir des formes circulaires et elliptiques de base, avec des options pour une manipulation plus poussée.

1. Outils de manipulation :

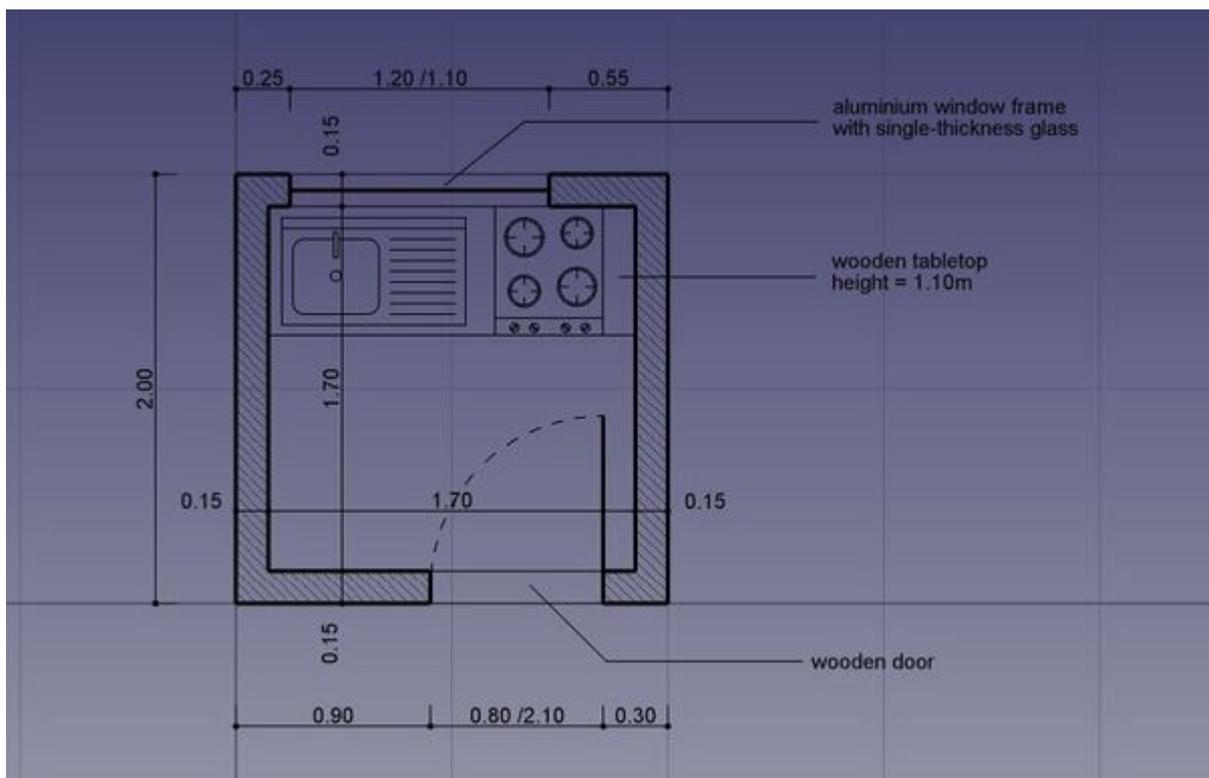
-  [Déplacer](#),  [Pivoter](#) ou  [Échelle](#) : ces opérations fonctionnent aussi bien en 3D qu'en 2D, ce qui offre une grande souplesse dans le positionnement des objets.
-  [Décaler](#) : semblable aux systèmes de CAO traditionnels, cette fonction vous permet de créer des lignes ou des courbes parallèles.
-  [Ajuster ou prolonger](#) et  [Étirer](#) : modifier les lignes et les formes en les coupant ou en les prolongeant pour qu'elles croisent ou rencontrent d'autres objets.
-  [Miroir](#) et  [Réseau](#) : ces outils reproduisent et modèlent les objets, ce qui est idéal pour les composants répétitifs.

Le système d'aimantation de l'atelier Draft de FreeCAD est conçu pour la précision. Que vous travailliez en 2D ou en 3D, vous pouvez vous aimanter à des points critiques tels que les points d'extrémité, les points médians et les centres de cercles, ce qui facilite le positionnement des éléments les uns par rapport aux autres. Des modes tels que l'aimantation perpendiculaire, tangente et à l'intersection améliorent encore la précision. Associés au plan de travail et au système de grille, ces outils garantissent l'alignement précis des objets et des composants.

La nature paramétrique de FreeCAD permet d'appliquer des contraintes aux éléments dessinés, ce qui garantit que les relations géométriques restent intactes. Par exemple, vous pouvez rendre les lignes parallèles ou perpendiculaires et définir des distances fixes entre les éléments. Ces contraintes peuvent être ajustées ultérieurement, ce qui rend les modifications de conception fluides et cohérentes sur l'ensemble du projet. L'atelier Draft s'intègre également de manière transparente à d'autres ateliers FreeCAD, tels que Sketcher, conçu pour une conception 2D paramétrique plus contraignante, et TechDraw, qui produit des dessins techniques en 2D à des fins de documentation.

Les fonctionnalités avancées de l'atelier Draft incluent la possibilité d'importer et d'exporter des fichiers dans des formats tels que DXF et SVG, ce qui vous permet de travailler avec des utilisateurs d'autres programmes de CAO ou de partager des conceptions avec eux. Les scripts Python améliorent encore les capacités de FreeCAD, vous permettant d'automatiser des tâches ou de créer des flux de travail personnalisés. Vous pouvez écrire des scripts qui génèrent des objets de dessin en fonction de règles géométriques spécifiques, rationalisant ainsi les tâches répétitives.

Pour montrer le travail et les possibilités de l'ébauche de travail, nous réaliserons un exercice simple, dont le résultat sera ce petit dessin, montrant le plan d'étage d'une petite maison qui ne contient qu'un dessus de cuisine (un plan d'appartement très absurde, mais on peut faire ce que nous voulons ici, n'est-ce pas?) :



- Passez à l'**atelier Draft**'
- Comme dans toutes les applications de dessin technique, il est sage de configurer votre environnement correctement, cela vous fera gagner beaucoup de temps. Si vous souhaitez personnaliser votre expérience dans l'atelier Draft, vous pouvez facilement le faire en ajustant divers paramètres dans le panneau des préférences de Draft en allant dans **Édition** → **Préférences** → **Draft**. Dans cet exercice, cependant, nous agissons comme si ces paramètres étaient laissés à leurs valeurs par défaut.

la souris. Il est souvent préférable de ne garder que ceux que vous utiliserez réellement.

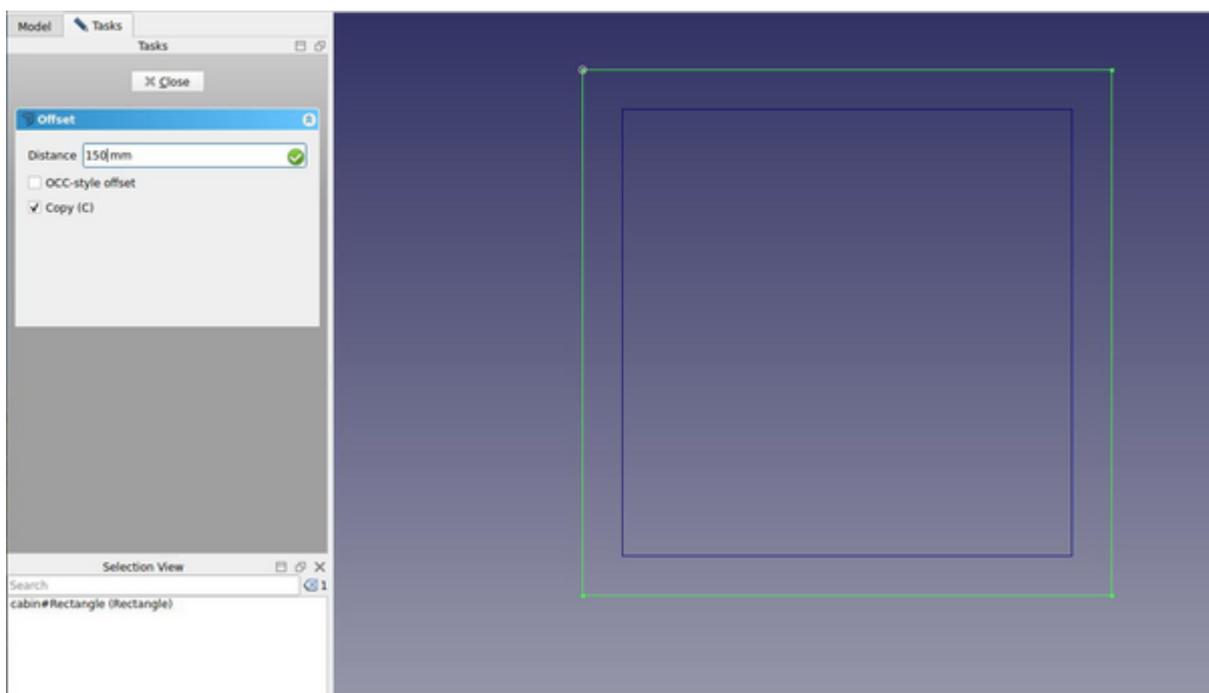
- Commençons par activer le **mode construction**, ce qui nous permettra de tracer des lignes directrices sur lesquelles nous dessinerons notre géométrie finale. Vous pouvez le faire en appuyant sur la commande [Basculer en mode construction](#).
- Si vous préférez, vous pouvez régler le plan de travail sur XY. Cela verrouillera le plan de travail, garantissant qu'il reste sur le plan XY quelle que soit la façon dont vous changez la vue. Si vous choisissez de ne pas le faire, le plan de travail s'adaptera automatiquement à la vue en cours, ce qui signifie que vous devrez vous assurer que vous êtes dans la vue du dessus chaque fois que vous souhaitez dessiner sur le plan XY (sol) afin d'éviter des changements d'orientation involontaires.

Commençons par tracer la forme de base de notre plan d'étage.

- Appuyez sur le bouton  **Draft Rectangle**.
- Dessinez un rectangle de 2 mètres par 2 mètres à partir du point (0,0,0), en laissant la coordonnée Z à zéro. Vous pouvez effectuer ce processus de manière efficace en utilisant le clavier, sans avoir besoin de la souris. Il suffit de taper :
 - **re, Enter, Enter, Enter, 2m, Enter, 2m, Enter, 0 et Enter.**

Ce flux de travail au clavier accélère la rédaction, en particulier pour les tâches répétitives ou les saisies de précision, ce qui en fait un outil idéal pour les utilisateurs désireux de rationaliser leur flux de travail. Vous pouvez visualiser les frappes pour chaque objet en survolant le bouton correspondant.

- Dupliquez ce rectangle de 15 cm à l'intérieur, en utilisant l'outil [Décalage](#), en activant son mode Copie et en lui donnant une distance de 15 cm :

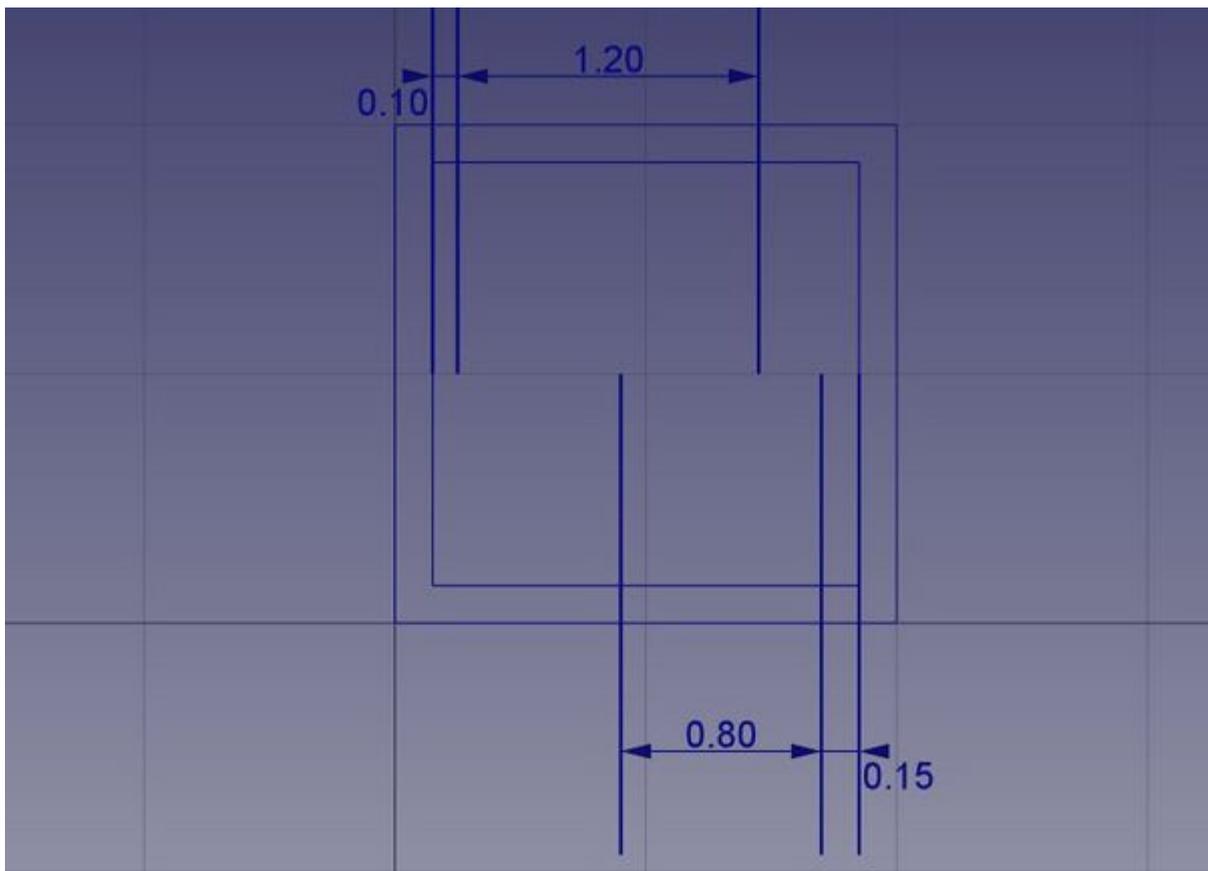


- Nous pouvons ensuite tracer quelques lignes verticales pour définir l'emplacement des portes et des fenêtres, en utilisant l'outil  [Ligne](#). (notez

que la case du mode « relatif » doit être décochée pour cette étape). Le croisement de ces lignes avec nos deux rectangles nous donnera des intersections utiles sur lesquelles nous pourrons fixer nos murs. Tracez la première ligne en définissant les points suivants :

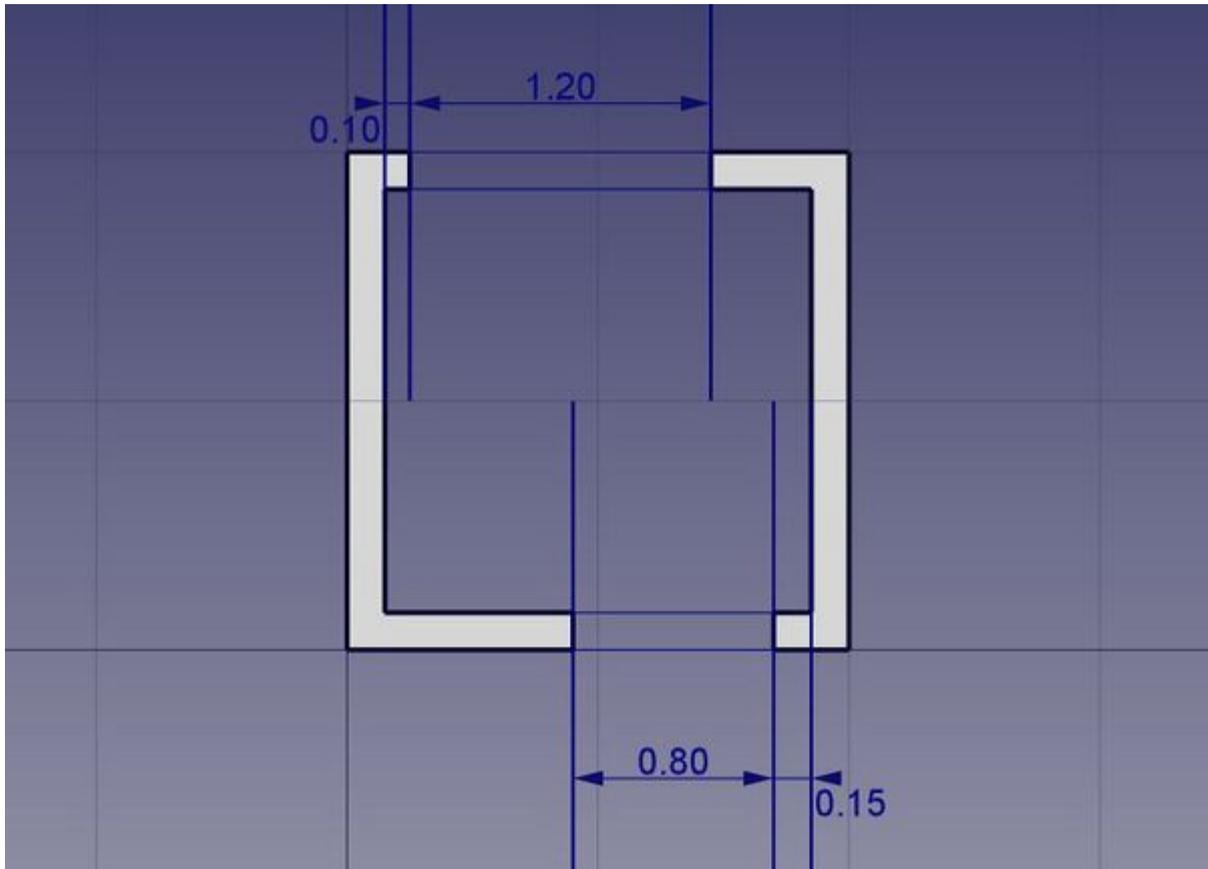
◦ **P1** (15cm, 1m, 0) et **P2** (15cm, 3m, 0).

- Nous allons maintenant dupliquer cette ligne 5 fois. Appuyez sur la  [Déplacer](#) avec le mode  [Cloner](#) activé. Assurez-vous que le **Mode relatif** est activé. Le mode Copie permet de s'assurer que chaque mouvement crée une nouvelle ligne au lieu de déplacer l'original, tandis que le mode relatif permet de définir des mouvements basés sur des distances relatives, ce qui facilite le positionnement sans avoir à calculer les coordonnées exactes. Commencez par sélectionner la ligne d'origine et lancez l'opération de déplacement en choisissant un point de départ quelconque, tel que (0,0,0). Après chaque déplacement, effectuez l'opération suivante sur la ligne nouvellement créée, de sorte que chaque copie s'appuie sur la précédente. Définissez les extrémités relatives de chaque nouvelle ligne
 - line001: x: 10cm
 - line002: x: 120cm
 - line003: x: -55cm, y: -2m
 - line004: x: 80cm
 - line005: x: 15cm



- C'est tout ce dont nous avons besoin maintenant, nous pouvons donc désactiver le mode de construction. Vérifiez que toute la géométrie de la construction a été placée dans un groupe "Construction", ce qui permet de le masquer tout à la fois ou même de le supprimer complètement plus tard.
- Maintenant, dessinons nos deux morceaux de mur en utilisant l'outil  [Polyligne](#). Assurez-vous que l'outil  [Aimantation d'intersection](#) est activé, car

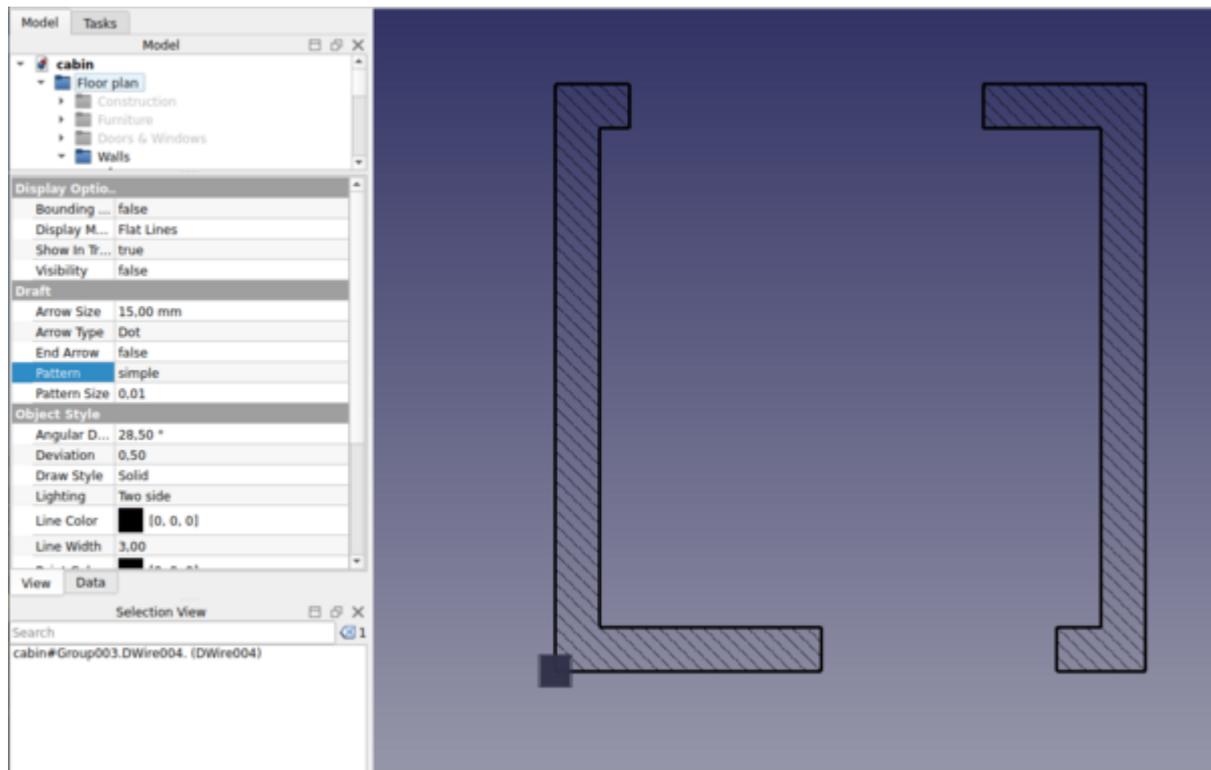
nous aurons besoin de nous aimanter aux intersections de nos lignes et rectangles. Dessinez deux fils comme suit, en cliquant sur tous les points de leurs contours. Pour les fermer, cliquez à nouveau sur le premier point, ou appuyez sur le bouton *Fermer* :



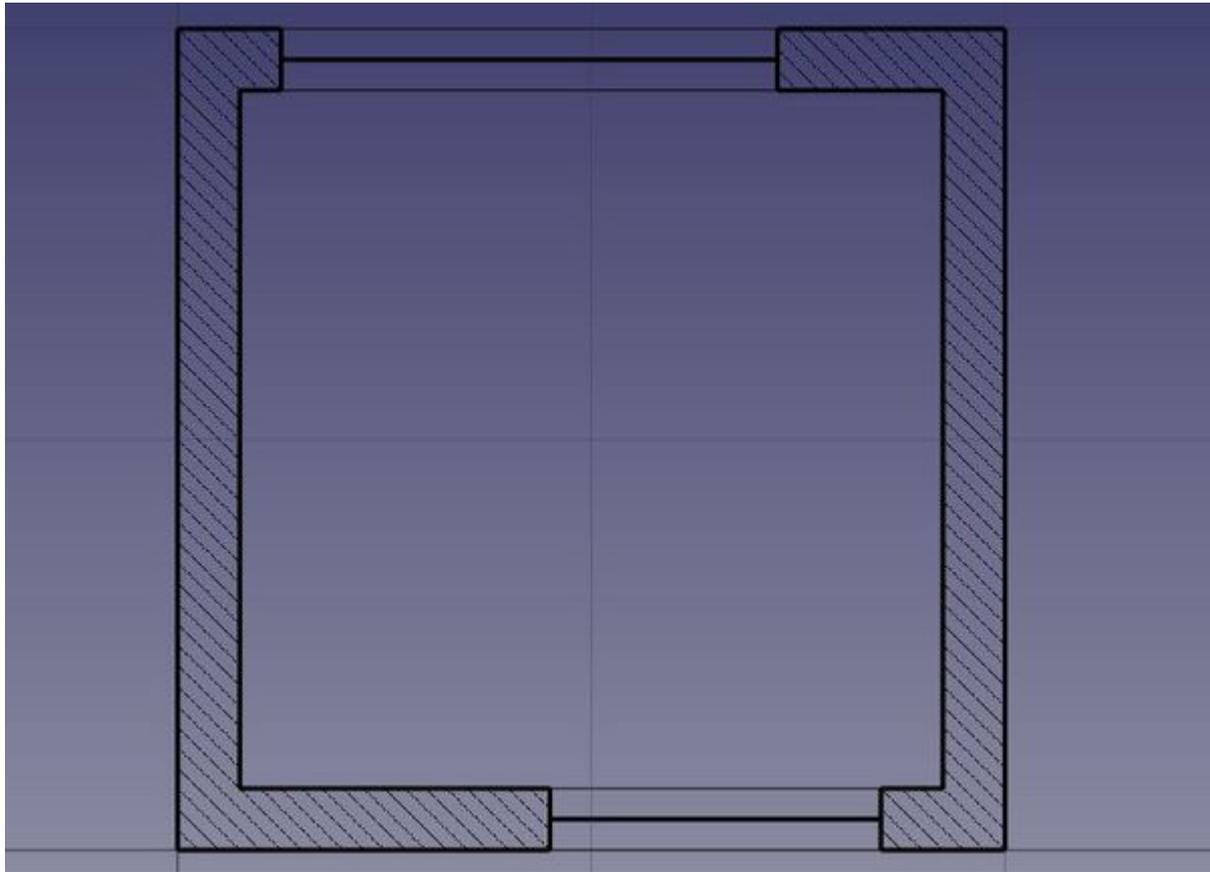
- Nous pouvons donner aux murs un joli motif de hachures. Sélectionnez les deux murs, assurez-vous que leur propriété **Make Face** (onglet Données) est réglée sur

`true`

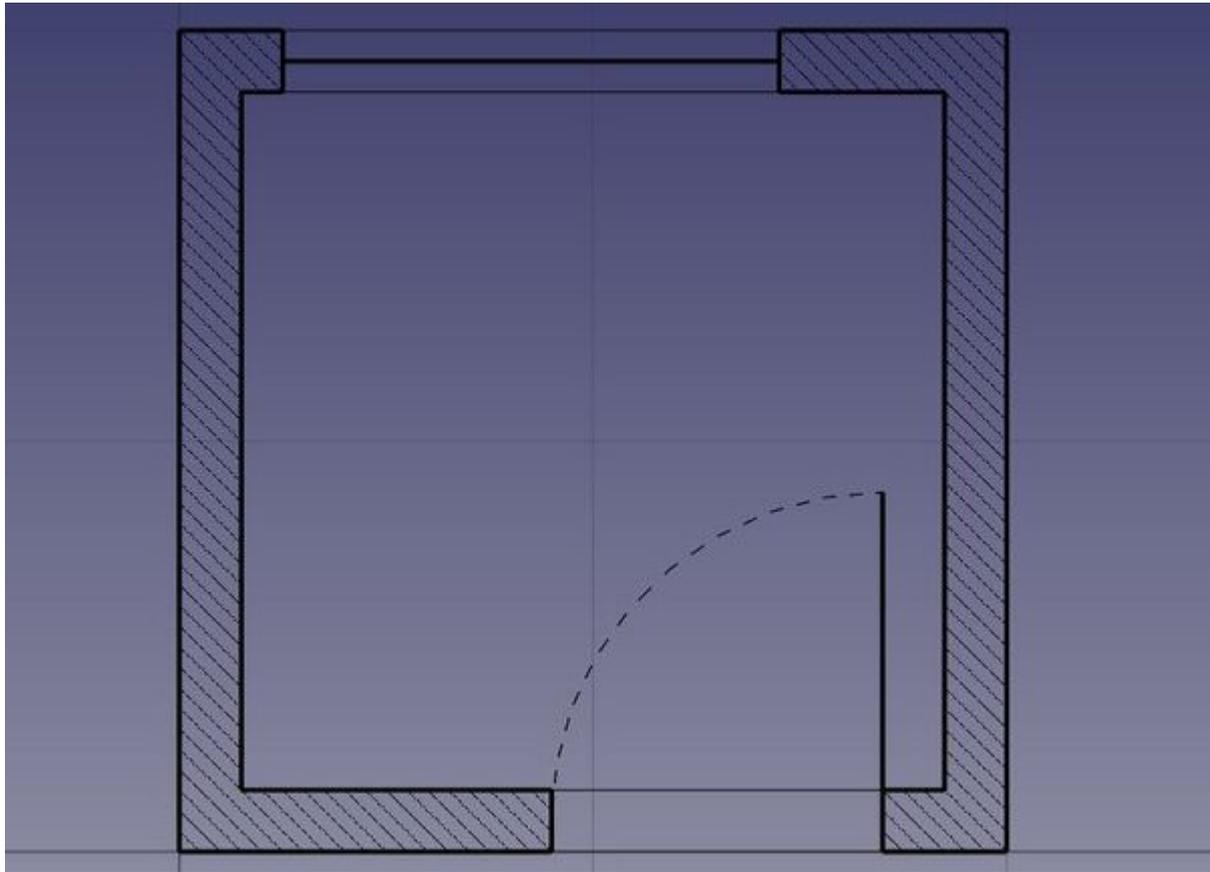
, puis réglez leur propriété **Pattern** sur **Simple** (onglet Vue), et leur **Pattern size** à votre convenance, par exemple **0.005**.



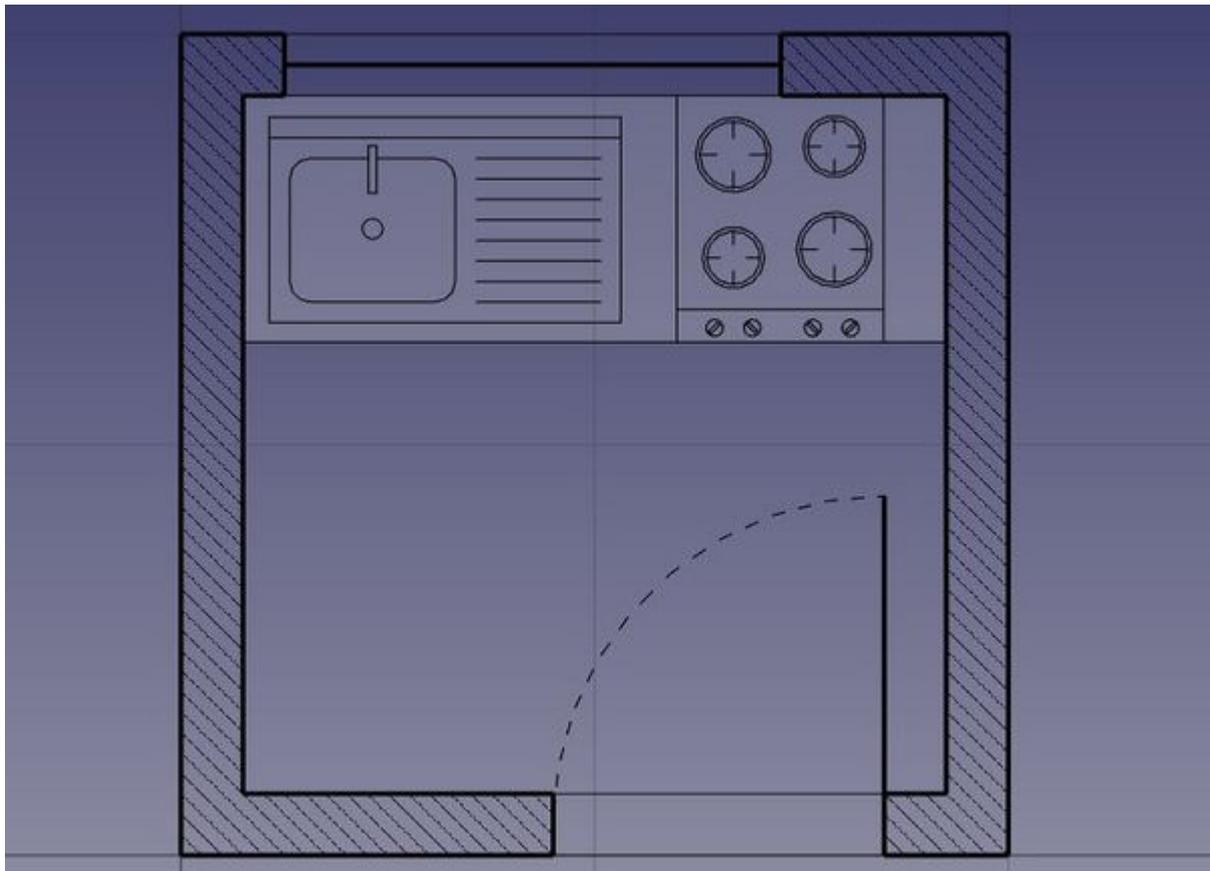
- Nous pouvons maintenant cacher la géométrie de construction en cliquant avec le bouton droit sur le groupe Construction et en choisissant **Masquer la sélection**.
- Dessinons maintenant les fenêtres et les portes. Assurez-vous que l'[aimantation au point milieu](#) est activé et tracez six lignes comme suit :



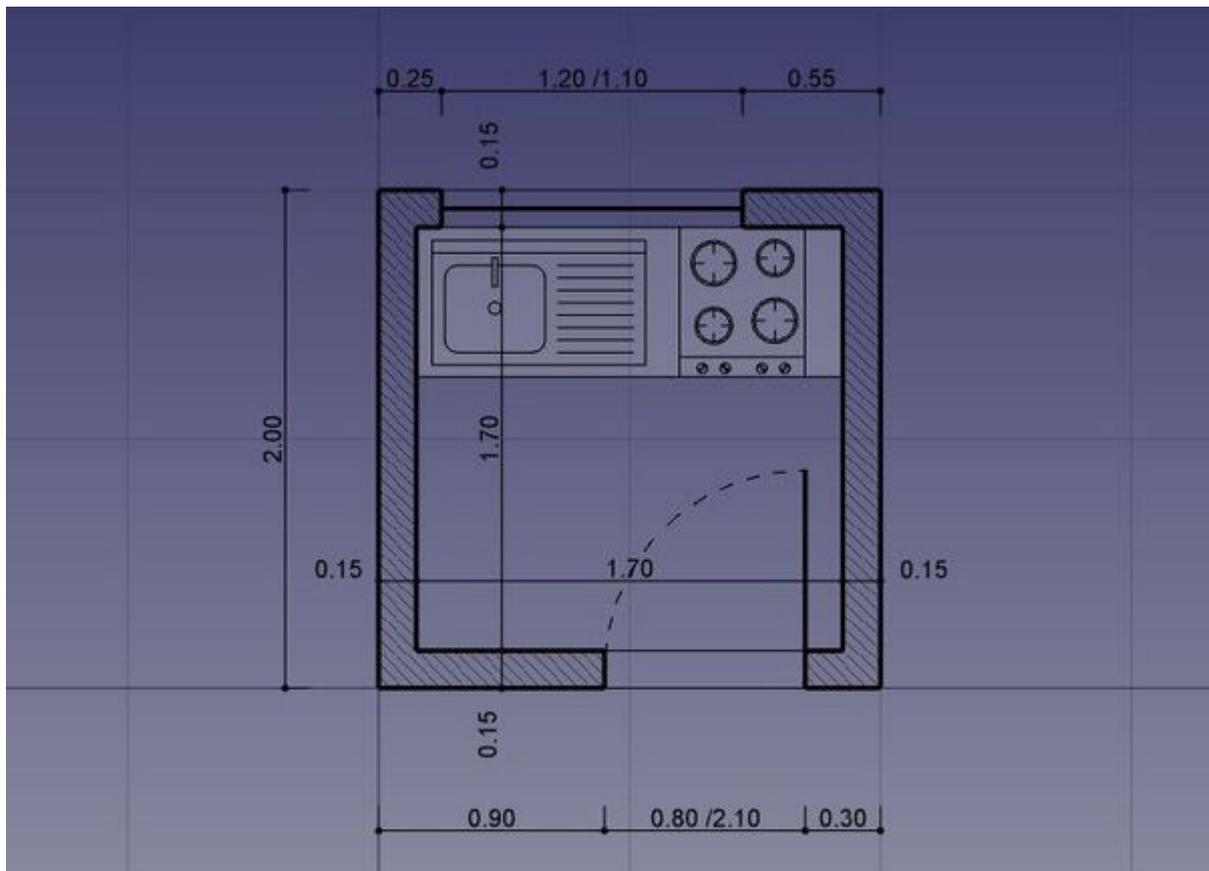
- Nous allons maintenant changer la ligne de porte pour créer un symbole de porte ouvert. Commencez par faire tourner la ligne à l'aide de l'outil [Pivoter](#). Cliquez sur l'extrémité de la ligne en tant que centre de rotation, puis spécifiez un angle de base de **0** et une rotation de **-90**.
- Ensuite, créez l'arc d'ouverture avec l'outil [Arc](#). Choisissez le même point que le centre de rotation que nous avons utilisé dans l'étape précédente comme le centre, cliquez sur l'autre point de la ligne pour donner le rayon, puis les points de début et de fin comme suit :



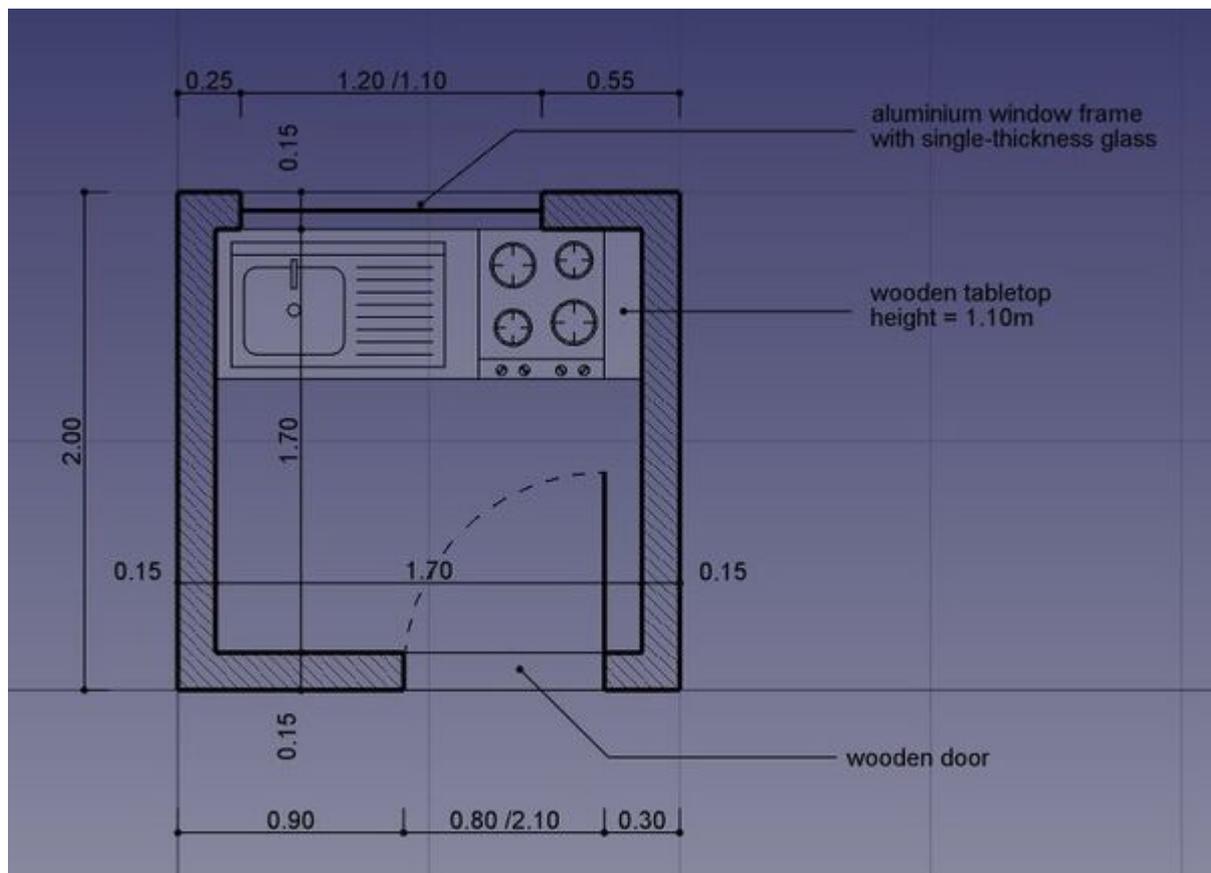
- Nous pouvons maintenant placer des meubles. Pour commencer, placez un compteur en dessinant un rectangle du coin supérieur gauche et en lui donnant une largeur de 170cm et une hauteur de -60cm. Dans l'image ci-dessous, la propriété **Transparence** du rectangle est configurée à 80%, pour lui donner un bon aspect des meubles.
- Ensuite, ajoutons un lavabo et une table de cuisson. Dessiner ces types de symboles à la main peut être très fastidieux, et ils sont généralement faciles à trouver sur Internet, par exemple sur <http://www.cad-blocks.net>. Dans la section **Téléchargements** ci-dessous, pour plus de commodité, nous avons récupéré un évier et une table de cuisson sur ce site et les avons sauvegardés en tant que fichiers DXF. Vous pouvez télécharger ces deux fichiers en visitant les liens ci-dessous et en cliquant avec le bouton droit sur le bouton **Raw**, puis en sélectionnant **Enregistrer sous**.
- L'insertion d'un fichier DXF dans un document FreeCAD ouvert peut être effectuée soit en choisissant l'option du menu **Fichier** → **Importer**, soit en faisant glisser et déposer le fichier DXF depuis votre explorateur de fichiers dans la fenêtre FreeCAD. Le contenu des fichiers DXF peut ne pas apparaître directement au centre de votre vue actuelle, selon l'endroit où ils se trouvaient dans le fichier DXF. Vous pouvez utiliser le menu **Affichage** → **Affichage standard** → **Afficher tout** pour faire un zoom arrière et trouver les objets importés. Insérez les deux fichiers DXF et déplacez-les à un emplacement approprié sur le dessus de la table :



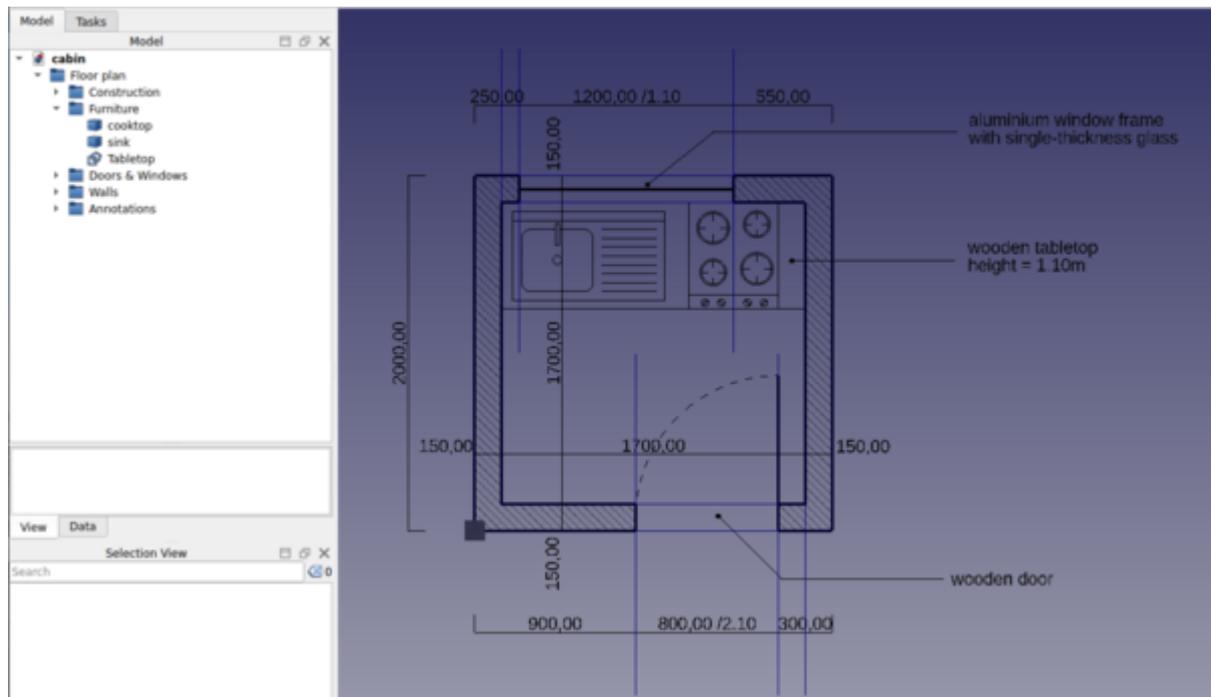
- Nous pouvons maintenant placer quelques dimensions à l'aide de l'outil  [Dimension](#). Pour créer une dimension, vous commencez par sélectionner trois points : le premier point définit le début de la mesure, le deuxième point définit le point final et le troisième point détermine l'endroit où la ligne de dimension et le texte seront placés. En cliquant soigneusement sur ces points, vous vous assurez que la dimension représente bien la distance entre les deux points sélectionnés. Si vous voulez forcer la dimension à être parfaitement horizontale ou verticale, même si les points de départ et d'arrivée ne sont pas alignés, maintenez la touche **Maj** enfoncée tout en cliquant sur le deuxième point. La dimension est ainsi verrouillée dans l'orientation souhaitée. Vous pouvez encore affiner la cote en ajustant les propriétés telles que la taille du texte, la précision et la couleur dans le panneau des propriétés, afin de vous assurer que les cotes correspondent aux normes visuelles et techniques de votre projet.
- Vous pouvez modifier la position d'un texte de cote en double-cliquant sur la dimension dans l'arborescence. Un point de contrôle vous permettra de déplacer le texte graphiquement. Dans notre exercice, les textes "0.15" ont été déplacés pour une meilleure clarté.
- Vous pouvez modifier le contenu du texte de cote en éditant leur propriété **Override**. Dans notre exemple, les textes des dimensions des portes et des fenêtres ont été édités pour indiquer leurs hauteurs :



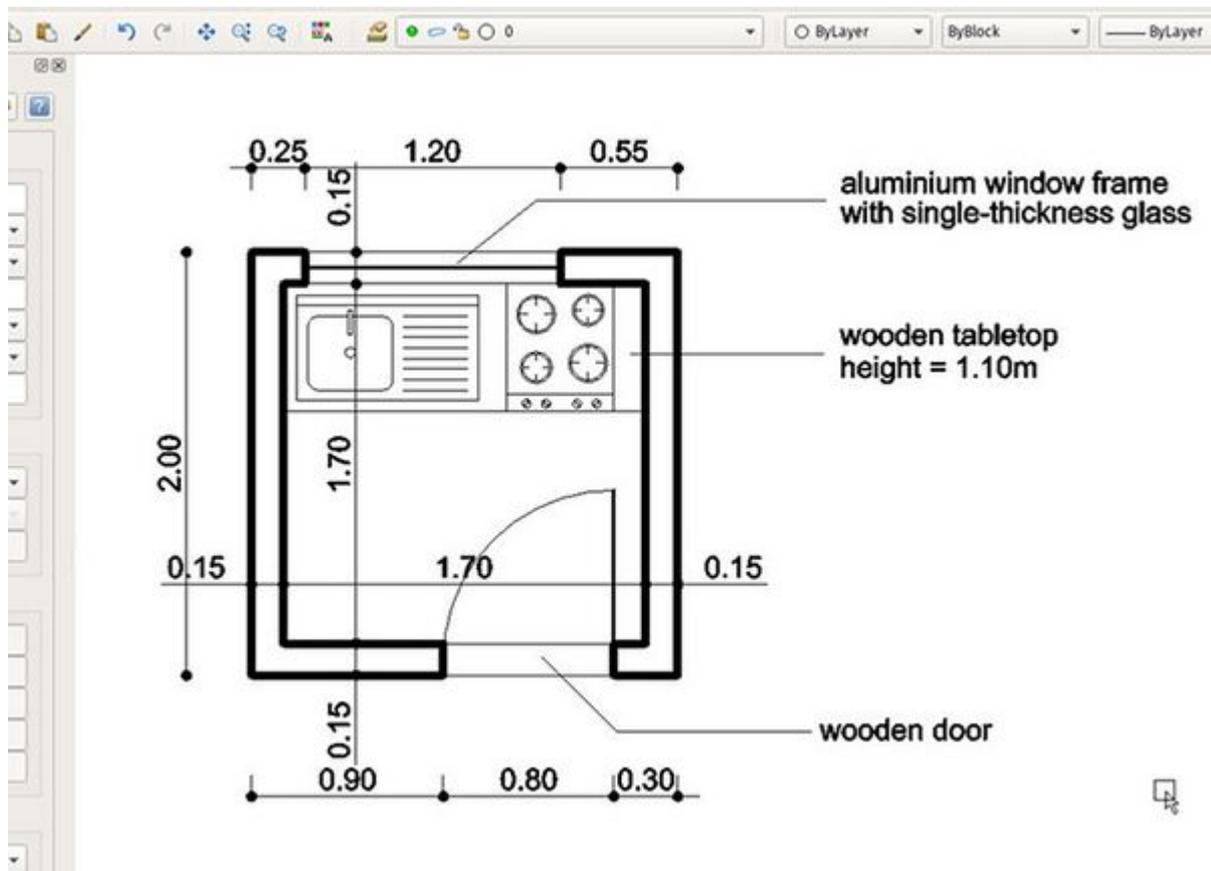
- Ajoutons des textes de description à l'aide de l'outil  [Texte](#). Cliquez sur un point pour positionner le texte, puis entrez les lignes de texte, en appuyant sur Entrée après chaque ligne. Pour terminer, appuyez deux fois sur Entrée.
- Les lignes de repère (également appelées «leaders») qui relient les textes à l'élément qu'ils décrivent se font simplement avec l'outil Polyligne. Dessinez les polygones, à partir de la position du texte, jusqu'à l'endroit décrit. Une fois que cela est fait, vous pouvez ajouter une bille ou une flèche à la fin des fils en définissant leur propriété **End Arrow** (extrémité de la flèche) sur **TRUE**.



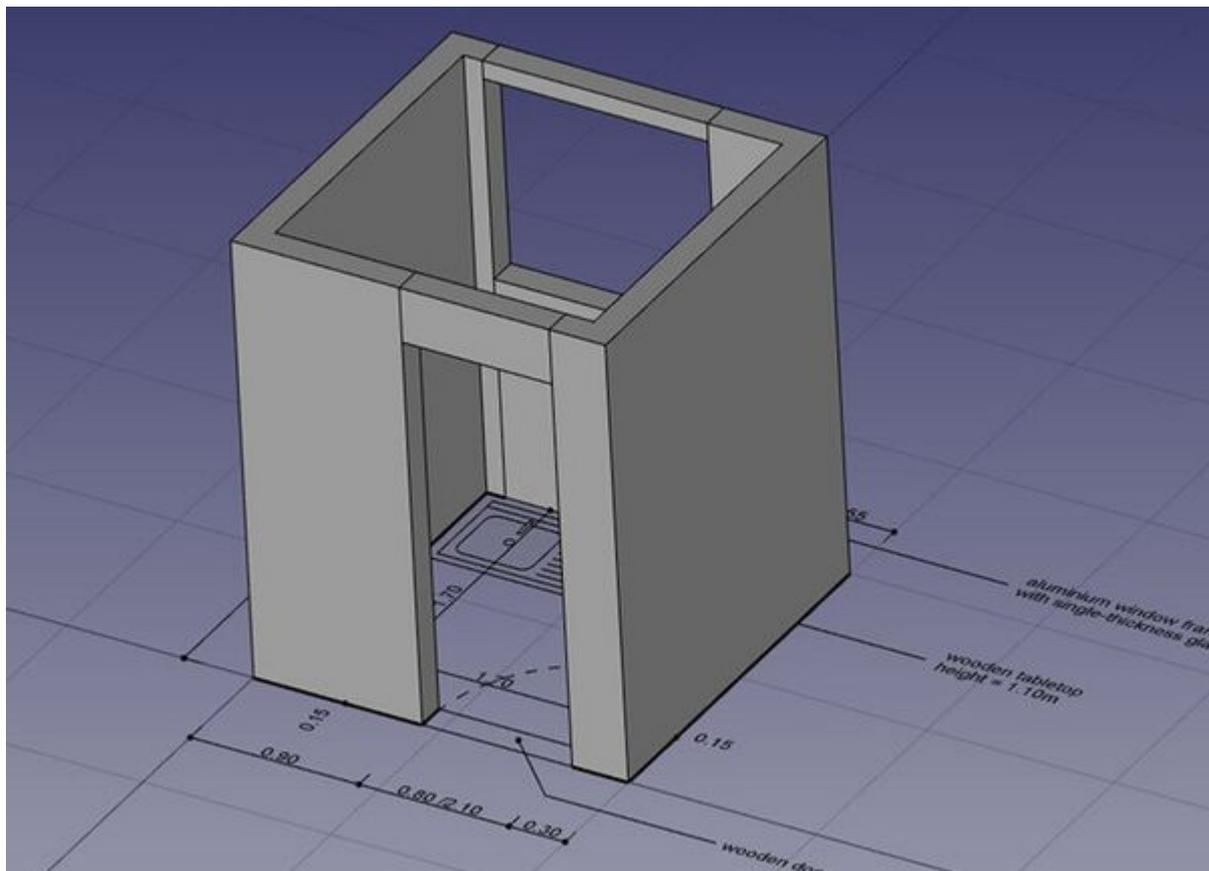
- Notre dessin est maintenant complet ! Compte tenu du nombre d'objets contenus dans le dessin, il convient de procéder à un certain nombre d'ajustements et de restructurations avant de considérer que le dessin est terminé. L'organisation de tous les éléments en groupes clairs et logiques permettra non seulement de maintenir le projet bien structuré, mais aussi de faciliter considérablement la navigation et la compréhension du fichier par d'autres personnes. En regroupant des éléments connexes, tels que des meubles, des appareils ou des éléments architecturaux, vous pouvez simplifier la mise en page et améliorer la clarté de la conception. Les modifications ou ajustements futurs seront également beaucoup plus faciles à gérer, en particulier si le projet doit être partagé ou faire l'objet d'une collaboration. En outre, une structure claire permet à toute personne qui examine le dessin de trouver rapidement des éléments spécifiques sans avoir à fouiller dans un espace de travail encombré, ce qui contribue en fin de compte à un produit final plus professionnel et plus soigné. Prendre le temps de s'organiser maintenant peut faire gagner beaucoup de temps et d'efforts par la suite :



- Nous pouvons maintenant imprimer notre travail en le plaçant sur une feuille de dessin, que nous allons montrer plus loin dans ce manuel, ou exporter directement notre dessin vers d'autres applications de CAO en l'exportant vers un fichier DXF. Il suffit de sélectionner notre groupe "Floor Plan", sélectionnez le menu **Fichier** → **Exporter**, et sélectionnez le format **Autodesk DXF**. Le fichier peut alors être ouvert dans n'importe quelle autre application CAO 2D telle que [LibreCAD](#). Vous pouvez remarquer quelques différences selon les configurations de chaque application.



- L'aspect le plus important de l'atelier Draft est que la géométrie 2D que vous créez peut servir de base à la création d'objets 3D. Vous pouvez facilement extruder ces formes en 3D à l'aide de l'outil [Part Extrusion](#) dans l'[atelier Part](#). Si vous préférez rester dans l'atelier Draft, vous pouvez utiliser l'outil [Ajuster ou prolonger](#), qui combine les fonctionnalités de découpe, d'extension et d'extrusion. L'outil Ajuster ou prolonger exécute essentiellement une extrusion de pièce sous le capot, mais le fait « à la manière d'une ébauche », ce qui vous permet d'indiquer visuellement et de fixer la longueur d'extrusion, vous donnant ainsi un meilleur contrôle et une plus grande précision lorsque vous travaillez directement dans votre environnement d'ébauche. Cette flexibilité rend la transition de la 2D à la 3D transparente et intuitive, en particulier pour ceux qui sont familiers avec les flux de travail 2D, tout en offrant des capacités de modélisation 3D avancées.
- En appuyant sur le bouton Sélectionner le [plan de travail](#) après avoir sélectionné une face d'un objet, vous pouvez également placer le plan de travail n'importe où et, par conséquent, dessiner des objets Draft dans différents plans, par exemple sur le dessus des murs. Ceux-ci peuvent ensuite être extrudés pour former d'autres solides 3D. Expérimentez le réglage du plan de travail sur l'une des faces supérieures des murs, puis tracez des rectangles dessus.



- Toutes sortes d'ouvertures peuvent également être faites aussi facilement en dessinant des objets Draft sur les faces des murs, puis en les extrudant, puis en utilisant les outils booléens de l'atelier Part pour les soustraire d'un autre solide, comme nous l'avons vu dans le chapitre précédent.

Fondamentalement, l'atelier Draft offre une approche plus graphique et intuitive de la création d'opérations de base, similaires à celles de l'atelier Part. Dans l'atelier Part, le positionnement des objets implique souvent d'ajuster manuellement des paramètres tels que les valeurs de placement (pour la position, la rotation, etc.), ce qui vous donne un contrôle précis mais peut parfois sembler moins intuitif, en particulier pour les modifications rapides. En revanche, l'atelier Draft vous permet d'effectuer ces mêmes opérations visuellement à l'écran, ce qui facilite le déplacement, la rotation et la manipulation des objets directement dans l'espace de travail à l'aide d'outils d'accrochage et d'options de positionnement relatif.

C'est dans cette différence que les ateliers se complètent. L'atelier Draft est idéal pour une conception rapide et interactive, car il permet de dessiner et de positionner des objets sans avoir à saisir constamment des valeurs numériques précises. En revanche, l'atelier Part offre un contrôle paramétrique plus détaillé sur les propriétés des objets, ce qui le rend plus adapté à des ajustements très précis, notamment dans le cadre de projets d'ingénierie ou de conception technique.

La beauté de FreeCAD est que vous n'avez pas besoin de choisir entre l'un ou l'autre. Vous pouvez créer des [barres d'outils personnalisées](#) en combinant des outils des ateliers Draft et Part, ce qui vous donne la possibilité de passer d'une méthode graphique à une méthode paramétrique en fonction de vos besoins. Vous pouvez ainsi bénéficier du meilleur des deux mondes - des ajustements rapides à l'écran à partir de l'atelier Draft et la précision de l'atelier Part - en fonction des besoins de votre projet. En outre, l'utilisation de raccourcis clavier et de barres d'outils

personnalisées peut accélérer votre flux de travail, en facilitant la transition entre les différentes opérations sans interrompre votre processus de conception.

Téléchargements

- Le fichier créé lors de cet exercice : <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/cabin.FCStd>
- Le fichier DXF de l'évier : <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/sink.dxf>
- Le fichier DXF de la table de cuisson : <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/cooktop.dxf>
- Le fichier final DXF produit lors de cet exercice : <https://github.com/yorikvanhavre/FreeCAD-manual/blob/master/files/cabin.dxf>

En relation

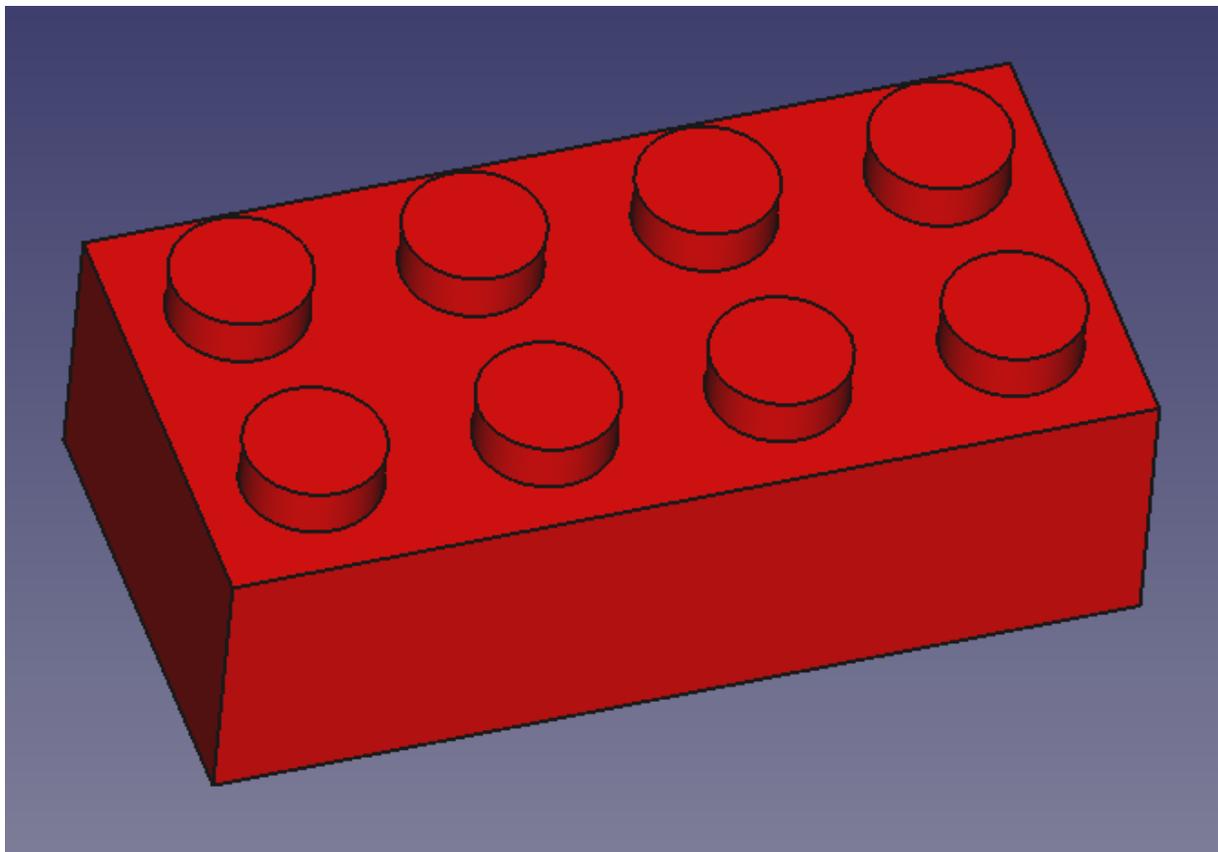
- [L'atelier Draft](#)
- [Draft Aimantation](#)
- [Draft Le plan de travail](#)

Manuel: Modélisation pour la conception de produits

L'[atelier PartDesign](#) dans FreeCAD est un outil polyvalent pour la création de modèles 3D paramétriques, particulièrement utile pour les conceptions de corps solides. Il vous permet de commencer par des esquisses en 2D, qui peuvent ensuite être transformés en objets 3D à l'aide d'opérations telles que des 🍌 [protrusions](#), des 🌀 [révolutions](#) et des 🏊 [cavités](#). Cet atelier est essentiel pour la conception de pièces nécessitant précision et contrôle paramétrique, car les modifications apportées aux esquisses ou aux fonctions mettent automatiquement à jour l'ensemble du modèle.

L'un des principaux avantages de l'atelier PartDesign est qu'il permet de créer des pièces pour l'impression 3D. Les imprimantes 3D ayant besoin de modèles solides et étanches, l'atelier PartDesign veille à ce que toutes les fonctions restent dans un seul corps cohérent. Cela permet d'éliminer les problèmes courants tels que les espaces ou le chevauchement des faces, qui peuvent poser des problèmes lors du découpage en tranches pour l'impression 3D. Une fois votre conception terminée, vous pouvez facilement exporter le modèle sous forme de fichier STL, un format largement pris en charge par les imprimantes 3D. L'atelier PartDesign est donc l'option idéale pour créer des objets imprimables de haute qualité, qu'il s'agisse de prototypage, de conception de pièces fonctionnelles ou de création de modèles complexes pour l'impression 3D.

Pour illustrer comment fonctionne l'atelier PartDesign, créons ce morceau bien connu de [Lego](#). Vous pouvez également vous référer à [PartDesign Tutoriel d'introduction V0.19](#) si vous souhaitez essayer un autre objet.



Nous allons maintenant utiliser exclusivement les outils des ateliers ([Sketcher](#)) et [PartDesign](#). Étant donné que tous les outils de l'atelier Sketcher sont également inclus dans l'atelier PartDesign, nous pouvons rester dans PartDesign et nous n'aurons pas à basculer entre les deux.

Dans l'atelier PartDesign de FreeCAD, les objets sont principalement construits à partir d'esquisses, qui sont des profils 2D composés de segments linéaires tels que des lignes, des arcs ou des ellipses, ainsi que d'une série de contraintes. Ces contraintes dictent des règles géométriques spécifiques pour l'esquisse et peuvent être appliquées aux segments eux-mêmes et à leurs points clés, tels que les points d'extrémité ou les points centraux. Par exemple, vous pouvez utiliser une contrainte verticale sur une ligne pour la maintenir parfaitement verticale, ou une contrainte de position (verrouillage) pour fixer un point d'extrémité et l'empêcher de bouger.

Une esquisse est considérée comme entièrement contrainte lorsque chaque point est verrouillé en position par le nombre approprié de contraintes, ce qui signifie qu'aucune partie de l'esquisse ne peut être déplacée librement. L'idéal est d'obtenir une esquisse entièrement contrainte, car cela garantit que la conception est bien définie et stable, ce qui permet d'apporter des modifications prévisibles à un stade ultérieur du processus de conception. D'un autre côté, si l'on ajoute plus de contraintes que nécessaire (on parle alors d'esquisse surcontrainte), cela peut entraîner des conflits dans la géométrie. FreeCAD vous signalera toute contrainte redondante ou conflictuelle, car une contrainte excessive peut entraîner des problèmes lors d'opérations ultérieures telles que des extrusions ou des coupes.

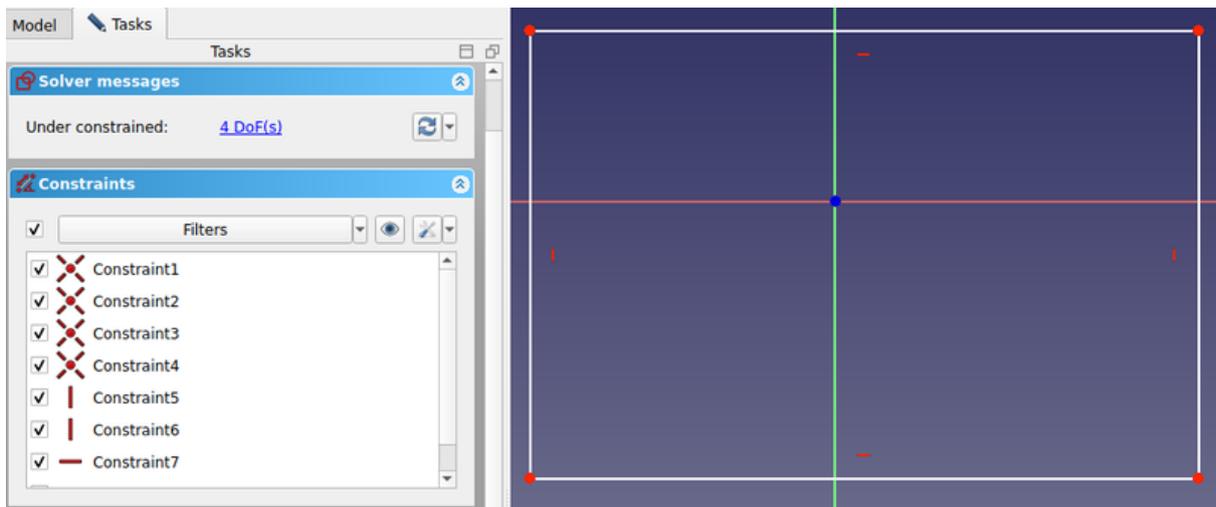
L'ajout des bonnes contraintes est essentiel pour créer un modèle paramétrique stable. En équilibrant soigneusement les contraintes, vous pouvez facilement modifier ou ajuster les esquisses sans casser la géométrie. Ce contrôle fait de l'atelier PartDesign un outil puissant pour une modélisation paramétrique précise, en particulier pour des tâches telles que l'impression 3D, où le maintien de relations géométriques correctes est essentiel pour produire des pièces précises et fonctionnelles.

Les esquisses disposent d'un mode d'édition, où leur géométrie et leurs contraintes peuvent être modifiées. Quand vous avez terminé l'édition, et quittez le mode d'édition, les esquisses se comportent comme n'importe quel autre objet FreeCAD, et peuvent être utilisées comme éléments de construction pour tous les outils de conception de pièces, mais aussi dans d'autres ateliers, tels que l'[atelier Part](#) ou l'[atelier Arch](#). L'[atelier Draft](#) dispose également d'un outil qui convertit les éléments Draft en esquisses, et vice versa.

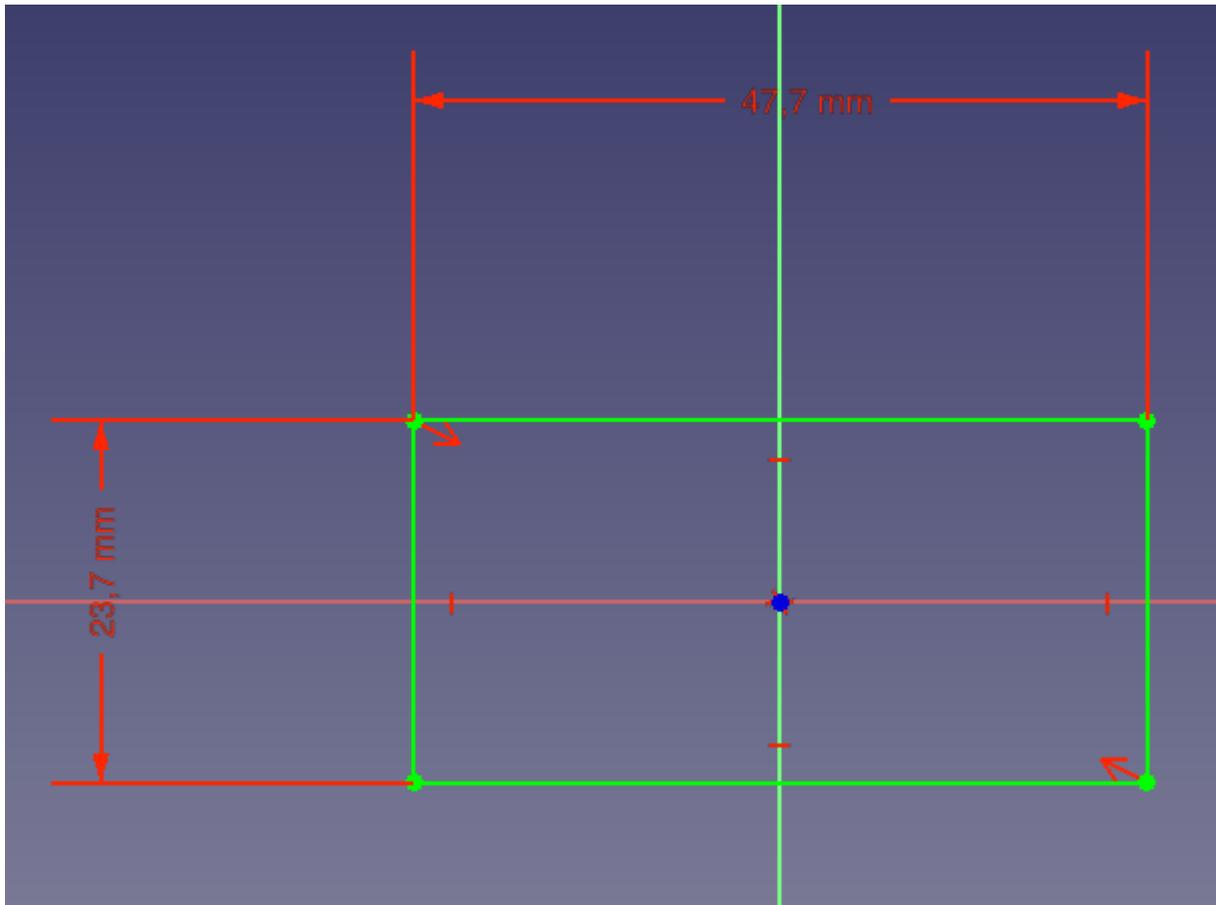
- Passez à l'[atelier PartDesign](#).
- Cliquez sur le bouton  [Créer une esquisse](#) et choisissez le plan **XY**, qui est le plan de base. L'esquisse sera créée et passera immédiatement en mode édition, et la vue sera tournée pour regarder votre esquisse de manière orthogonale.
- Dessinez un rectangle en sélectionnant l'outil  [Rectangle](#) et en cliquant sur deux points d'angle. Vous pouvez placer les deux points n'importe où, mais ne cliquez pas sur l'un des axes. L'emplacement correct des points sera défini dans l'une des étapes suivantes.
- Vous remarquerez que quelques contraintes ont été automatiquement ajoutées à notre rectangle : les segments verticaux ont reçu une contrainte verticale, les segments horizontaux une contrainte horizontale, et chaque coin une contrainte point sur point qui colle les segments ensemble. Vous pouvez

essayer de déplacer le rectangle en faisant glisser ses lignes avec la souris, toute la géométrie continuera d'obéir aux contraintes.

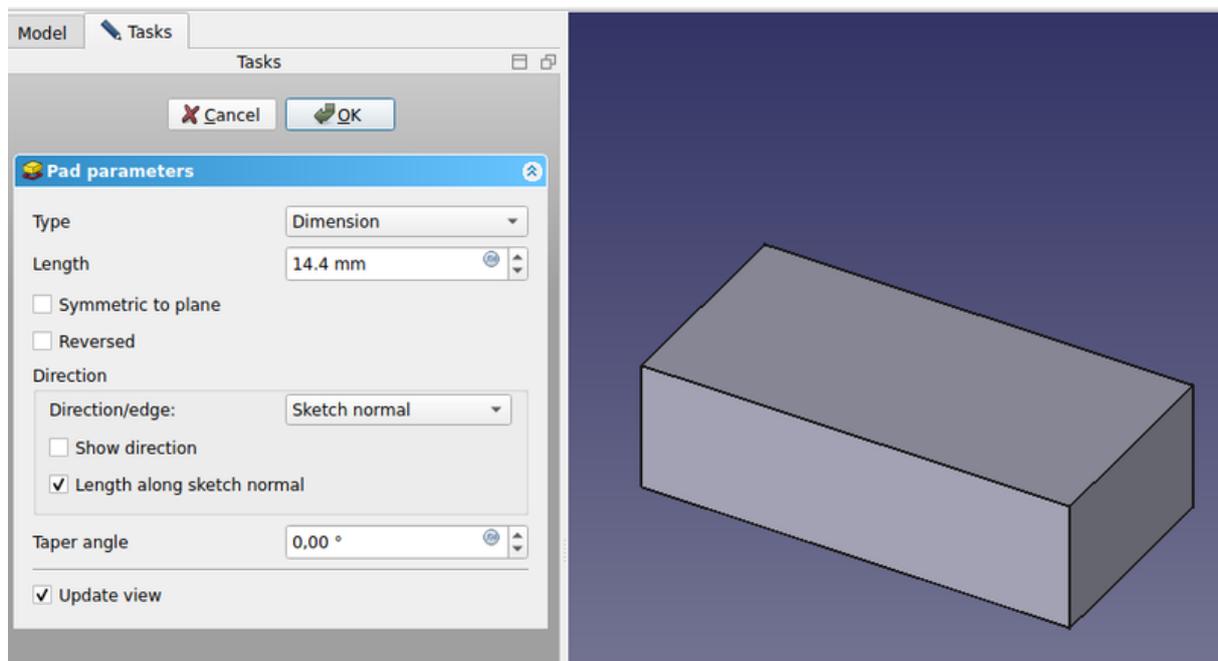
- Pour l'instant, notre esquisse est sous-contrainte, car il lui manque quatre contraintes : sa longueur, sa largeur et ses positions X et Y. Cette absence de contraintes vous permet d'agir librement sur la géométrie. Cette absence de contraintes vous permet de déplacer librement l'esquisse le long des axes X et Y. Tant que ces contraintes ne sont pas définies, la géométrie n'est pas totalement verrouillée, ce qui signifie que la taille et la position de l'esquisse restent ajustables. Pour définir complètement l'esquisse, nous devons appliquer des contraintes qui spécifient



- Maintenant, ajoutons trois autres contraintes:
 - Activez l'outil  [Contrainte de dimension](#), sélectionnez l'un des segments verticaux et réglez sa longueur à 23,7 mm.
 - L'outil étant toujours actif, sélectionnez l'un des segments horizontaux et réglez sa longueur à 47,7 mm.
 - Activez l'outil  [contrainte de symétrie](#), sélectionnez le point d'angle supérieur gauche du rectangle, puis le point d'angle inférieur droit, et enfin le point d'origine (le point où les axes rouge et vert se croisent). Cela permet de s'assurer que le rectangle reste centré sur le point d'origine, ce qui limite son amplitude de mouvement et assure une symétrie entre les deux axes.
- Vous remarquerez maintenant que notre rectangle est devenu vert, ce qui indique qu'il est entièrement contraint. Cela signifie que chaque aspect de l'esquisse, y compris sa position, sa taille et sa forme, est maintenant entièrement défini et verrouillé. Il est généralement conseillé de s'assurer que les esquisses sont entièrement contraintes, car cela permet de garder le contrôle de la conception et d'éviter les modifications involontaires au cours des opérations ultérieures.



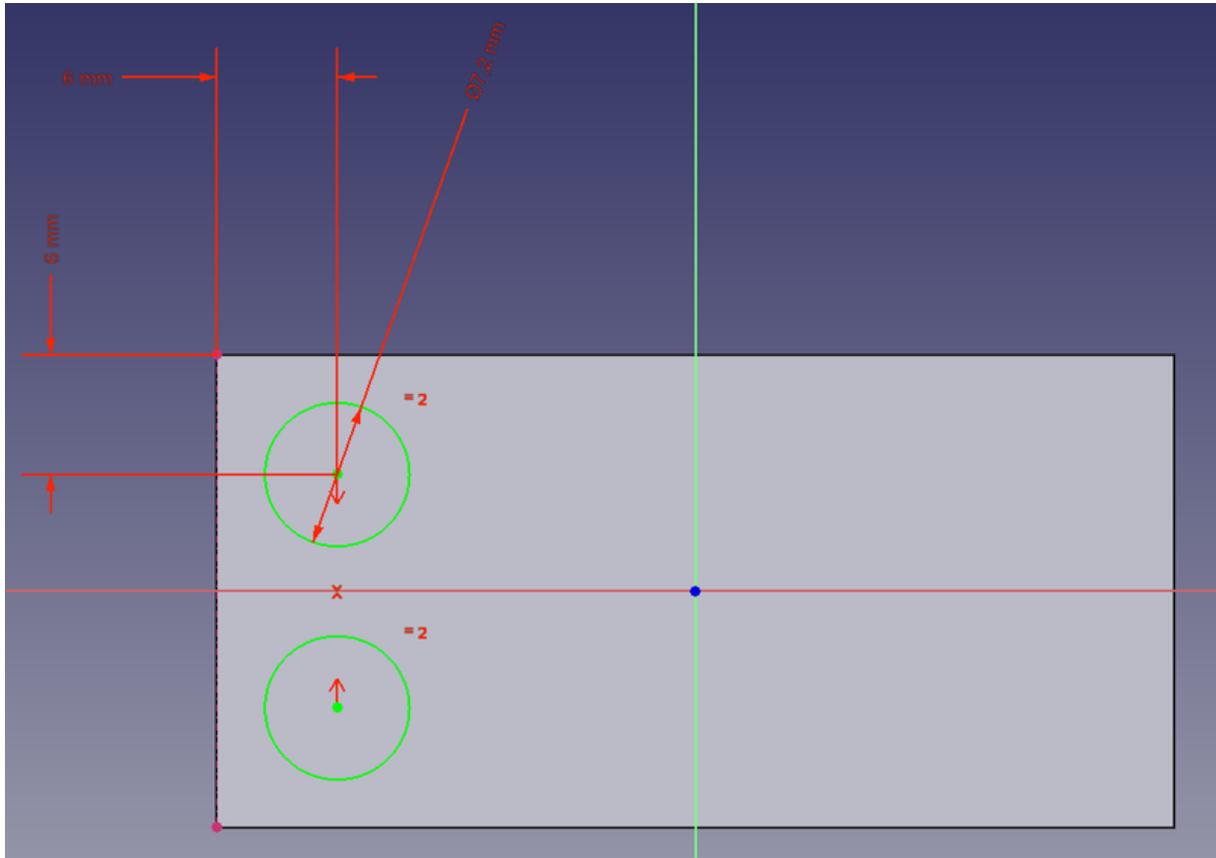
- Notre esquisse de base est maintenant prête, nous pouvons quitter le mode d'édition en appuyant sur le bouton **Fermer** en haut du panneau de tâches, ou simplement en appuyant sur la touche **Échap**. Si nécessaire, plus tard, nous pouvons réactiver le mode d'édition à tout moment en double-cliquant sur le nom de l'esquisse dans l'arbre de construction ou en cliquant avec le bouton droit de la souris et en appuyant sur l'option éditer l'esquisse.
- Extrudons l'esquisse en utilisant l'outil Protrusion ( [PartDesign Protrusion](#)) et donnons-lui une dimension de 14,4 mm. Les autres options peuvent être laissées à leurs valeurs par défaut:



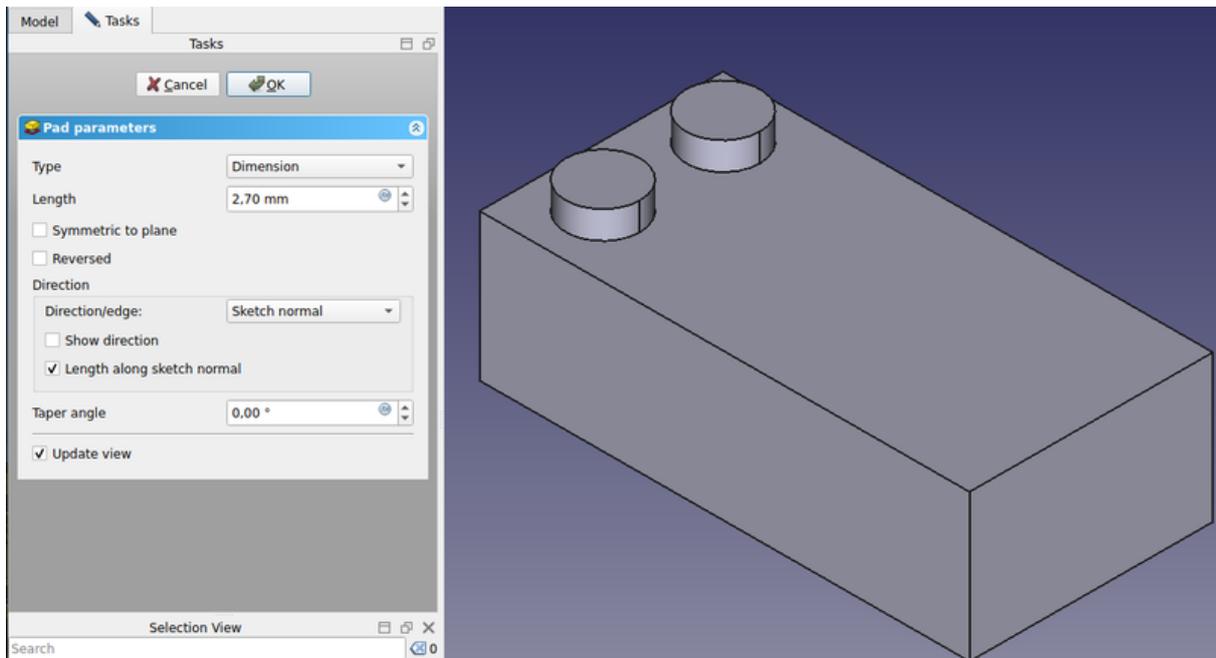
- L'outil **Protrusion** est similaire à l'outil [Part Extrusion](#) de l'[atelier Part](#) mais avec une différence essentielle : une extrusion est toujours liée à son esquisse et ne peut pas être déplacée de manière indépendante. Pour repositionner l'extrusion, vous devez déplacer l'esquisse de base, en veillant à ce que l'extrusion reste solidement attachée. L'extrusion fera toujours partie du même corps, préservant ainsi la continuité de votre conception, ce qui est particulièrement utile pour les pièces complexes dont les fonctions doivent être construites progressivement et alignées les unes avec les autres. Cela ajoute de la stabilité à votre conception, en particulier lorsque vous voulez vous assurer que tout reste correctement aligné et fixé en place.
- Créons les huit cylindres sur la face supérieure du bloc. Tout d'abord, sélectionnez la face supérieure du bloc, puis cliquez sur l'option [Aligner sur la sélection](#) pour aligner la vue sur cette face. Vous obtiendrez ainsi une vue claire et directe, ce qui facilitera le placement précis des cylindres.
- Cliquez sur le bouton [Créer une esquisse](#). La nouvelle esquisse sera créée directement sur la face supérieure.
- Créez deux [cercles](#) à l'endroit de votre choix.
- Choisissez le centre des deux cercles et l'axe des x (ligne rouge). Puis appuyez sur le bouton [contrainte de symétrie](#).
- Choisissez les deux cercles et appliquez une [contrainte d'égalité](#).
- En utilisant l'outil [Contrainte de dimension](#), définissez le diamètre d'un cercle à 7,2 mm. Comme nous avons déjà contraint les deux cercles à avoir le même diamètre, il n'est pas nécessaire de définir le diamètre du deuxième cercle : il s'ajustera automatiquement au premier.
- Nous devons maintenant positionner les cercles par rapport aux arêtes de la face. Cependant, vous remarquerez peut-être que nous ne pouvons pas sélectionner directement des points ou des arêtes. Pour résoudre ce problème, nous pouvons utiliser l'outil [Géométrie externe](#) pour référencer les arêtes de la face, ce qui nous permet de contraindre avec précision les cercles par rapport à la face. Appuyez sur le bouton et sélectionnez l'arête gauche de la face. Cette arête sera maintenant surlignée en rouge et vous pourrez créer des points de référence à partir de celle-ci, ce qui vous permettra d'appliquer des

contraintes pour positionner précisément les cercles par rapport aux limites de la face.

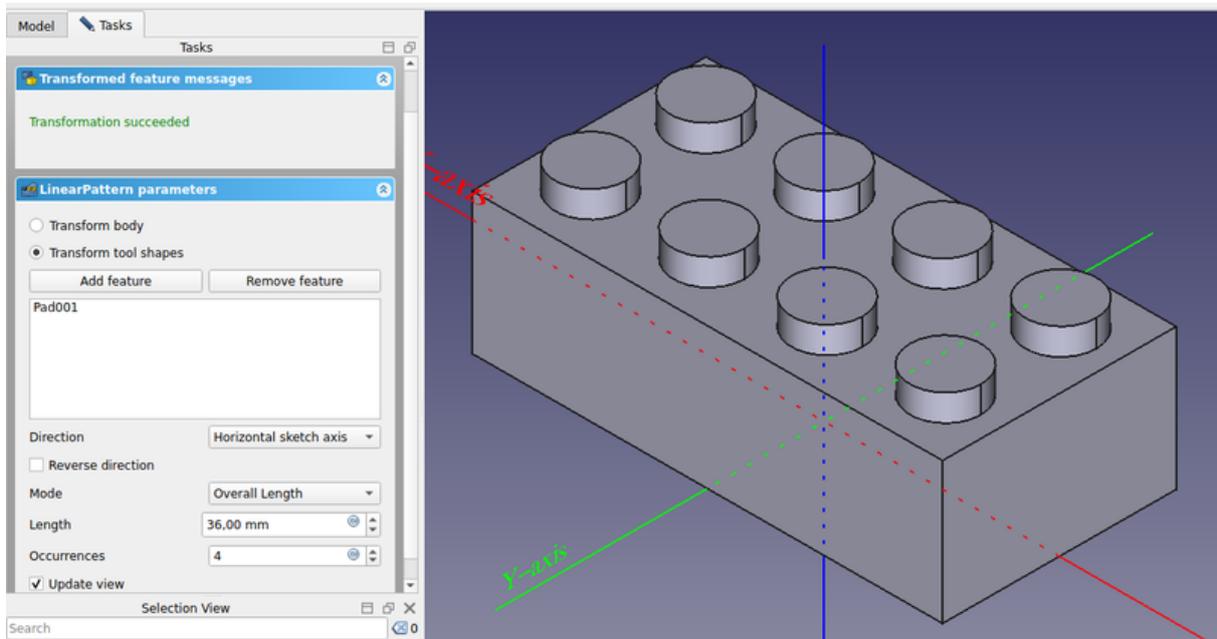
- Vous pouvez maintenant régler les distances X et Y de l'un des cercles sur 6 mm. Comme les cercles sont contraints l'un à l'autre, le deuxième cercle s'ajustera en conséquence.



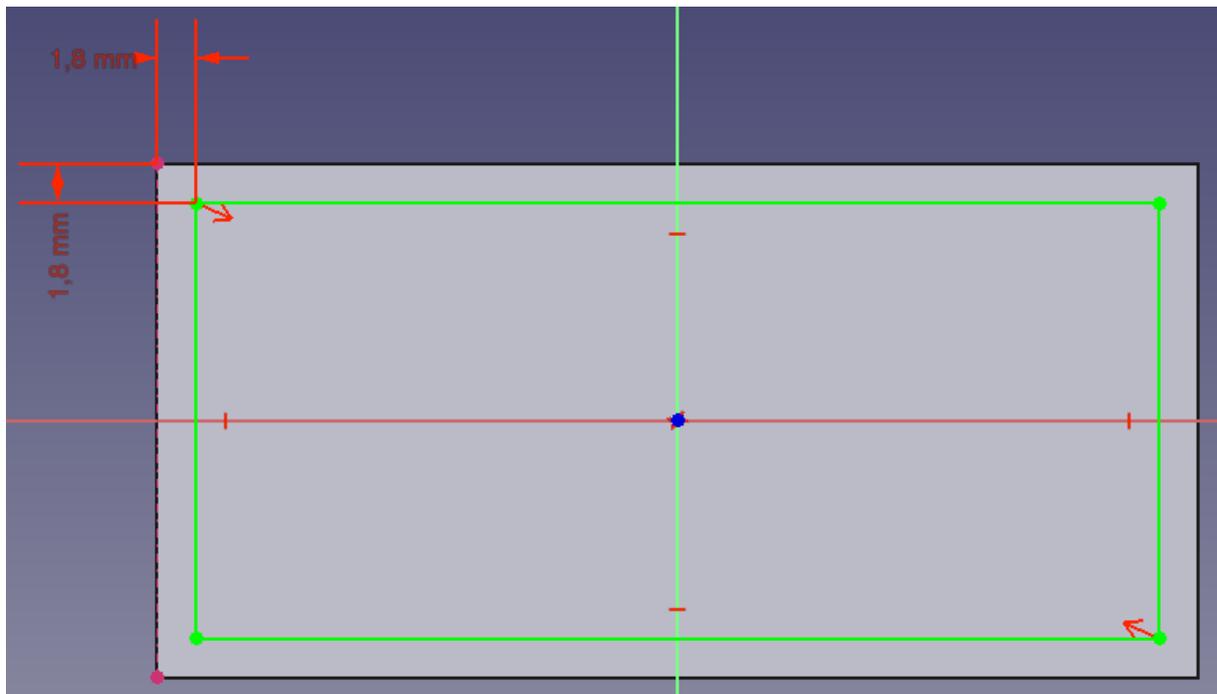
- Notez comment, une fois de plus, lorsque vous verrouillez la position et la dimension de tous les éléments de votre esquisse, celle-ci devient complètement contrainte. Cela vous assure du résultat pour la suite. Vous pourriez modifier la première esquisse maintenant, tout ce que nous avons fait ensuite serait conservé.
- Quittez le mode d'édition, sélectionnez cette nouvelle esquisse et créez une protrusion (🔧 [PartDesign Protrusion](#)) de 2.7mm :



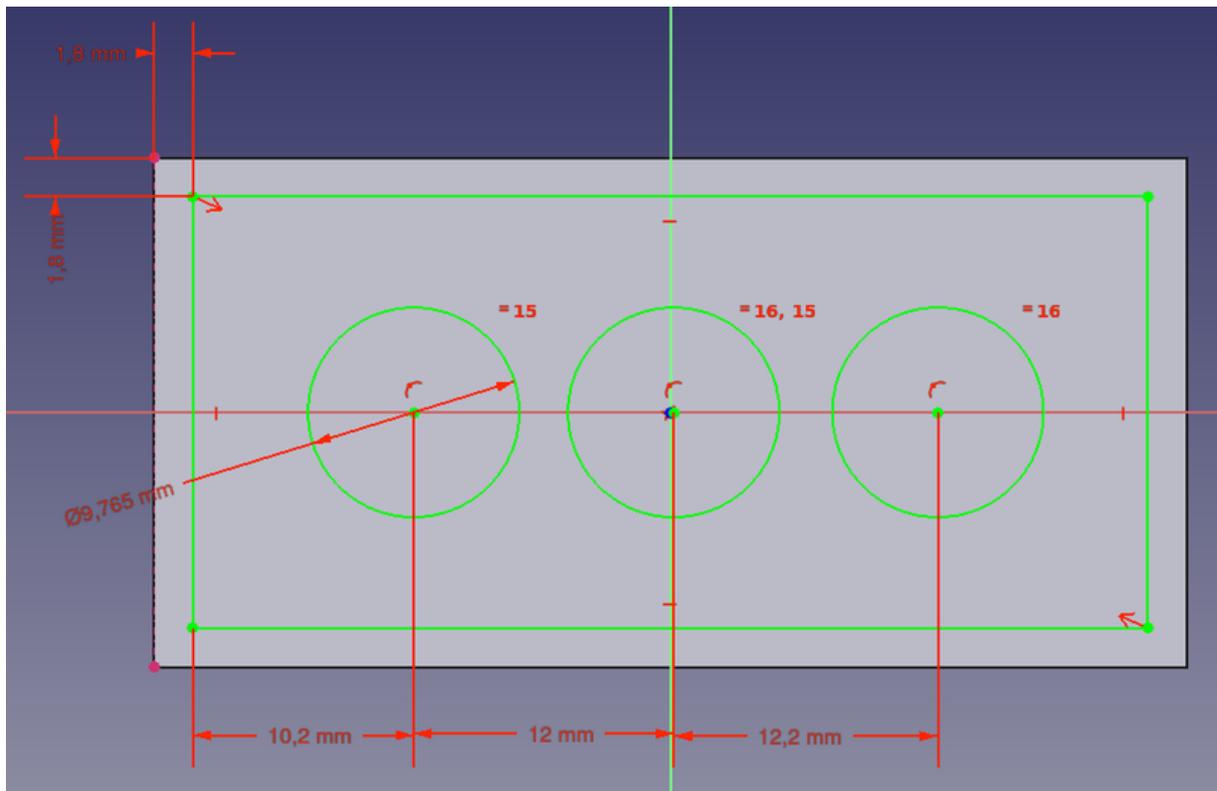
- Comme nous avons utilisé la face supérieure de notre bloc de base comme base de cette nouvelle esquisse, toute opération de conception de pièce qui lui est appliquée sera correctement construite sur la forme de base. Les deux cercles ne sont pas des objets indépendants ; ils seront extrudés directement à partir du bloc existant. Il s'agit là du principal avantage du travail dans l'atelier PartDesign : tant que vous vous assurez que chaque étape est construite sur la précédente, vous construisez effectivement un objet solide unique et cohésif.
- Nous pouvons maintenant dupliquer nos deux points quatre fois. Sélectionnez la dernière protrusion que nous venons de créer.
- Appuyez sur la touche  [Répétition linéaire](#).
- Donnez-lui une longueur de 36 mm (qui est la longueur totale dans laquelle nous voulons que nos copies tiennent), dans la direction de l'« axe horizontal de l'esquisse », et faites-en 4 occurrences :



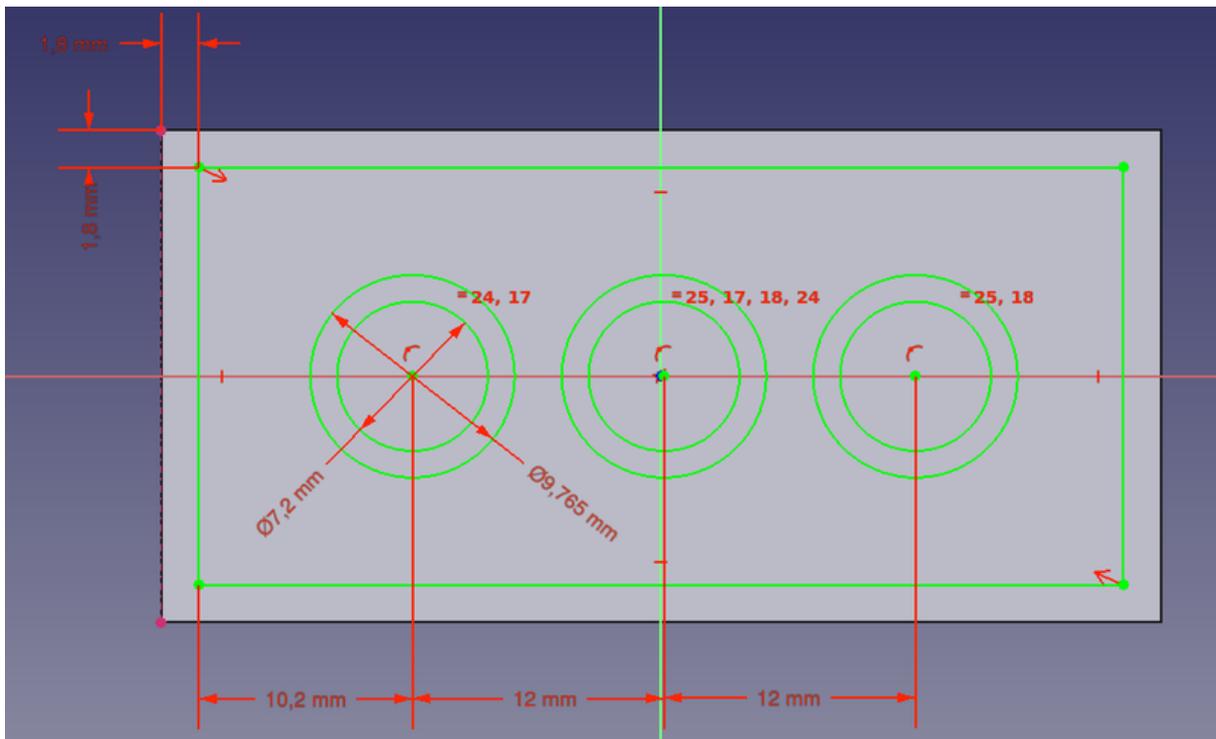
- Nous allons maintenant sculpter l'intérieur du bloc, en utilisant l'outil  [Cavité](#), qui est la version PartDesign de [Part Soustraction](#). Pour créer une cavité, nous allons créer une esquisse sur la face inférieure de notre bloc, qui sera utilisée pour enlever une partie du bloc.
- La face inférieure étant sélectionnée, appuyez sur la touche  [Créer une esquisse](#).
- Dessinez un  [rectangle](#) sur la face.
- Appliquez une  [contrainte de symétrie](#) en sélectionnant le point d'angle supérieur gauche du rectangle, puis le point d'origine (le point où les axes rouge et vert se croisent), et le point d'angle inférieur droit.
- En utilisant la  [géométrie externe](#), choisissez l'arête gauche de la face inférieure. Notez à nouveau qu'elle sera surlignée en rouge.
- Définissez la distance horizontale et verticale du rectangle par rapport au point supérieur droit à 1,8 en utilisant la  [contrainte de dimension](#).



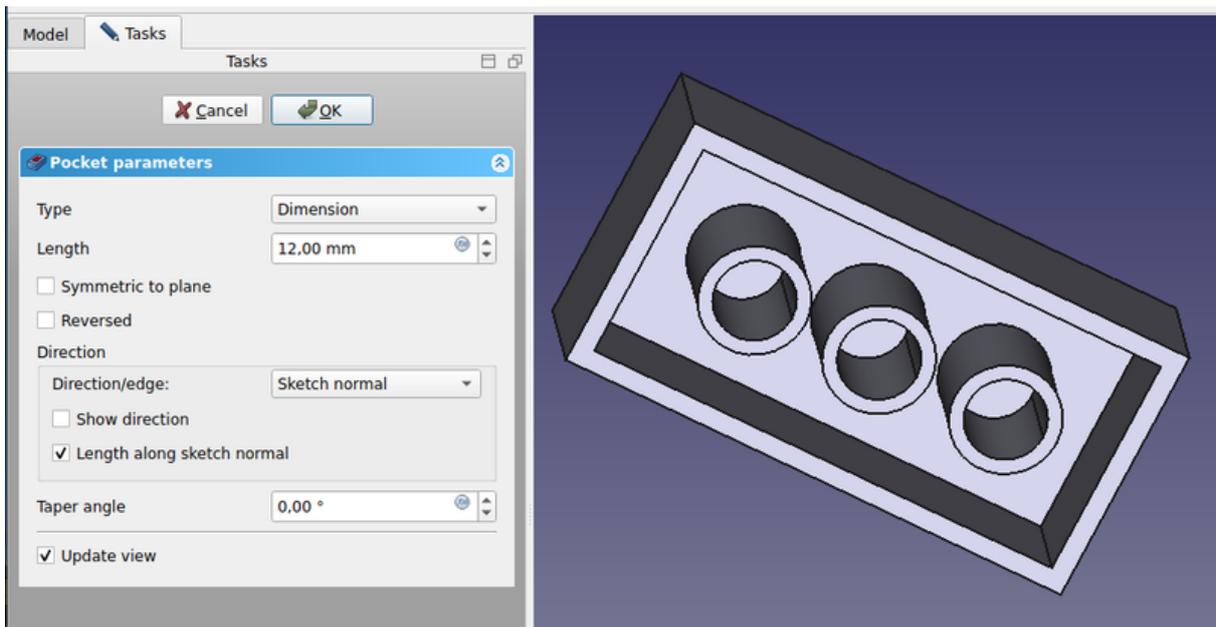
- Créez trois  [cercles](#) en s'assurant que leur centre est situé sur l'axe X (ligne rouge).
- En choisissant les trois cercles, appliquez une  [contrainte d'égalité](#).
- Définissez le diamètre d'un cercle à 9,765 mm.
- Définissez la distance centrale du cercle gauche par rapport à l'arête inférieure du rectangle que nous avons créé précédemment à 10,2 mm.
- Réglez la distance entre les cercles de gauche et du milieu à 12 mm. Répétez cette étape pour définir la même distance de 12 mm entre les cercles du milieu et de droite.



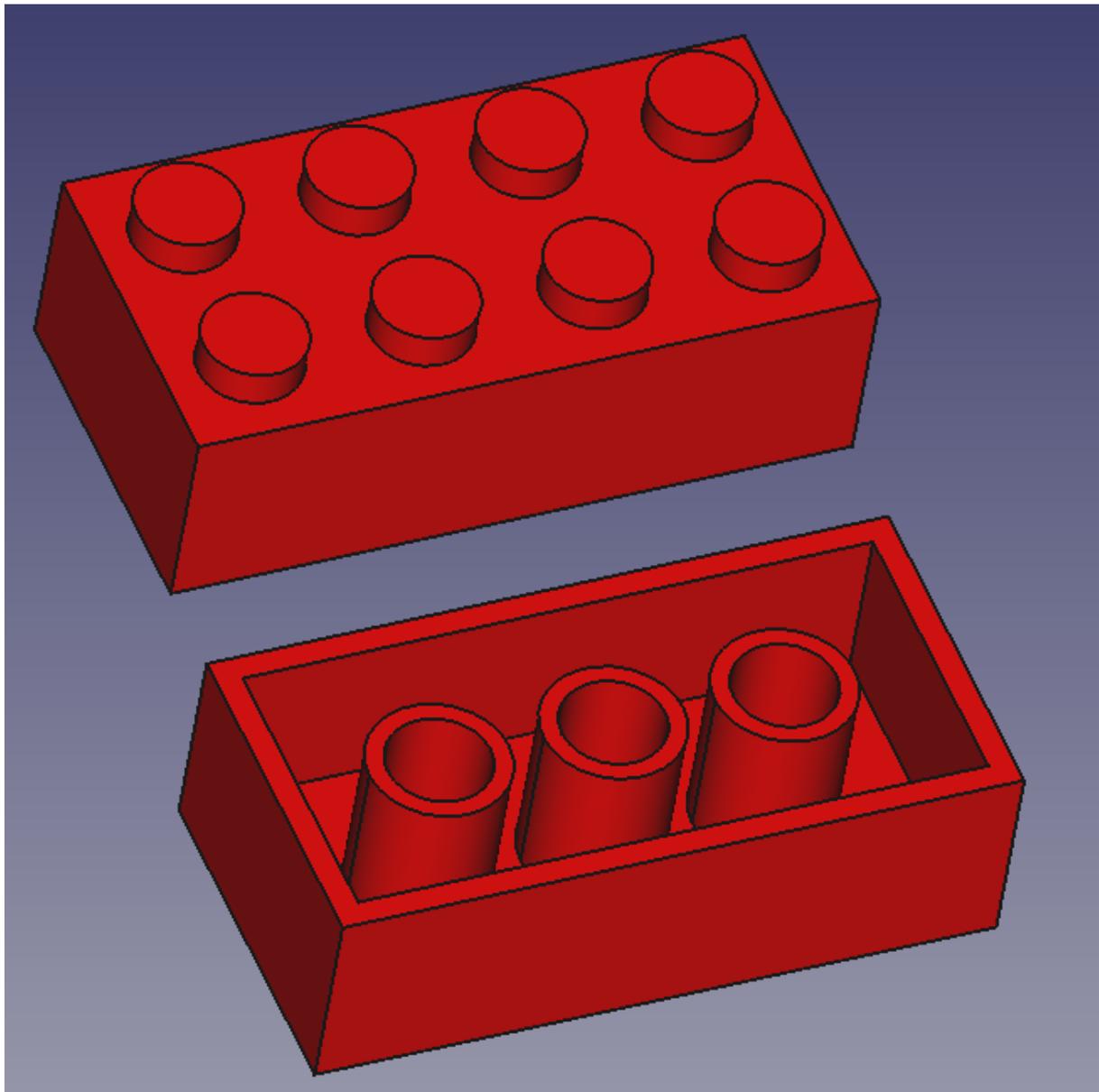
- Nous avons presque terminé.
- Créer trois autres  [cercles](#) en veillant à ce que chaque nouveau cercle soit concentrique à l'un des cercles dessinés précédemment. Vous pouvez également placer les nouveaux cercles n'importe où dans l'esquisse et utiliser la fonction  [contrainte de coïncidence](#) pour aligner leurs centres sur les centres des cercles existants.
- En choisissant les trois cercles, appliquez une  [contrainte d'égalité](#).
- Définissez le diamètre d'un cercle à 7,2 mm.
- Vous pouvez maintenant quitter l'esquisse.



- En choisissant le croquis terminé, utilisez l'outil  [Cavité](#) en réglant la longueur sur 12 mm.



- Ça y est. Notre brique est prête. Si vous souhaitez changer sa couleur, vous pouvez le faire en allant dans l'onglet **Vue**.



Vous remarquerez peut-être que l'historique de la modélisation dans l'arborescence est devenu très complet. Cette fonction est extrêmement précieuse, car elle nous permet de revoir et de modifier à tout moment n'importe quelle étape du processus de conception. Par exemple, adapter ce modèle pour créer une brique de 2x2 au lieu de 2x4 est un jeu d'enfant : quelques ajustements des dimensions et des occurrences du modèle suffisent. Cette même flexibilité nous permet de concevoir des pièces plus grandes et personnalisées qui ne font pas partie du jeu Lego original. La nature paramétrique de FreeCAD facilite la modification des modèles existants, ce qui nous permet d'adapter ou d'étendre la conception en fonction des besoins.

Mais nous pourrions aussi vouloir nous débarrasser de l'historique, par exemple si nous voulons modéliser un château avec cette brique, et ne voulons pas que cet historique soit répété 500 fois dans notre fichier.

Il existe deux façons simples de se débarrasser de l'historique, on utilise l'outil [Part Création de copie simple](#) de l'[atelier Part](#), qui créera une copie de notre pièce qui ne dépend plus de l'historique (vous pouvez supprimer l'historique complet après), l'autre façon est d'exporter la pièce dans un fichier STEP et de la réimporter.

Téléchargements

- Le modèle produit lors de cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/blob/master/files/lego.FCStd>

Lire plus d'informations

- [L'atelier Sketcher](#)
- [L'atelier PartDesign](#)
- [L'atelier Assembly2](#)

Manuel : Préparation des modèles pour l'impression 3D

L'un des principaux objectifs de FreeCAD est de concevoir des objets qui peuvent être transformés en produits physiques réels. Ces conceptions peuvent être partagées avec d'autres pour la fabrication ou, de plus en plus, exportées directement vers des [imprimantes 3D](#) ou des [fraiseuses à commande numérique et machines à commande numérique](#) pour une fabrication automatisée. Avec FreeCAD, vous pouvez créer des modèles précis et détaillés qui sont prêts pour diverses méthodes de production. Ce chapitre vous guidera dans le processus de préparation de vos modèles pour ces machines, en vous assurant qu'ils répondent aux spécifications nécessaires pour une fabrication réussie, que vous travailliez avec une équipe ou que vous gériez l'ensemble du processus vous-même.

Si vous avez été soigneux lors de la modélisation, la plupart des défis associés à l'impression 3D de votre modèle devraient déjà être minimisés. Les principaux aspects sur lesquels il faut se concentrer sont les suivants :

- **S'assurer que vos objets sont solides** : tout comme les objets du monde réel, vos modèles 3D doivent être solides. FreeCAD, en particulier avec l'atelier PartDesign, vous aide à vous assurer que vos modèles restent solides tout au long du processus de conception. Le logiciel vous avertit si une opération compromet la solidité de l'objet. En outre, l'atelier Part propose l'outil  [Vérifier la géométrie](#), qui vous permet d'identifier les défauts potentiels ou les problèmes susceptibles d'interférer avec le processus d'impression 3D.
- **Confirmation de l'exactitude des dimensions** : la précision est essentielle. Ce que vous concevez dans FreeCAD se traduira directement par des mesures dans le monde réel. Un millimètre dans FreeCAD est un millimètre dans l'objet physique, c'est pourquoi chaque dimension doit être soigneusement étudiée et vérifiée pour en garantir l'exactitude.
- **Gestion de la dégradation** : il est important de se rappeler qu'aucune imprimante 3D ou machines à commande numérique ne peut traiter directement les fichiers FreeCAD. Ces machines utilisent le G-Code, un langage machine avec différents dialectes en fonction de la machine ou du fournisseur. Le processus de conversion de votre modèle en G-Code peut souvent être effectué automatiquement par un logiciel de slicer, mais vous avez également la possibilité de le faire manuellement pour un meilleur contrôle. Cependant, au cours de cette conversion, une certaine perte de détails ou de qualité est inévitable, en particulier lors de la conversion du modèle en un format de maillage pour l'impression. Vous devez vous assurer que cette dégradation reste dans des limites acceptables et n'affecte pas la fonctionnalité ou l'apparence de votre objet final.
- **Compatibilité des formats d'exportation** : pour l'impression 3D, le format STL est le plus couramment utilisé, mais il convertit intrinsèquement votre modèle en un maillage de triangles, ce qui peut entraîner une perte de détails. Il est essentiel de choisir la bonne résolution lors de l'exportation vers STL, en trouvant un équilibre entre la conservation des détails et la taille du fichier. De même, pour l'usinage CNC, des formats comme STEP ou IGES sont préférables car ils conservent mieux que STL l'intégrité géométrique originale.

de la conception. Le choix du bon format garantit la précision de la conversion en G-Code.

- **Analyse et étalonnage du maillage** : avant d'exporter votre modèle vers un slicer ou un générateur de parcours d'une CNC, il est conseillé d'effectuer une analyse de maillage à l'aide de l'[atelier Mesh](#) de FreeCAD afin de détecter les irrégularités, les arêtes non pliables ou d'autres problèmes de maillage qui pourraient compliquer le processus de fabrication. En outre, même avec un modèle parfait, assurez-vous que votre imprimante 3D ou votre machine CNC est correctement calibrée (par exemple, pour le nivellement du lit, les réglages du moteur pas à pas ou la configuration de l'extrudeuse) afin d'éviter les problèmes de qualité dans le produit final.

Dans les sections suivantes, nous supposons que vous avez déjà pris soin de créer des modèles solides aux dimensions correctes. Nous allons maintenant nous concentrer sur la gestion du processus de conversion en G-Code, en veillant à ce que votre modèle conserve la qualité nécessaire pour l'impression 3D ou l'usinage CNC. En tenant compte de ces considérations, vous serez mieux équipé pour produire des objets physiques réussis directement à partir de vos modèles FreeCAD.

Exporter vers les slicers

La technique la plus courante pour préparer un modèle 3D à l'impression consiste à exporter l'objet 3D de FreeCAD vers un logiciel spécialisé appelé « slicer ». Le slicer génère du G-code en découpant le modèle en fines couches, que l'imprimante 3D suivra pour construire l'objet couche par couche. Étant donné que de nombreuses imprimantes 3D, en particulier les modèles fabriqués à la maison ou par des amateurs, ont des configurations uniques, les programmes de slicer offrent un large éventail de paramètres avancés. Ces réglages vous permettent de personnaliser des paramètres clés, tels que la hauteur des couches, la vitesse d'impression, la densité de remplissage et les structures de support, afin que le G-code soit adapté aux caractéristiques et aux capacités spécifiques de votre imprimante.

De nombreux slicers offrent également des fonctions de simulation et de validation de l'impression qui sont très utiles pour prévisualiser le processus d'impression. Vous pouvez visualiser le parcours de l'outil pour chaque couche, ce qui permet de détecter les problèmes potentiels tels que les surplombs qui peuvent nécessiter des supports ou les zones où le refroidissement peut être insuffisant. Cette validation avant impression garantit que votre modèle est correctement préparé avant le début de l'impression, ce qui permet d'éviter les échecs d'impression ou le gaspillage de matériau.

Les slicers incluent souvent des informations supplémentaires, telles que l'estimation du temps d'impression, de l'utilisation des matériaux et du coût en fonction du filament ou de la résine utilisés. Cela vous permet de prendre des décisions éclairées sur le processus d'impression et d'ajuster les paramètres dans un souci d'efficacité ou de conservation des matériaux. Bien que les subtilités de l'impression 3D, telles que l'étalonnage de la machine, la sélection des matériaux et le post-traitement, dépassent le cadre de ce guide, nous nous concentrerons sur la manière d'exporter correctement votre modèle FreeCAD et d'utiliser le logiciel de slicer pour s'assurer que le résultat est correct et optimisé pour votre imprimante spécifique.

Convertir des objets en maillage

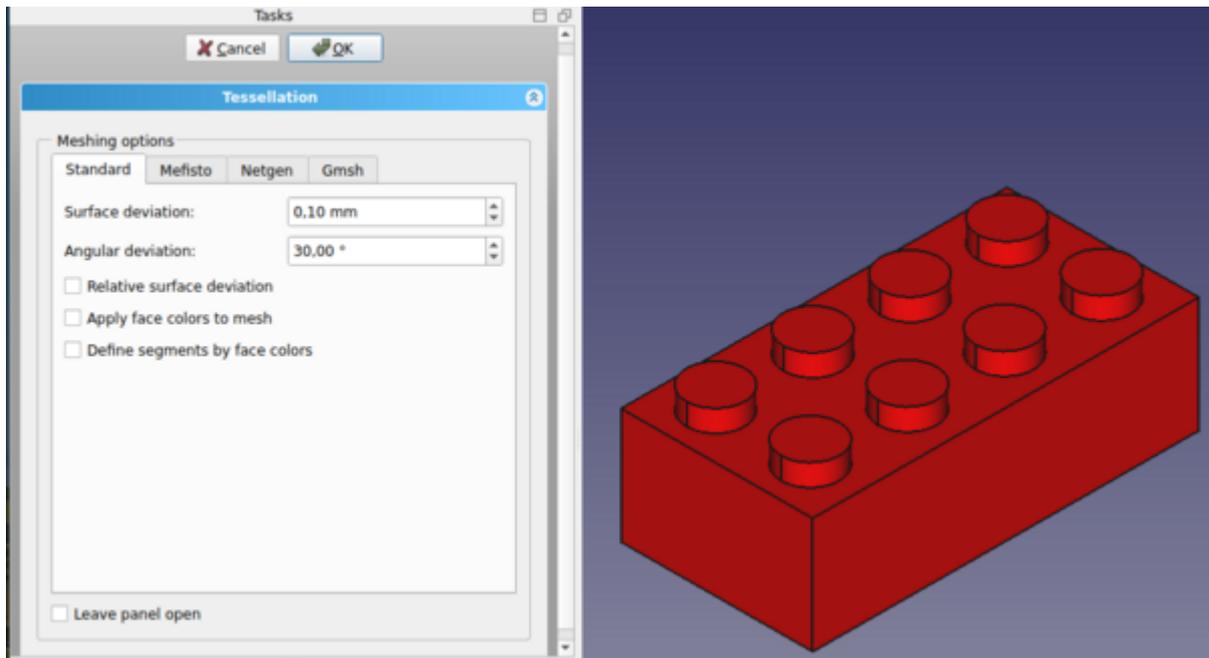
Aucun des slicers actuellement disponibles ne peut accepter directement la géométrie solide produite dans FreeCAD. Les slicers comme Cura et PrusaSlicer travaillent avec des formats basés sur des [maillages](#) tels que STL, OBJ ou 3MF, qui représentent la géométrie de la surface de l'objet à l'aide d'un réseau de triangles. Par conséquent, pour utiliser un modèle créé dans FreeCAD, il faut d'abord le convertir dans un format de maillage que ces slicers peuvent interpréter.

Le format le plus couramment utilisé pour l'impression 3D est le STL. L'une des raisons pour lesquelles le STL est privilégié est sa simplicité : il représente la géométrie 3D sous la forme d'un maillage de triangles sans inclure de détails complexes tels que les couleurs, les matériaux ou les textures. Cette approche minimaliste garantit la légèreté des fichiers STL et leur compatibilité avec la quasi-totalité des slicers et des imprimantes 3D, ce qui en fait le standard de l'industrie. Les fichiers OBJ et 3MF sont également pris en charge, mais ils peuvent contenir des informations supplémentaires telles que des textures et des matériaux, qui sont inutiles pour la plupart des tâches d'impression 3D et peuvent compliquer le processus de découpage.

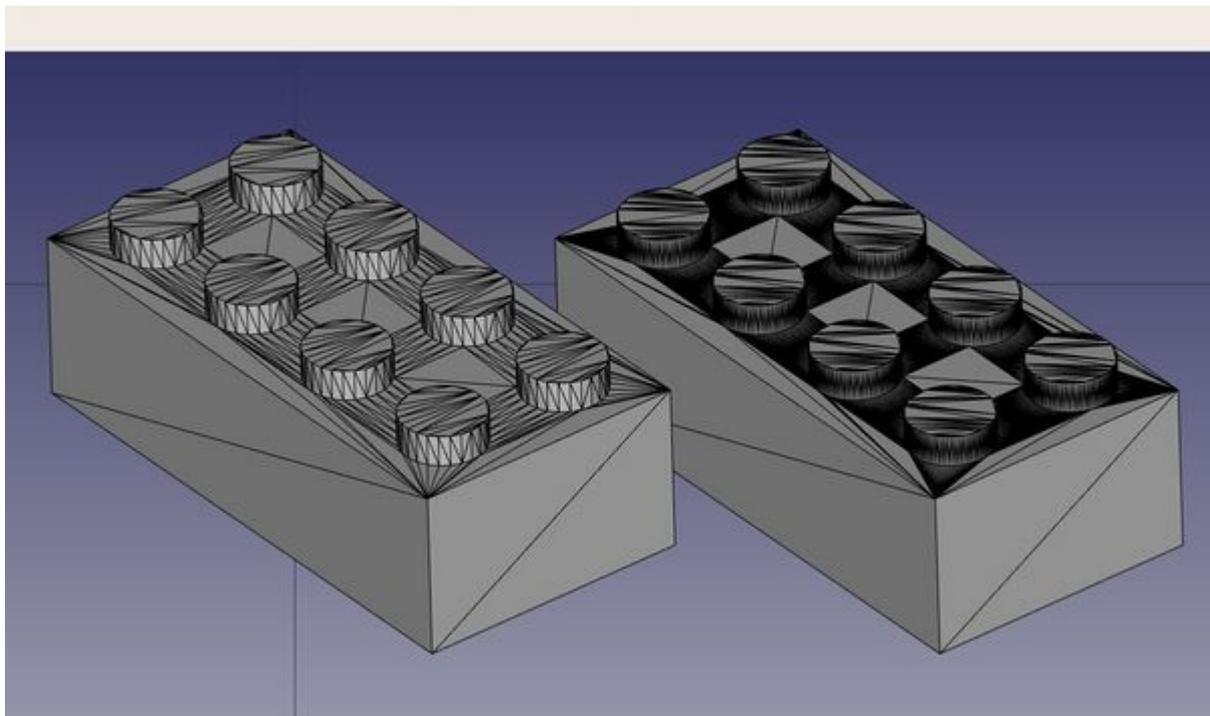
Heureusement, la conversion d'un objet solide en maillage dans FreeCAD est simple, même si la reconversion d'un maillage en solide est une opération plus compliquée. Au cours du processus de conversion, il est essentiel de garder à l'esprit qu'une certaine dégradation de la qualité du modèle peut se produire, en particulier lors de la réduction d'une géométrie complexe en un simple maillage triangulaire. Vous devez vous assurer que cette dégradation reste dans des limites acceptables pour maintenir la précision de votre objet imprimé.

Dans FreeCAD, l'[atelier Mesh](#) gère toutes les tâches liées au maillage. Cet atelier contient des outils non seulement pour la conversion entre les objets Part et Mesh, mais aussi pour l'analyse et la réparation des maillages. Bien que la manipulation des maillages ne soit pas l'objectif principal de FreeCAD, elle devient essentielle lors de la préparation des modèles pour l'impression 3D. Les objets Mesh sont largement utilisés dans d'autres applications, et l'atelier Mesh vous permet de gérer et d'ajuster entièrement ces objets, en veillant à ce qu'ils soient prêts pour l'étape suivante du processus d'impression.

- Convertissons la pièce Lego que nous avons créée dans le dernier chapitre en un maillage STL. La géométrie peut être téléchargée à la fin de ce chapitre.
- Ouvrez le fichier FreeCAD contenant la pièce Lego.
- Passez à l'[atelier Mesh](#)
- Sélectionnez la brique Lego
- Sélectionnez le menu **Maillages** → **Créer un maillage à partir d'une forme**.
- Un panneau de tâches s'ouvre avec plusieurs options. Certains algorithmes de maillage supplémentaires (Mefisto ou Netgen) peuvent ne pas être disponibles, en fonction de la façon dont votre version de FreeCAD a été compilée. L'algorithme de maillage Standard sera toujours présent. Il offre moins de possibilités que les deux autres, mais est tout à fait suffisant pour les petits objets qui entrent dans la taille d'impression maximale d'une imprimante 3D.



- Sélectionnez le maillage **standard** et laissez la valeur de déviation à la valeur par défaut de **0.10**. Appuyez sur **OK**.
- Un objet mesh sera créé, exactement par dessus notre objet solide. Pour comparer les deux vous pouvez soit cacher le solide, soit déplacer l'un des objets par rapport à l'autre.
- Modifiez la propriété **Affichage** → **Style de représentation** du nouvel objet maillage en **Filaire ombré** afin de voir comment s'est déroulée la triangulation.
- Si vous n'êtes pas satisfait et pensez que le résultat est trop grossier, vous pouvez répéter l'opération, en abaissant la valeur de déviation. Dans l'exemple ci-dessous, le maillage gauche a utilisé la valeur par défaut de **0.10**, tandis que celui de droite utilise **0.01** :



Dans la plupart des cas, les valeurs par défaut donneront un résultat satisfaisant.

- Nous pouvons maintenant exporter notre maillage à un format de maillage, par exemple [STL](#), actuellement le format le plus utilisé dans l'impression 3D, en utilisant le menu **Fichier** → **Exporter** et choisir le format de fichier STL.

Dans FreeCAD, l'atelier Mesh propose plusieurs algorithmes pour convertir un modèle solide en maillage, notamment Standard, Mefisto, Netgen et Gmsh. L'algorithme Standard est couramment utilisé pour les objets de petite et moyenne taille, car il offre un équilibre entre la vitesse et la qualité du maillage. Lors de la création d'un maillage, deux paramètres essentiels sont l'écart de surface et l'écart angulaire. L'écart de surface détermine dans quelle mesure le maillage suit la géométrie d'origine. Des valeurs plus faibles permettent d'obtenir un maillage plus fin et plus précis, mais peuvent entraîner une augmentation de la taille des fichiers. La déviation angulaire définit l'ampleur de la déviation autorisée en fonction des changements d'angle du modèle, en particulier pour les courbes et les arêtes vives. D'autres options, telles que la déviation de surface relative, vous permettent d'ajuster la précision de manière dynamique en fonction de l'échelle du modèle, et des fonctions telles que l'application de couleurs de face ou la définition de segments de maillage par couleur sont utiles pour le rendu avancé ou le regroupement de différentes régions du modèle. Une fois le maillage généré, il peut être exporté dans des formats tels que STL, OBJ ou 3MF, qui sont essentiels pour préparer les modèles à l'impression 3D. La qualité du maillage est essentielle pour que les imprimantes 3D interprètent correctement le modèle. Le choix du bon algorithme de maillage et des paramètres de déviation peut donc avoir une incidence considérable sur le résultat final de l'impression.

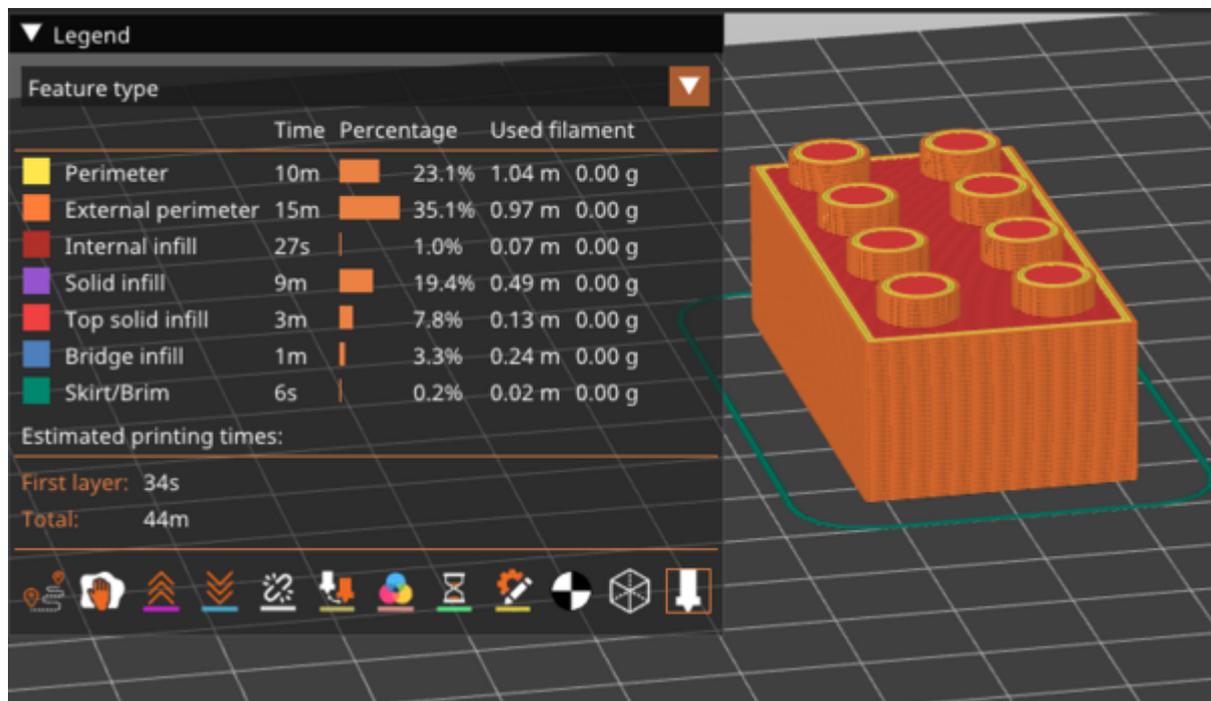
Utiliser PrusaSlicer

[PrusaSlicer](#) est une application qui convertit les objets STL, OBJ et 3MF en G-code qui peut être envoyé directement aux imprimantes 3D. Comme FreeCAD, il est gratuit, open-source et disponible sur Windows, Mac OS et Linux. Bien qu'il soit développé par Prusa Research et optimisé pour les imprimantes 3D Prusa, PrusaSlicer peut être utilisé avec presque toutes les imprimantes 3D, ce qui le rend polyvalent pour une large gamme de machines. PrusaSlicer est basé sur Slic3r, le logiciel slicer original, mais avec des améliorations significatives et des mises à jour plus fréquentes. Slic3r n'est plus activement mis à jour, tandis que PrusaSlicer continue d'évoluer, ajoutant de nouvelles fonctionnalités telles que des hauteurs de couche adaptatives, des supports d'arbre et des stratégies d'impression améliorées.

Configurer correctement un slicer pour l'impression 3D est un processus complexe qui nécessite une bonne compréhension des capacités de votre imprimante 3D. Bien que la génération de G-code sans cette connaissance puisse résulter en un fichier qui ne fonctionne pas bien sur d'autres imprimantes, PrusaSlicer fournit toujours un excellent moyen de vérifier que votre fichier STL est correctement formaté et imprimable. Les fonctions de simulation du slicer vous permettent de prévisualiser les chemins du G-code et de vérifier tout problème d'impression potentiel avant de commencer l'impression réelle.

Voici notre fichier STL exporté, ouvert dans PrusaSlicer. En appuyant simplement sur le bouton **slice**, le logiciel divise votre modèle en couches, génère les parcours d'outils pour l'imprimante 3D et applique les paramètres de vitesse et de température

nécessaires. Il calcule le remplissage, les structures de soutien et les périmètres, puis crée le G-code, qui contient des instructions détaillées pour l'imprimante. Vous pouvez prévisualiser le modèle découpé couche par couche, vérifier le temps d'impression estimé et l'utilisation du filament, et enfin enregistrer ou envoyer le G-code à votre imprimante pour le processus d'impression proprement dit.



Outre PrusaSlicer, il existe plusieurs autres logiciels de découpe disponibles pour l'impression 3D. [Cura](#), développé par Ultimaker, est l'un des slicers open-source les plus populaires et prend en charge une large gamme d'imprimantes avec une personnalisation poussée. [Simplify3D](#) est un slicer payant connu pour ses fonctionnalités avancées et sa génération efficace de parcours d'outils. [MatterControl](#) est un slicer open-source qui inclut également des outils de CAO de base, tandis que [IdeaMaker](#) offre une interface conviviale avec des hauteurs de couche adaptatives, développée par Raise3D. Enfin, [OrcaSlicer](#), une option open-source plus récente basée sur PrusaSlicer et Bambu Studio, offre des fonctionnalités supplémentaires pour diverses imprimantes. Chaque slicer possède des atouts uniques, ce qui fait que le meilleur choix dépend des modèles d'imprimantes spécifiques et des exigences d'impression.

Générer du G-code

L'[atelier CAM](#) de FreeCAD fournit des options avancées pour générer du G-code directement pour les machines CNC, offrant une plus grande flexibilité et un meilleur contrôle par rapport aux outils de découpage automatique tels que ceux utilisés pour l'impression 3D. Alors que les slicers d'impression 3D peuvent convertir automatiquement un modèle en G-code avec un minimum de données, le fraisage CNC nécessite une implication beaucoup plus importante de l'utilisateur pour assurer un contrôle précis des parcours d'outils, des vitesses, des profondeurs et d'autres paramètres d'usinage. L'atelier CAM est donc essentiel pour les tâches qui exigent un G-code précis, en particulier pour le fraisage CNC, où la complexité de la

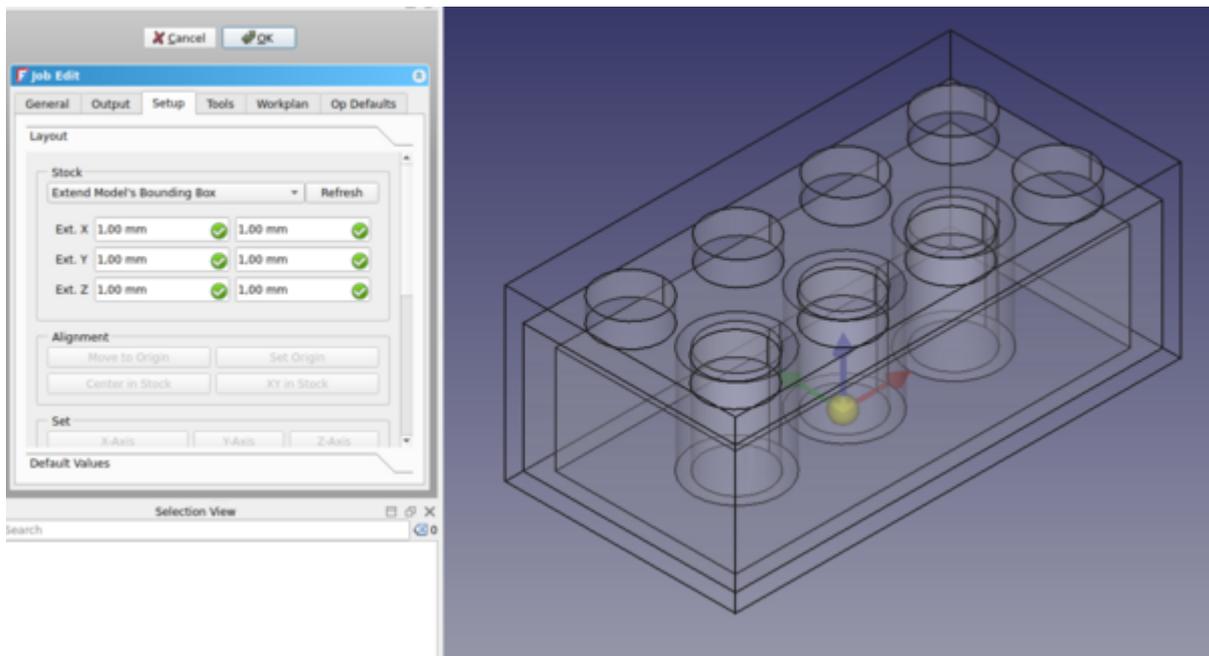
machine et la variété des opérations (comme la coupe, le perçage et le contournage) exigent une planification minutieuse.

Dans l'atelier CAM, la génération de parcours en G-code est hautement personnalisable. Elle comprend des outils permettant de générer des parcours machine complets pour diverses opérations ou, au contraire, de construire des segments partiels de G-code et de les assembler pour obtenir une opération de fraisage complète. Cette approche modulaire vous permet de personnaliser chaque étape du processus d'usinage, en optimisant les parcours d'outils en fonction de l'efficacité, du type de matériau et des capacités spécifiques de la machine.

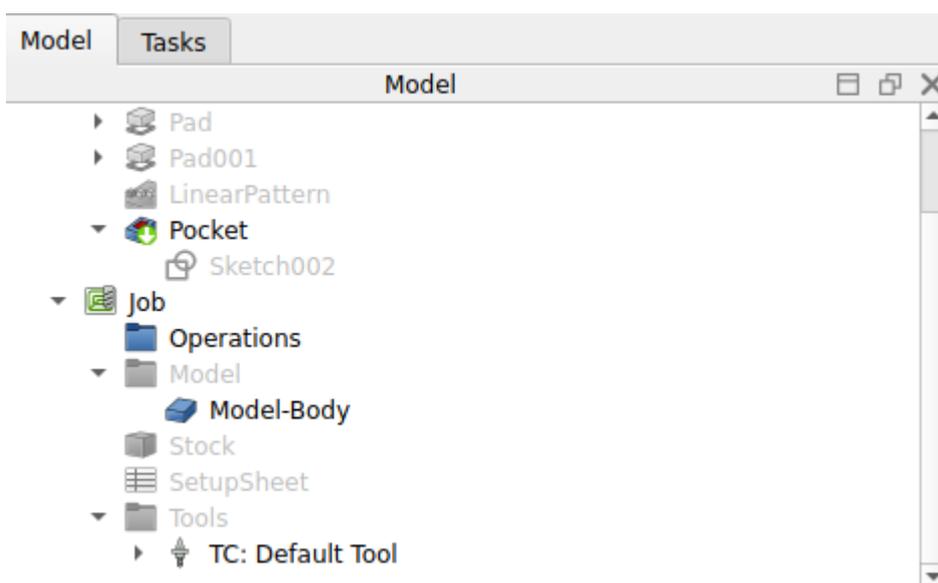
Le processus de CAM est en effet beaucoup plus complexe que l'impression 3D, car les machines à commande numérique utilisent des outils différents et doivent tenir compte de l'enlèvement de matière, de la géométrie de l'outil et des marges de sécurité, qui sont tous configurés manuellement. Dans FreeCAD, la construction d'un projet CAM simple nécessite de définir des parcours d'outils, d'ajuster les profondeurs de coupe, de sélectionner les outils appropriés et de configurer les décalages, les avances et les vitesses de travail. Contrairement aux logiciels de slicer, qui gèrent la plupart de ces opérations automatiquement, l'atelier CAM vous laisse le contrôle, ce qui le rend hautement personnalisable, mais aussi plus complexe.

Bien que la génération de parcours de fraisage CNC soit un sujet trop vaste pour être traité en détail ici, nous allons montrer comment créer un projet CAM simple dans FreeCAD. Bien que nous ne nous attarderons pas sur tous les détails de l'usinage CNC dans le monde réel, ce guide vous présentera les étapes essentielles, en mettant l'accent sur le niveau d'entrée requis pour garantir des résultats précis et efficaces. Cette complexité supplémentaire est essentielle pour les projets CNC, où la précision et la personnalisation sont essentielles pour obtenir les résultats d'usinage souhaités.

- Chargez le fichier contenant notre pièce de Lego, et passez à l' [atelier CAM](#).
- Appuyez sur le bouton  [Créer une tâche](#) et sélectionnez notre pièce de lego.
- Comme cette section n'a pas pour but de fournir un tutoriel approfondi de l'atelier CAM, nous utiliserons les valeurs par défaut. Si vous souhaitez un tutoriel plus détaillé, veuillez vous référer à [CAM Tutoriel pas à pas pour l' impatient](#). Gardez à l'esprit que dans l'atelier de CAM, un corps pour le stock est automatiquement créé autour de votre objet, représentant la matière première qui sera usinée. Pour l'instant, ce corps de stock s'étend sur 1 mm dans toutes les directions à partir de l'objet.



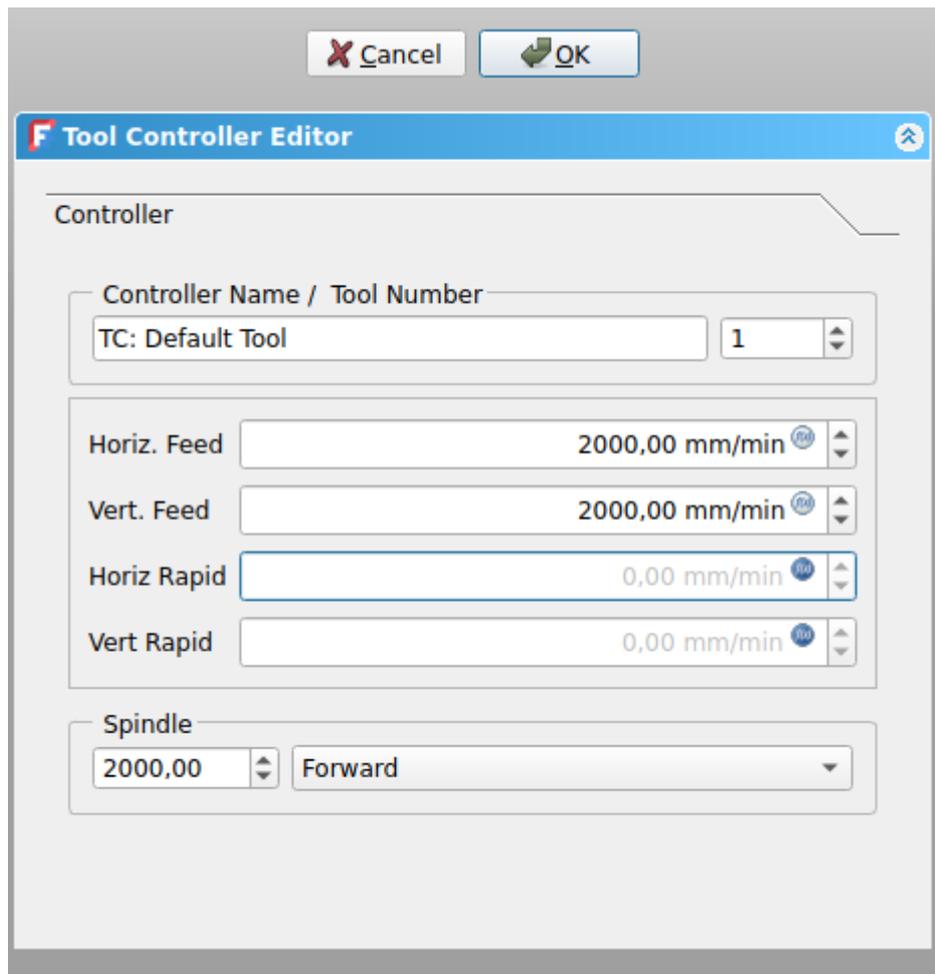
- La première étape consiste à supprimer les matériaux inutiles autour de notre objet. À ce stade, nous partons d'un bloc solide de matériau brut et nous devons découper la brique de Lego à partir de ce bloc. Ce processus consiste à définir les parcours d'outils qui découperont progressivement l'excès de matière, en laissant derrière eux la forme souhaitée de la brique Lego.
- L'image suivante montre la configuration de l'atelier CAM de FreeCAD pour l'usinage d'un bloc Lego. L'arborescence du modèle comprend des opérations de modélisation solide telles que Protrusion, Cavité et Répétition linéaire, qui ont été utilisées pour façonner la pièce. Une tâche est créée, contenant des parcours d'outils sous les opérations qui définissent la manière dont la matière sera enlevée du stock. L'outil par défaut est sélectionné pour l'usinage et le corps-modèle représente la pièce 3D sur laquelle on travaille. Cette configuration prépare l'objet à la génération du G-code pour contrôler la CNC.



- Avant de commencer à découper le matériau excédentaire, nous allons procéder à quelques ajustements de l'outil de fraisage qui sera utilisé. Bien que

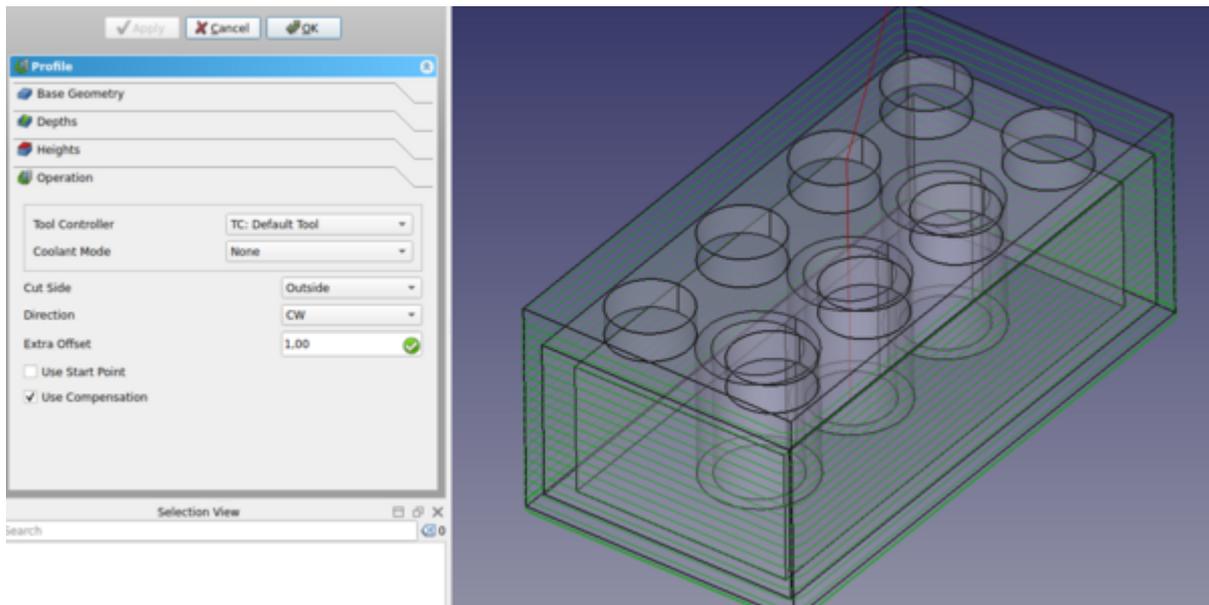
l'atelier CAM vous permette de définir des outils personnalisés, nous allons, pour des raisons de simplicité, modifier l'outil par défaut. Ainsi, les paramètres seront optimisés pour notre projet sans qu'il soit nécessaire de créer un nouvel outil à partir de zéro.

- Cliquez sur le texte **TC:Outil par défaut**. Cela ouvrira l'**éditeur du contrôleur d'outil**. Modifiez les vitesses d'avance et les vitesses de broche comme indiqué sur l'image. Les vitesses d'avance pour la coupe horizontale et verticale sont réglées sur 2000 mm/min, tandis que la vitesse de la broche est réglée sur 2000 RPM avec une rotation vers l'avant. Ces paramètres contrôlent le mouvement et la vitesse de coupe de l'outil pendant le processus d'usinage.



- Double-cliquez sur l'outil lui-même et modifiez son diamètre à 1 mm.
- Nous sommes maintenant prêts à retirer l'excédent de matériau du bloc et à découper progressivement la géométrie Lego. Ce processus impliquera les parcours d'outils que nous avons définis, afin de s'assurer que la forme finale correspond à la conception prévue.
- Cliquez sur l'option  [Profilage](#). Cette option est utilisée pour découper le matériau inutile autour du périmètre de la pièce, en façonnant efficacement les limites extérieures pour obtenir les dimensions générales de la pièce Lego.
- Normalement, vous n'aurez pas à modifier les valeurs par défaut, à l'exception du **Décalage supplémentaire** situé dans l'onglet Opération. Réglez cette option à 1 mm pour vous assurer que l'objet restant correspond correctement aux limites du Lego.

- Une fois que vous avez appuyé sur **appliquer**, vous devriez être en mesure de voir ces lignes vertes autour de l'objet. Ces lignes visualisent le parcours que notre objet de découpe suivra lors de la découpe du bloc initial.



- L'étape suivante consiste à créer les 6 cylindres extrudés sur la face supérieure du bloc Lego.
- Choisissez la face supérieure et cliquez sur le bouton  [Créer une poche](#). Dans l'onglet **Extensions**, activez les Extensions et cliquez sur le bord de la face supérieure (il devrait normalement être automatiquement ajouté dans la boîte de longueur par défaut).
- Enfin, dans l'onglet **Opération**, entrez -1.5 mm dans la case **Extension de passage** et changez l'option de motif en Décalage du zigzag.
- Appuyez sur **appliquer** et fermez l'onglet.
- De la même manière, nous pouvons créer les trois cylindres au bas de la pièce Lego.
- Nous pouvons facilement visualiser les étapes suivies pendant le fraisage de l'objet en utilisant l'option  [Simulateur GL](#).

Téléchargements

- Le fichier STL généré dans cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/lego.stl>
- Le fichier généré lors de cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/path.FCStd>
- Le fichier G-code généré dans cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/lego.gcode>

Lire plus d'informations

- [L'atelier Mesh](#)
- [Le format de fichier STL](#)
- [Slic3r](#)
- [Cura](#)
- [L'atelier Cura](#)
- [L'atelier CAM](#)
- [Camotics](#)

Vidéos

- [Comment utiliser FreeCAD pour l'impression 3D | Utilisation de la branche Realthunder](#) Une playlist de vidéos par Maker Tales sur la façon d'utiliser FreeCAD pour l'impression 3D.

Manuel : Génération de dessins 2D

Lorsque votre modèle ne peut pas être imprimé ou fraisé directement par une machine, par exemple s'il est trop grand (un bâtiment) ou qu'il nécessite un assemblage manuel une fois les pièces prêtes, vous devrez généralement expliquer à une autre personne comment le faire. Dans les domaines techniques (ingénierie, architecture, etc.), cela se fait généralement avec des dessins. Les dessins sont remis à la personne responsable de l'assemblage du produit final et vous expliquent comment procéder.

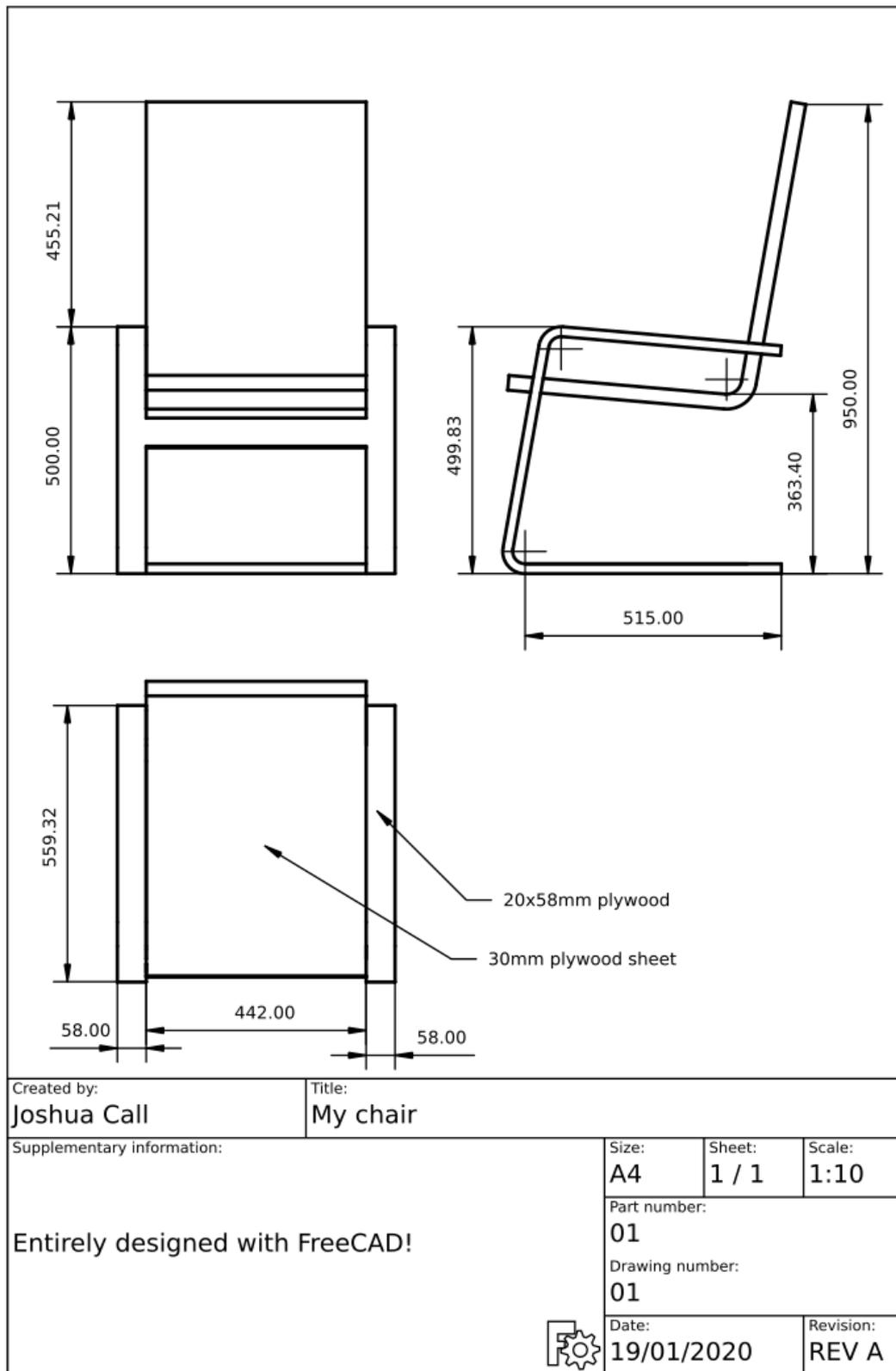
Les exemples typiques sont les instructions Ikea, les dessins d'architecture ([Dessins d'architecture](#)) et les plans ([blueprints](#)). Ces dessins contiennent généralement non seulement le dessin lui-même, mais aussi de nombreuses annotations, telles que les textes, dimensions, chiffres, symboles qui aideront les autres à comprendre ce qui doit être fait et comment.

Dans FreeCAD, l'atelier chargé de faire de tels dessins est l' [Atelier TechDraw](#).

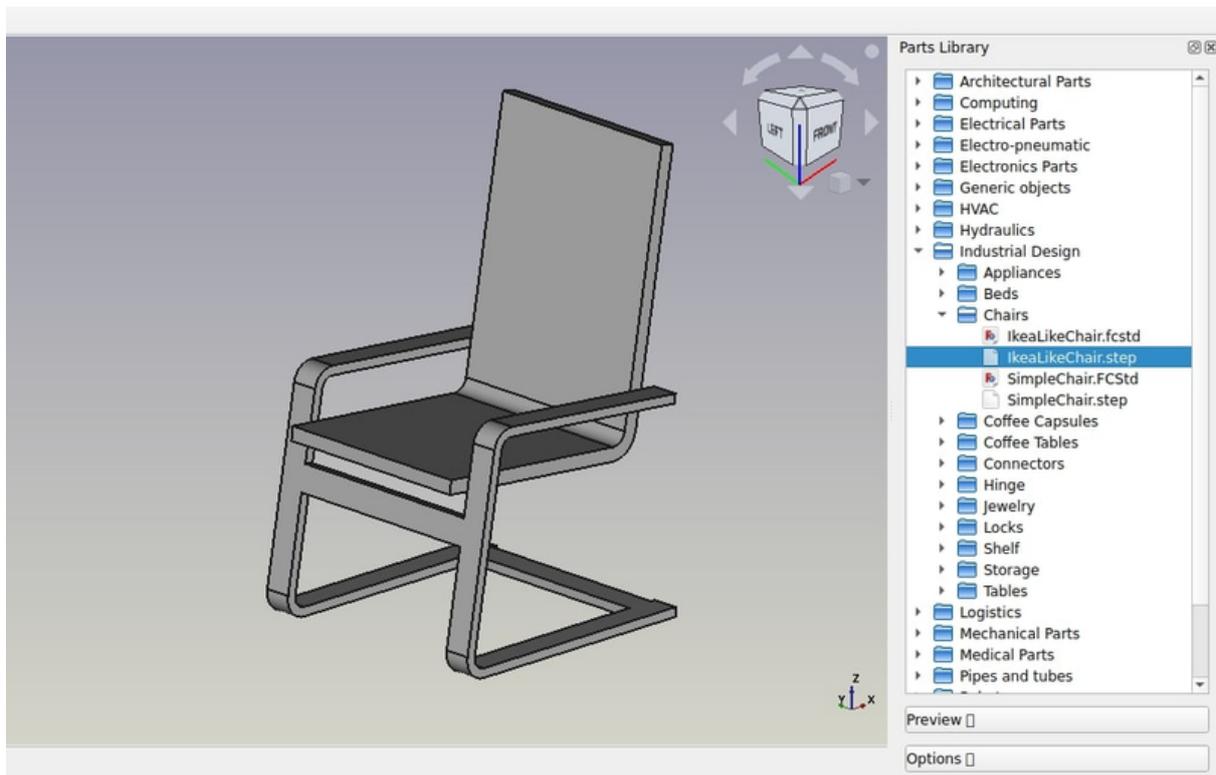
L'atelier TechDraw vous permet de créer des feuilles, qui peuvent être vides ou utiliser un [TechDraw Modèle](#) pour avoir déjà une série d'éléments sur la feuille, comme des bordures et un titre. Sur ces feuilles, vous pouvez ensuite placer des vues des objets 3D que vous avez modélisés précédemment et configurer la façon dont ces vues apparaîtront sur la feuille. Vous pouvez également placer toutes sortes d'annotations sur la feuille, telles que les dimensions, les textes et autres symboles couramment utilisés dans les dessins techniques.

Les feuilles de dessin, une fois complétées, peuvent être imprimées ou exportées sous forme de fichiers [SVG](#), PDF ou [DXF](#).

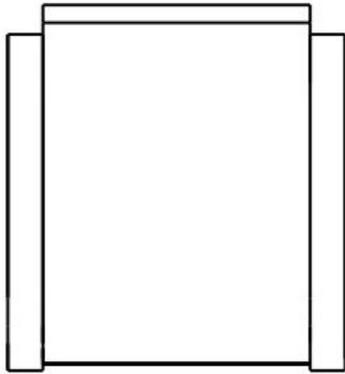
Dans l'exercice suivant, nous verrons comment créer un dessin simple d'un modèle de chaise trouvé dans la [bibliothèque FreeCAD](#) (Industrial Design → Chairs → IkeaLikeChair). La bibliothèque FreeCAD peut facilement être ajoutée à votre installation FreeCAD (reportez-vous au chapitre de ce [Manuel : Installation](#)), ou vous pouvez télécharger simplement le modèle à partir de la page Web de la bibliothèque ou via le lien direct fourni au bas de ce chapitre.



- Chargez le fichier IkeaLikeChair à partir de la bibliothèque. Vous pouvez choisir entre la version [.FCStd](#), qui chargera l'historique de la modélisation complète, ou la version [.step](#), qui ne créera qu'un seul Objet, sans l'historique. Comme nous n'aurons plus besoin de modéliser maintenant, il est préférable de choisir la version [.step](#), car le fichier sera plus facile à manipuler.

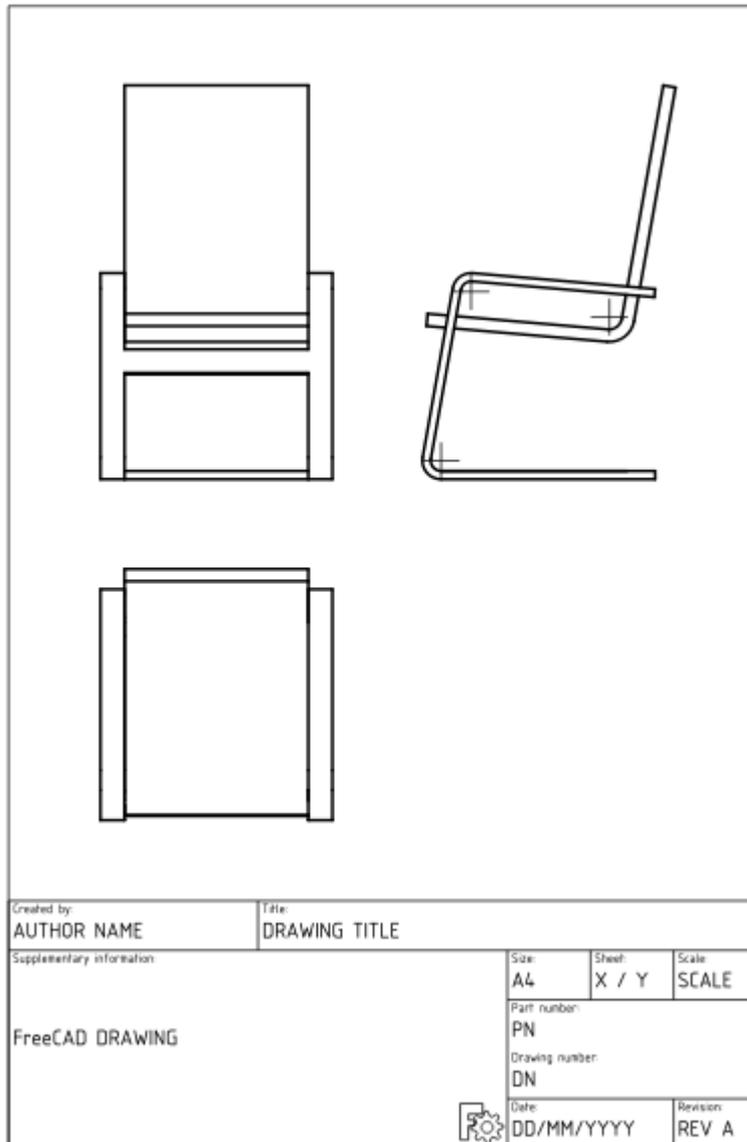


- Passer à l'  [atelier TechDraw](#)
- Appuyez sur le bouton  [TechDraw Page à partir d'un modèle](#).
- Sélectionnez le modèle **A4 Portrait ISO7200TD**. Un nouvel onglet s'ouvrira dans votre fenêtre FreeCAD montrant la nouvelle page.
- Dans la [vue en arborescence](#) (ou dans l'onglet modèle), sélectionnez le modèle de chaise. Il sera très probablement nommé quelque chose comme "Open CASCADE STEP translator."
- Appuyez sur le bouton  [TechDraw Vue](#).
- Un objet Vue sera créé sur notre page. Sélectionnez l'objet Vue dans l'arborescence puis donnez à la vue les [propriétés](#) dans l'onglet de données de la vue combinée:
 - Sous la catégorie Base:
 - X: 70 mm
 - Y: 120 mm
 - Rotation : 0
 - Échelle : 0,1
 - Sous la catégorie Projection (appuyez sur la flèche déroulante pour modifier individuellement les composants x, y et z de ces propriétés):
 - Direction : [0 0 1]
 - XDirection : [0 -1 0] (changez d'abord le champ y, puis le champ x)
- Nous avons maintenant une belle vue de dessus de notre chaise. Appuyez sur le bouton  [TechDraw Bascule des cadres](#) pour désactiver les cadres, étiquettes et sommets de vue.



Created by: AUTHOR NAME	Title: DRAWING TITLE		
Supplementary information: FreeCAD DRAWING	Size: A4	Sheet: X / Y	Scale: SCALE
	Part number: PN		
	Drawing number: DN		
	Date: DD/MM/YYYY		Revision: REV A

- Répétez l'opération deux fois pour créer deux autres vues. Nous allons définir leurs valeurs X et Y, qui indiquent la position de la vue sur la page, afin de les distinguer de la vue de dessus, et leur orientation, pour créer différentes orientations de vue. Donnez à chaque nouvelle vue les propriétés suivantes :
 - View001 (vue de face) : X: 70, Y: 220, Scale: 0.1, Rotation: 0, Direction: (-1,0,0), XDirection: (0,-1,0)
 - View002 (vue latérale) : X: 150, Y: 220, Scale: 0.1, Rotation: 0, Direction: (0,-1,0), XDirection: (1,0,0)
- Nous obtenons alors la page suivante :

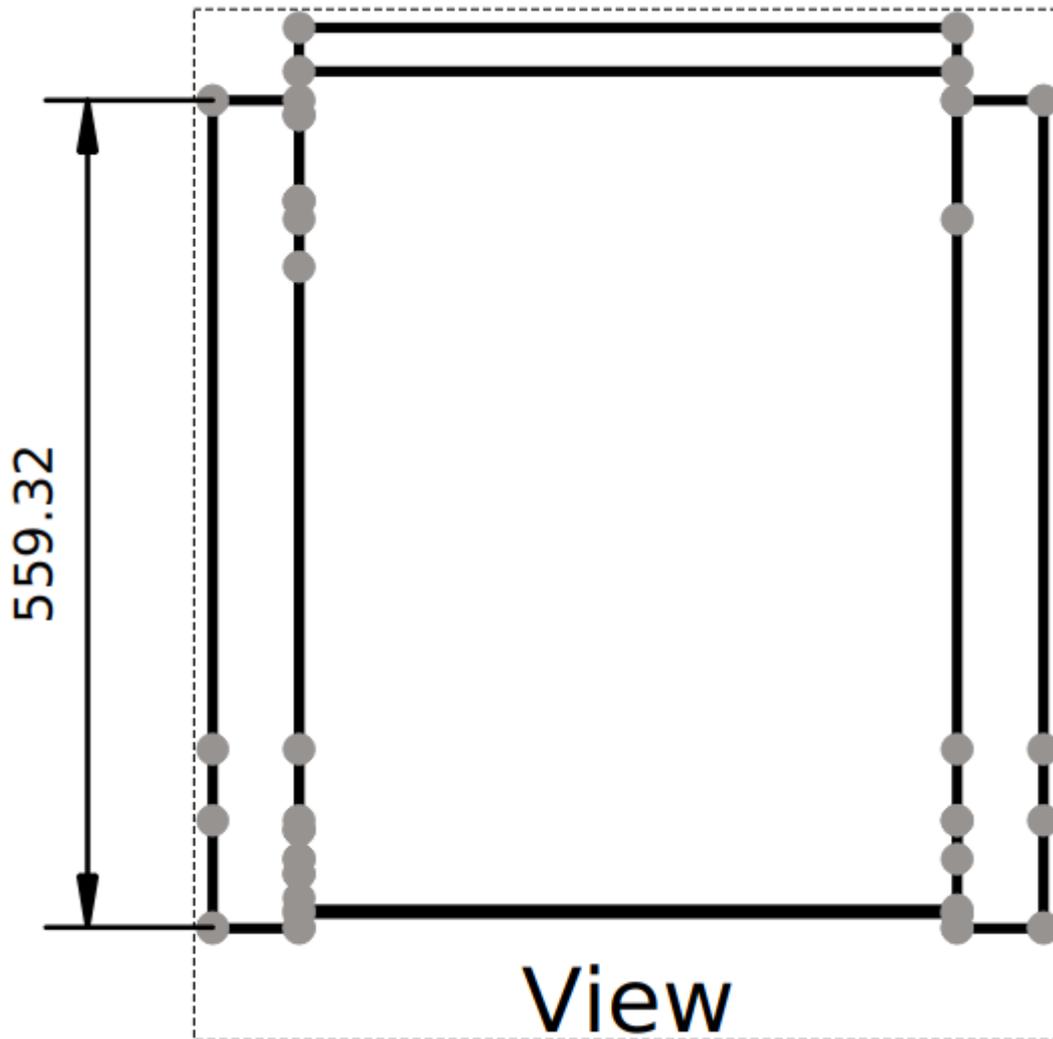


- Notez qu'il peut y avoir des moyens plus faciles d'obtenir les vues que vous souhaitez. Vous pouvez simplement [faire pivoter](#) la vue 3D de votre modèle, et une fois que vous avez la vue souhaitée, sélectionnez le modèle dans l'arborescence et appuyez sur  Nouvelle Vue. Cela insérera automatiquement une vue avec les propriétés de rotation et de direction souhaitées. Vous pouvez également utiliser l'outil  [TechDraw Groupe de projection](#).
- Nous pouvons modifier l'aspect de nos vues si nous le voulons, par exemple, nous pouvons changer leur propriété **Line Width** (Largeur de ligne) (sous l'onglet Vue dans la vue combinée) à 0,5.

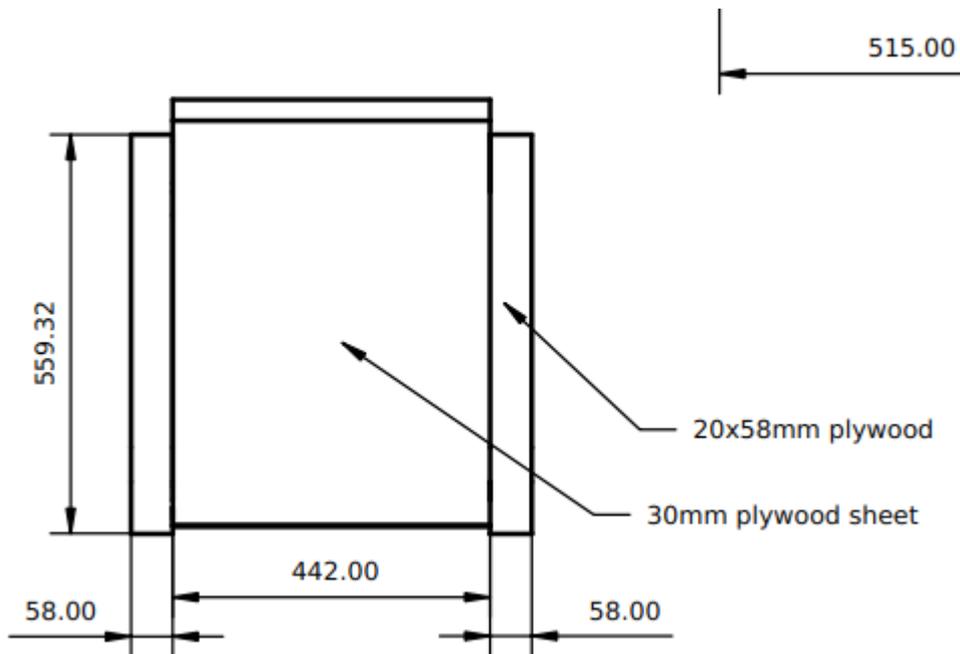
Nous allons maintenant placer des dimensions et des indications sur notre dessin. Il existe deux façons d'ajouter des cotes à un modèle: l'une consiste à placer les cotes à l'intérieur du modèle 3D, à l'aide de l'outil  [Draft Dimension](#) de l'[atelier Draft](#) puis en plaçant une vue de ces dimensions sur notre feuille avec l'outil  [TechDraw Vue](#)

Draft. L'autre consiste à faire les choses directement sur la feuille TechDraw. Nous utiliserons cette dernière méthode.

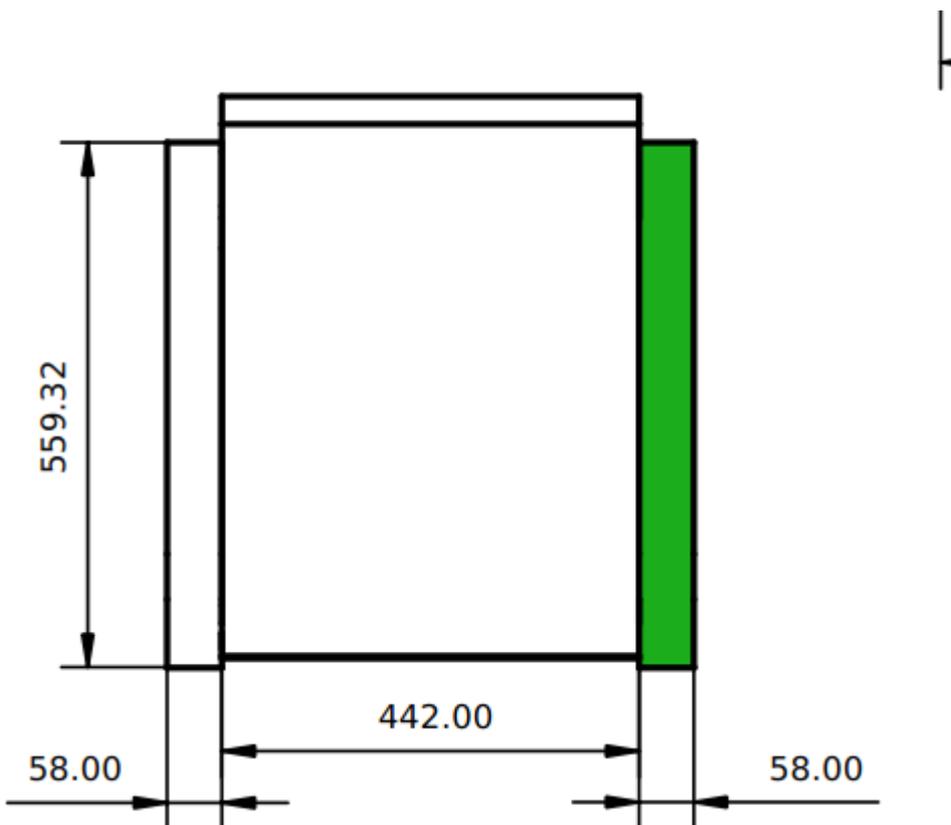
- Appuyez sur le bouton  pour activer les sommets.
- Utilisez Ctrl + Clic gauche de la souris pour sélectionner les deux sommets entre lesquels vous souhaitez mesurer la distance.
- Appuyez sur le bouton  [TechDraw Cote de longueur](#).



- Répétez l'opération jusqu'à ce que toutes les dimensions que vous souhaitez indiquer soient placées. Utilisez les outils  [TechDraw Cote verticale](#) et  [TechDraw Cote horizontale](#) le cas échéant.
- Prenez une minute pour regarder les [propriétés](#) de l'objet Dimension dans la vue combinée.
- Veuillez noter que si vous dimensionnez une vue [axonométrique](#) (par exemple, une vue isométrique) au lieu d'une vue [multiview](#) (par exemple, vue de face) comme nous l'avons fait ici, vous devrez utiliser l'outil  [TechDraw Lier une dimension](#) pour obtenir une dimension précise.



- Nous allons maintenant placer les deux légendes de l'image ci-dessus, en utilisant l'outil  [TechDraw Infobulle](#).



1. En regardant la page dans la [vue 3D](#), sélectionnez la vue à laquelle le ballon sera attaché comme indiqué dans l'image ci-dessus.
2. Appuyez sur le bouton  Infobulle.
3. Le curseur est maintenant affiché sous forme d'icône de l'infobulle. Cliquez sur la page pour placer l'origine de l'infobulle à la position souhaitée.
4. L'infobulle peut être déplacée vers la position souhaitée.

5. Modifiez les propriétés de l'infobulle en double-cliquant sur l'étiquette de l'infobulle ou sur l'objet de l'infobulle dans la [vue en arborescence](#). Cela ouvrira la boîte de dialogue de l'infobulle. Définissez le champ Valeur sur le texte souhaité et modifiez la sélection du menu déroulant Symbole sur **Aucun**.
6. Appuyez sur .
7. Répétez l'opération pour la deuxième légende.

- Nous allons maintenant remplir le bloc de titre de la feuille.
 - Assurez-vous que les cadres de vue, les étiquettes et les sommets sont visibles. Sinon, appuyez sur le bouton  Bascule.
 - Modifiez le texte dans chaque section du bloc de titre de la feuille en cliquant sur le petit carré vert à gauche du texte.

Notre page peut maintenant être exportée au format SVG pour être plus développée dans des applications graphiques comme [Inkscape](#) ou au format DXF. Sélectionnez la page dans la [vue en arborescence](#) puis sélectionnez le menu **Fichier** → **Exporter**. Le format DXF est importable dans presque toutes les applications de CAO 2D existantes. Les pages TechDraw peuvent également être directement imprimées ou exportées au format PDF.

Téléchargements

- Le fichier créé lors de cet exercice: [drawing.FCStd](#)
- La feuille SVG produite à partir de ce fichier: [drawing.svg](#)

Lire plus d'informations

- [Atelier TechDraw](#)
- [TechDraw Comment créer un modèle](#)
- [Tutoriel d'introduction à TechDraw](#)
- [La bibliothèque FreeCAD](#)
- [Inkscape](#)

Regarder les tutoriels

- [la playlist TechDraw de Sliptonic](#)
- [Symboles et vues](#)

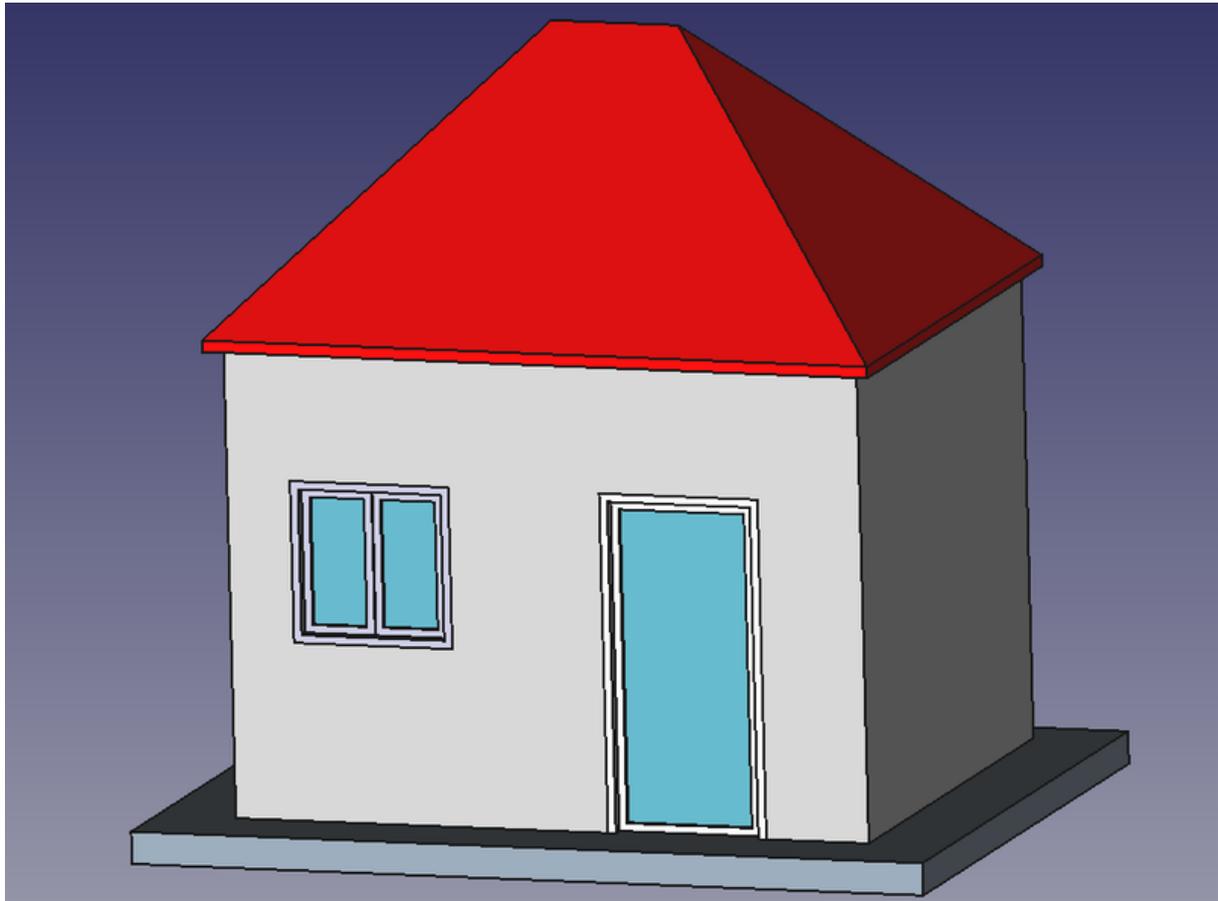
Manuel : Modélisation BIM

BIM signifie [Building Information Modeling](#). C'est un processus utilisé en architecture, ingénierie et construction pour créer et gérer des représentations numériques de structures physiques. Il intègre non seulement la géométrie en 3D, mais aussi des données essentielles telles que les matériaux, les coûts et les calendriers, ce qui permet une analyse avancée et une collaboration tout au long du cycle de vie d'un projet.

Dans FreeCAD, la fonction BIM a évolué de manière significative, en particulier avec la version 1.0, où les ateliers Arch et BIM, auparavant séparés, ont été fusionnés en un atelier BIM intégré. Cette consolidation rationalise les flux de travail, permettant aux utilisateurs de modéliser, de documenter et de gérer les projets de construction plus efficacement dans un environnement unique.

Une avancée majeure introduite dans FreeCAD v1.0 est l'adoption du concept IFC natif. Auparavant, comme la plupart des applications BIM, FreeCAD traduisait les données dans les deux sens entre son modèle de données interne et le format de fichier IFC (Industry Foundation Classes), ce qui entraînait une perte potentielle de données au cours des processus d'ouverture et d'enregistrement. Avec Native IFC, les utilisateurs de FreeCAD peuvent désormais ouvrir, manipuler et enregistrer directement des fichiers IFC, le fichier IFC lui-même servant de structure de données. Cette approche élimine la traduction inutile des données et garantit que les modifications sont enregistrées sans réécrire l'intégralité du fichier, ce qui la rend compatible avec les systèmes de contrôle de version tels que Git et offre un flux de travail plus transparent et plus précis pour la manipulation des fichiers IFC.

Dans ce chapitre, nous verrons comment modéliser ce petit bâtiment :



- Créez un nouveau document et passez à l'[atelier BIM](#).
- Depuis le menu **Edition** → **Préférences** → **Draft** → **Grille et aimantation** et réglez:

- **Lignes principales toutes les**

10

- **Espacement de la grille**

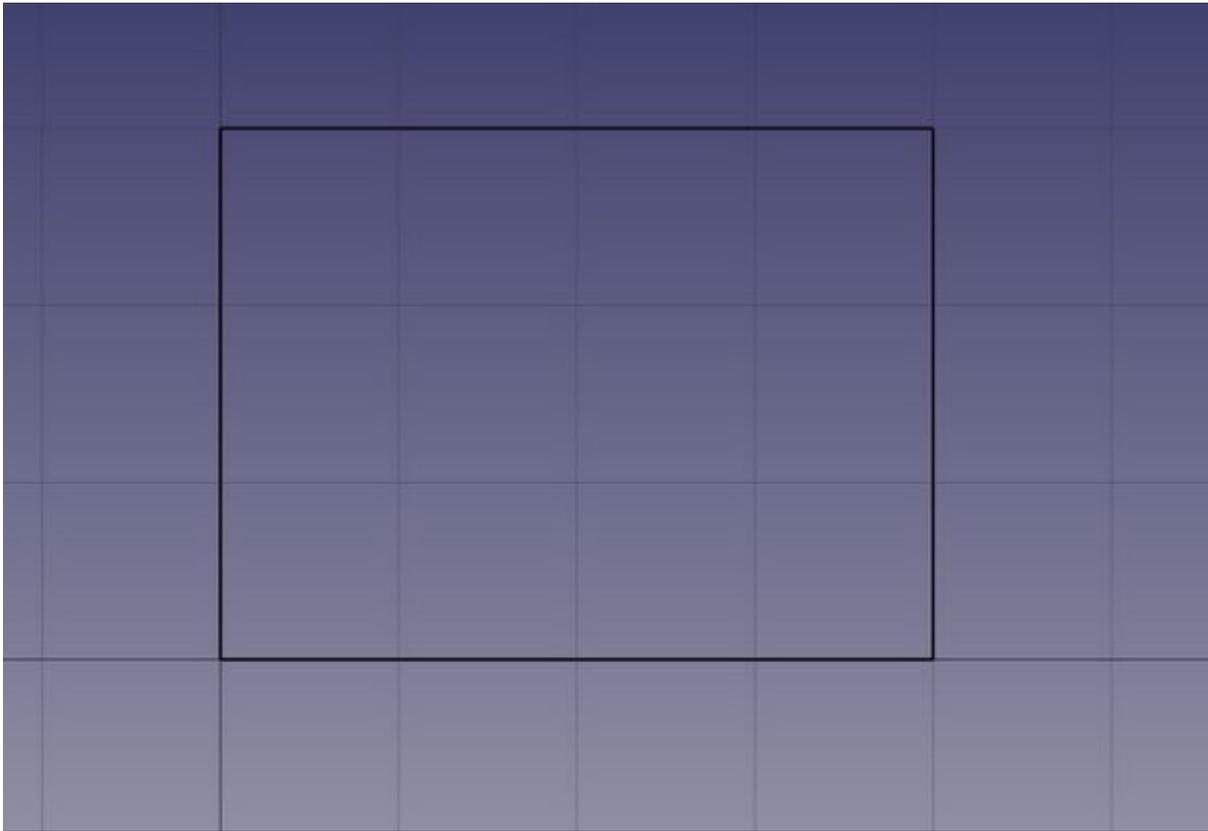
1000mm

pour avoir une grille d'un mètre, ce qui convient à la taille de notre bâtiment.

- **Taille de la grille**

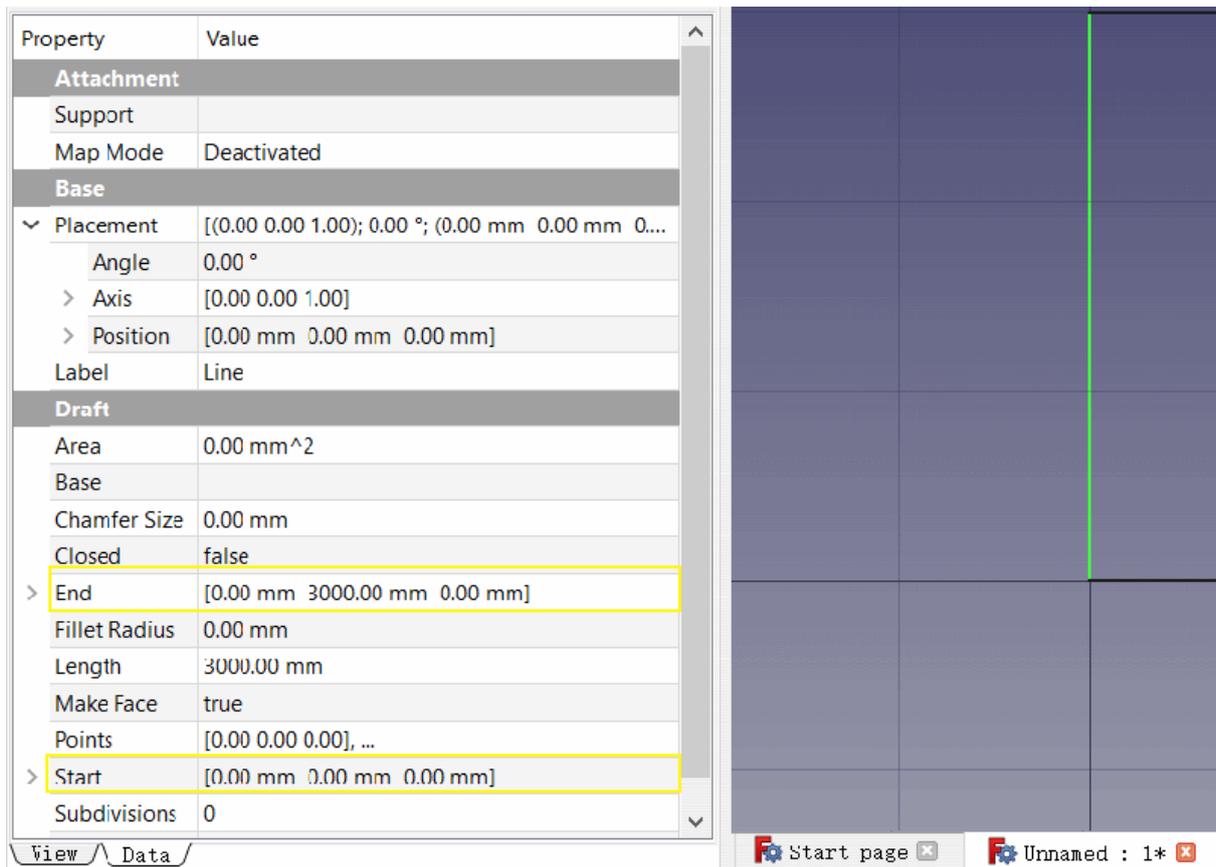
100 lignes

- Sur la barre d'outils **Aimantation**, assurez-vous que le bouton [# Aimantation Grille](#) soit activé, afin que nous puissions utiliser la grille autant que possible.
- Si vous ne voyez pas les axes, cliquez sur le bouton [Basculer la grille](#).
- Réglez le [Plan de travail](#) sur le plan **XY**
- Dessinez quatre lignes avec l'outil [Draft Ligne](#). Vous pouvez entrer les coordonnées manuellement ou simplement sélectionner les points sur la grille avec la souris. Nous utiliserons des mètres pour nos dimensions :
 - Du point (0,0) au point (0,3)
 - Du point (0,3) au point (4,3)
 - Du point (4,3) au point (4,0)
 - Du point (4,0) au point (0,0)



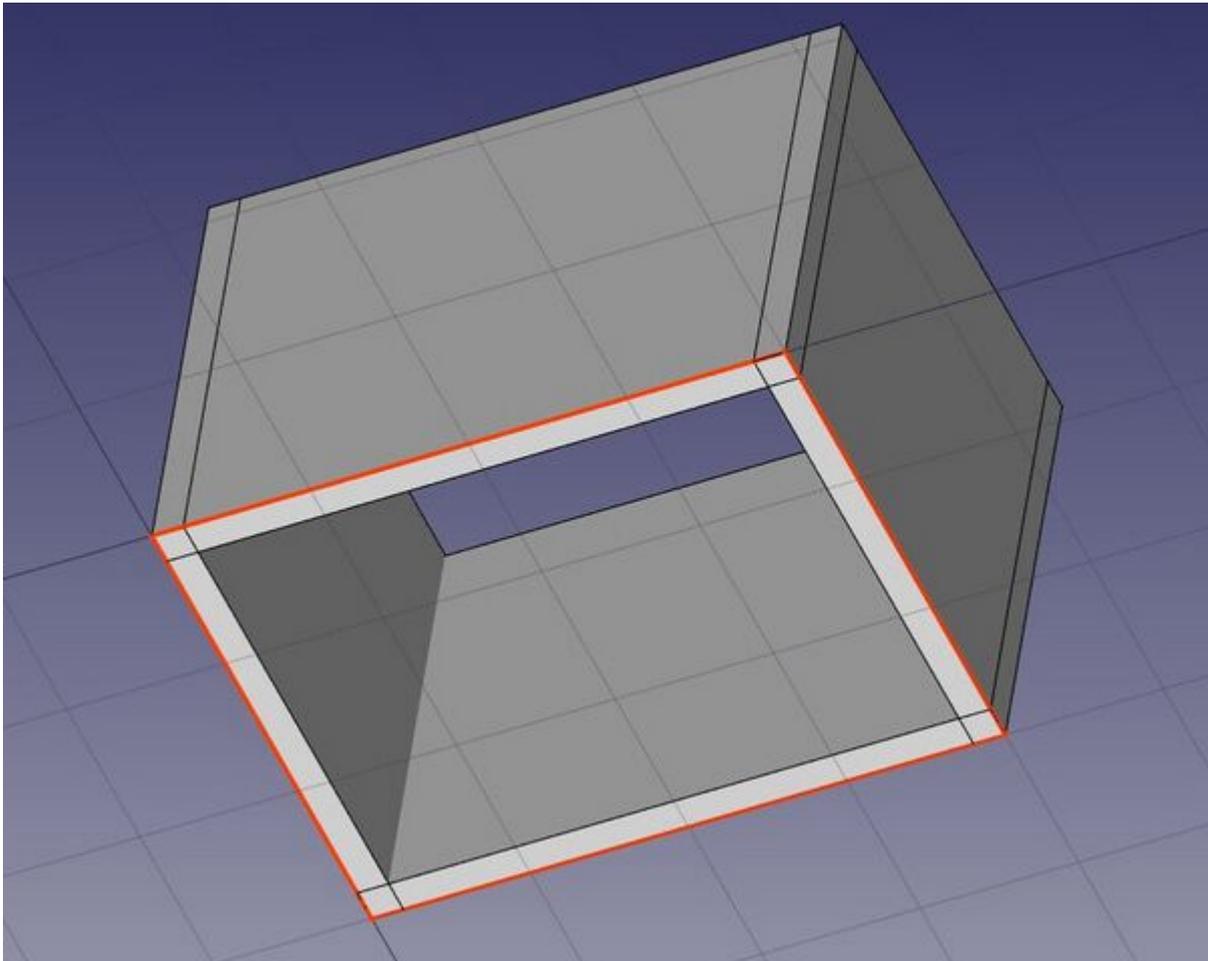
Remarquez que nous avons toujours tracé les lignes dans le même sens (sens des aiguilles d'une montre). Bien que cela ne soit pas obligatoire, cela permet de s'assurer que les murs que nous construirons ensuite seront orientés de la même manière à gauche et à droite. Vous vous demandez peut-être pourquoi nous n'avons pas simplement dessiné un rectangle, ce qui aurait été plus simple. Cependant, l'utilisation de quatre lignes distinctes nous permet de présenter d'autres fonctions de BIM, comme la combinaison de plusieurs objets en un seul, qui est un élément essentiel du flux de travail.

- Une fois que vous avez créé les lignes, vérifiez leurs points de départ et d'arrivée et ajustez-les si nécessaire pour les obtenir exactement.



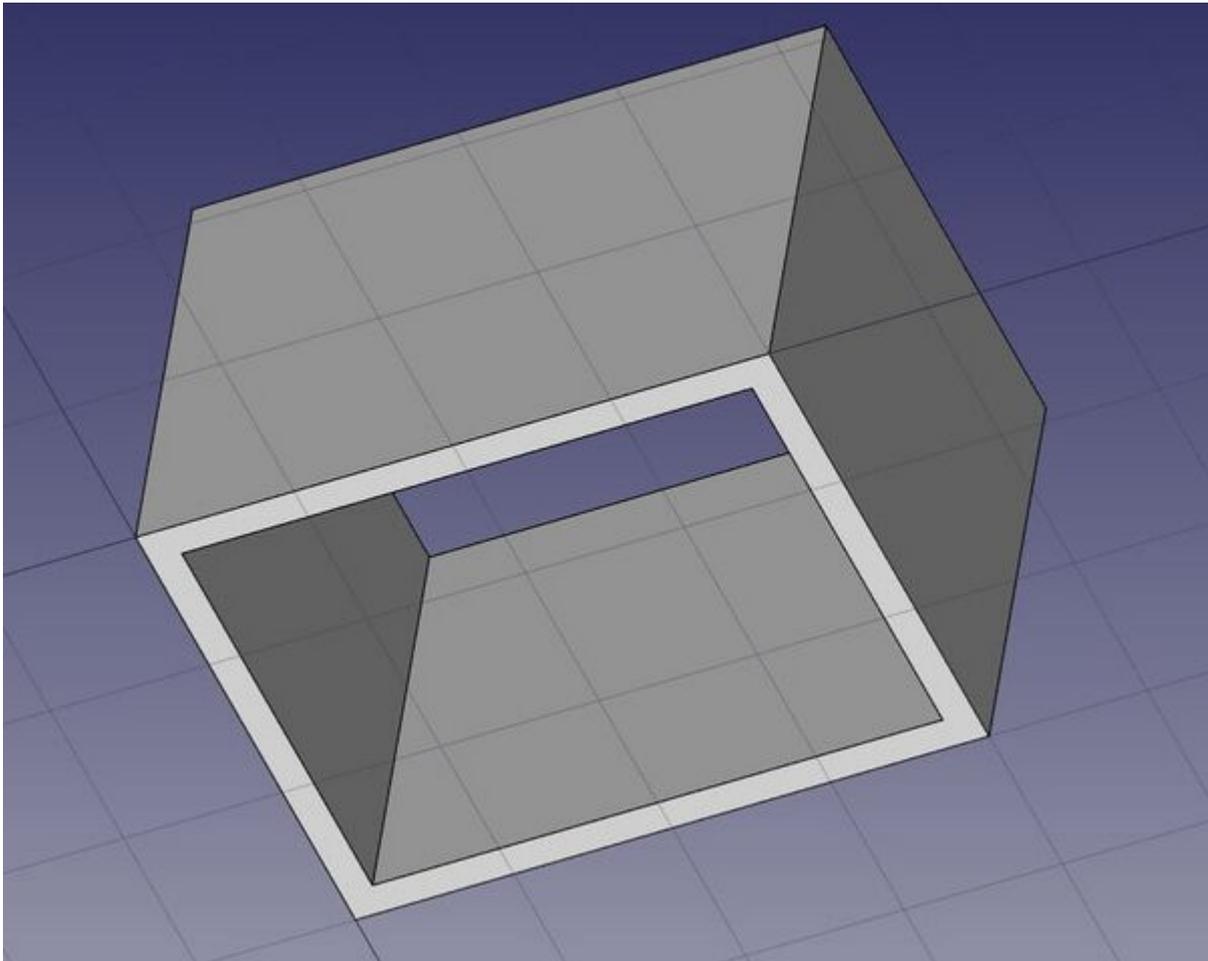
- Sélectionnez les quatre lignes, puis appuyez sur le bouton  [Arch Mur](#).
- Définissez la **Hauteur** des murs à 3m (par défaut).
- Définissez la propriété **Alignement à gauche** En définissant la propriété Alignement à gauche, vous vous assurez que les murs que vous créez seront positionnés à gauche des lignes que vous avez tracées. Dans le BIM Workbench de FreeCAD, les murs sont généralement générés sur la base d'une ligne de référence, et l'alignement gauche ou droit indique de quel côté de la ligne le mur sera placé.

Si vous n'avez pas tracé les lignes dans l'ordre indiqué (dans le sens des aiguilles d'une montre), l'orientation de certains murs peut être inversée, ce qui signifie qu'ils peuvent être placés de l'autre côté de la ligne (à droite au lieu de gauche). Dans ce cas, vous devrez ajuster l'alignement à droite pour ces murs spécifiques afin de vous assurer qu'ils sont tous alignés de manière cohérente. Une fois ce réglage effectué correctement, vous obtiendrez quatre murs qui se croisent aux angles, positionnés à l'intérieur de la ligne de base, formant ainsi la disposition souhaitée.



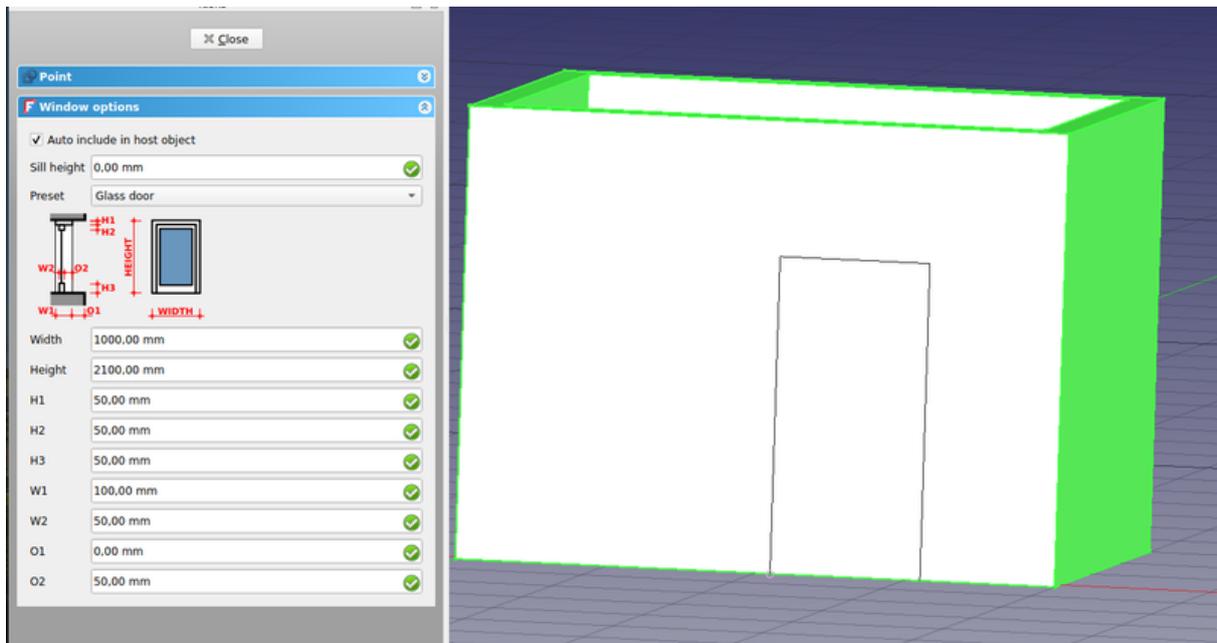
Après avoir créé des murs, l'étape suivante consiste à les joindre pour qu'ils se croisent correctement. Cette opération est nécessaire lorsque les murs ne se connectent pas proprement à leurs intersections. Pour ce faire, vous sélectionnez un mur en tant qu'« hôte » et vous ajoutez les autres murs en tant qu'« ajouts », en fusionnant leur géométrie avec celle de l'hôte. Tous les objets de l'atelier BIM peuvent avoir plusieurs ajouts (qui ajoutent de la géométrie) ou soustractions (qui enlèvent de la géométrie). Ces relations peuvent être gérées à tout moment en double-cliquant sur l'objet dans l'arborescence, ce qui permet de procéder à des ajustements souples pour garantir une intégration harmonieuse des murs et des autres éléments architecturaux.

- Sélectionnez les quatre murs en appuyant sur la touche **Ctrl**, le dernier étant le mur que vous avez choisi de devenir l'hôte.
- Appuyez sur le bouton **+** [Arch Ajouter](#). Les quatre murs ont maintenant été transformés en un seul :

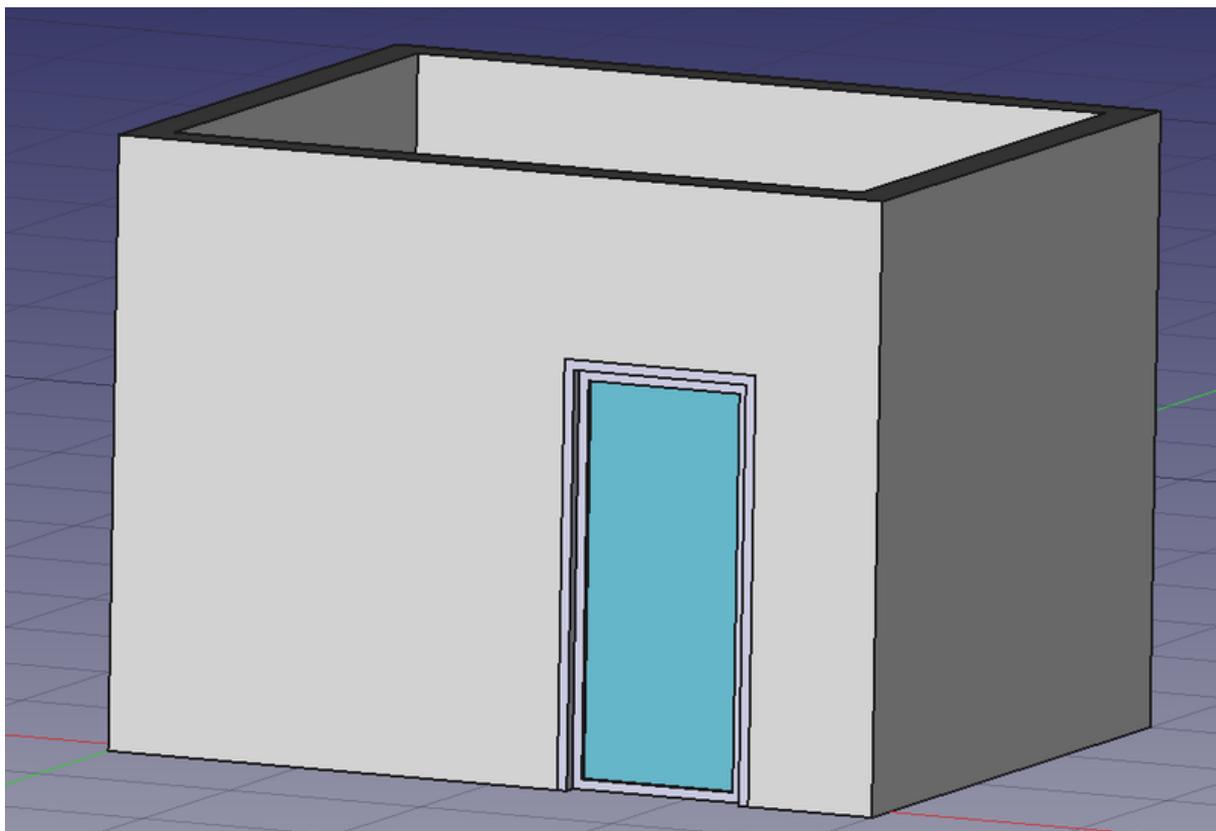


Les parois individuelles sont cependant toujours accessibles, en développant l'objet mur dans l'arborescence (petite flèche).

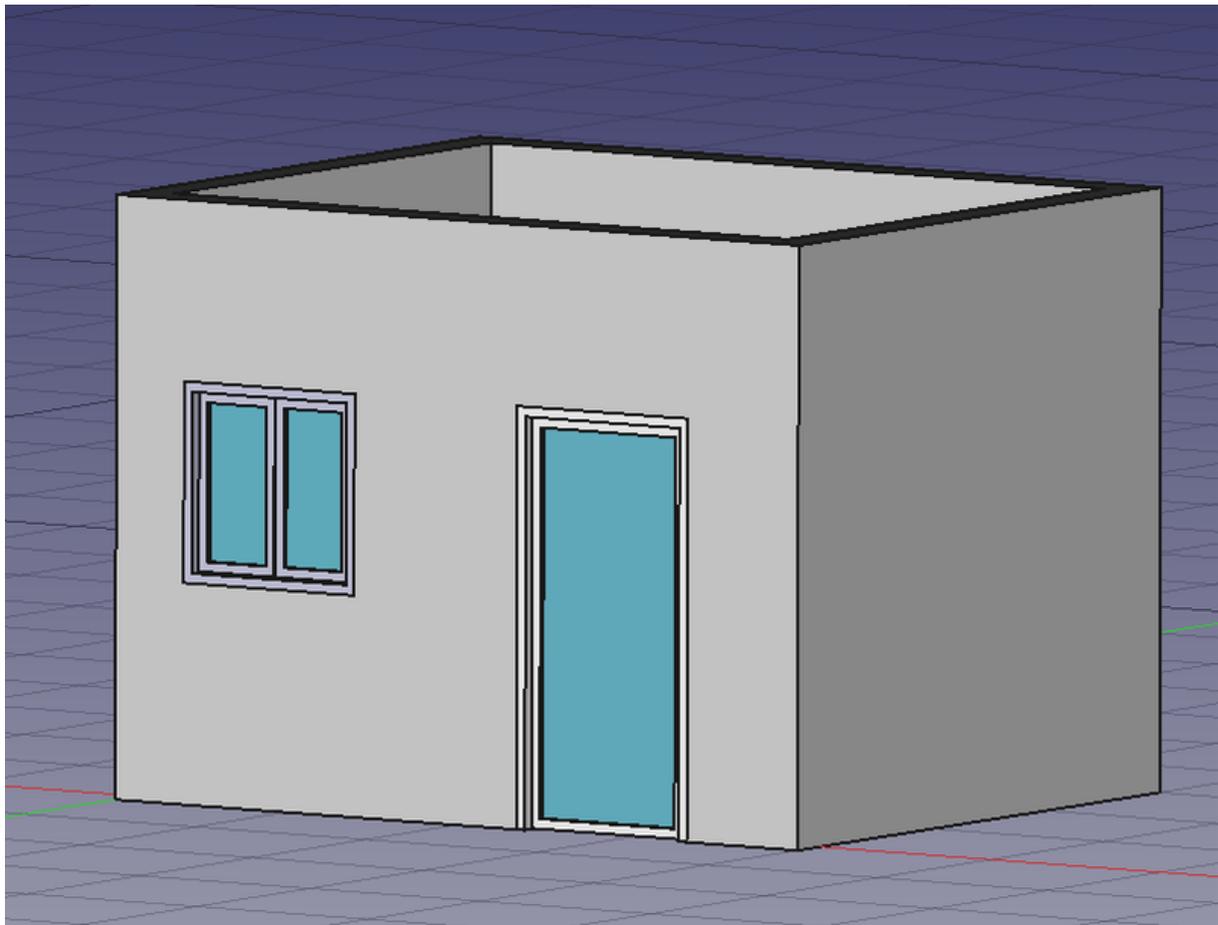
- Plaçons maintenant une porte en appuyant sur l'outil  [Porte](#).
- Commencez par sélectionner le mur. Bien que cette étape ne soit pas obligatoire, c'est une habitude utile à prendre. Si un objet est sélectionné avant de commencer une opération, l'opération s'appliquera automatiquement à cette entité par défaut.
- Définissez le  [plan de travail](#) à **auto** pour ne pas être limité au plan du sol.
- Appuyez sur le bouton  [Porte](#).
- Dans le panneau de création de la porte, sélectionnez le pré-réglage **Porte en verre** et réglez sa **Largeur** à 1 m et sa **Hauteur** à 2,1m. Vous remarquerez que vous pouvez choisir entre différents types de portes et les paramétrer comme vous le souhaitez. Dans FreeCAD, une porte est dérivée par une opération de [fenêtre](#).
- Assurez-vous que l'option  [Aimantation Au plus proche](#) est activée, pour que nous puissions cliquer sur les visages.
- Placez votre porte à peu près au milieu de la face avant du mur :



- Nous pouvons maintenant définir l'emplacement précis en développant le mur et les objets de la fenêtre dans l'arborescence et en changeant la propriété de **Placement** de l'esquisse de base de notre porte. Réglez sa position sur **$x = 0.5$ m, $y = 0$, $z = 0$** . Notre porte est maintenant exactement là où nous le voulons :

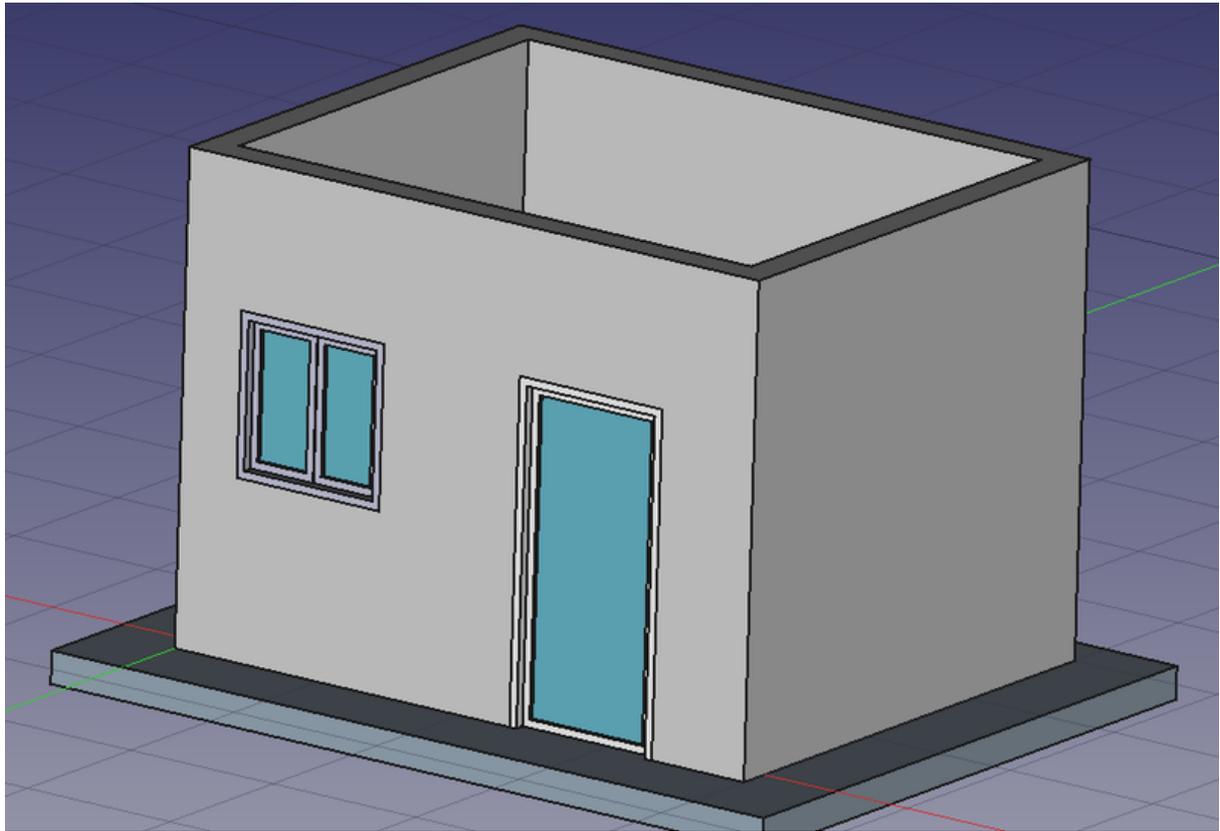


- Plaçons une fenêtre à côté de notre porte. Sélectionnez le mur, appuyez sur l'outil  [Fenêtre](#), sélectionnez le préréglage **Open 2-pane**, et placez une fenêtre **1m x 1m** sur la même face que la porte. Définissez le placement de l'esquisse sous-jacente à la position **$x = 0$, $y = 0$, $z = 1,1$ m**, de sorte que la ligne supérieure de la fenêtre soit alignée sur le haut de la porte.

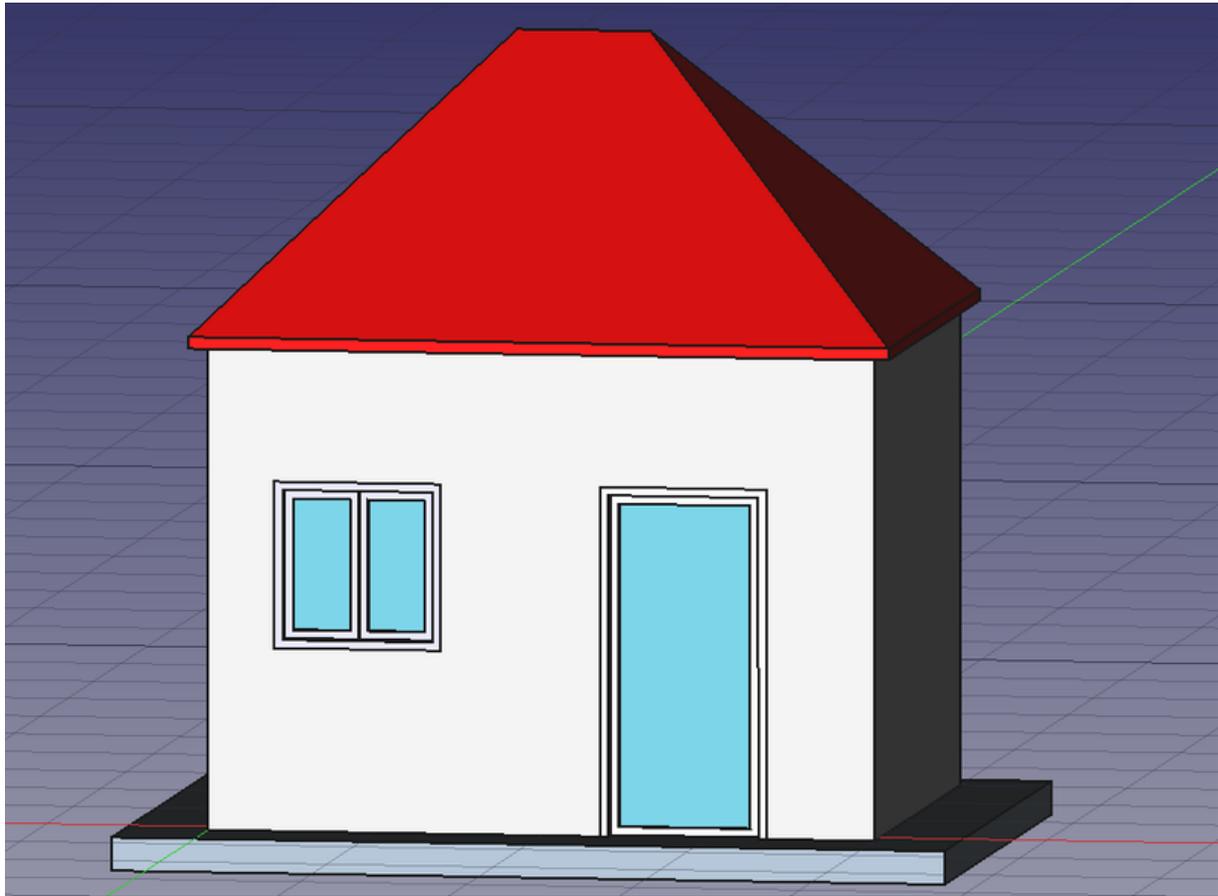


Les fenêtres sont toujours basées sur des esquisses. Vous pouvez facilement créer des fenêtres personnalisées en dessinant d'abord une esquisse sur une face, puis en transformant cette esquisse en fenêtre en la sélectionnant et en cliquant sur le bouton de fenêtre. Vous pouvez ensuite définir les paramètres de la fenêtre (par exemple, les parties de l'esquisse qui doivent être extrudées et leur quantité) en double-cliquant sur la fenêtre dans l'arborescence. Passons maintenant à la création d'une dalle :

- Réglez le [Draft Plan de travail](#) sur le plan **XY**.
- Créez un [Draft Rectangle](#) avec une **longueur** de 5m, une **hauteur** de 4m et placez-le à la position **x: -0.5m, y: -0.7m, z: 0**.
- Sélectionnez le rectangle.
- Cliquez sur l'outil [Dalle](#) pour créer une dalle à partir du rectangle.
- Gardez les valeurs par défaut de 0.2m pour la propriété **height** et définissez la **direction** normale à (0,0,-1), afin que l'extrusion aille vers le bas. Nous aurions pu déplacer l'objet de 0,2 m vers le bas, mais c'est une bonne pratique de garder les objets extrudés alignés avec leur profil de base pour maintenir la cohérence et la précision du modèle.
- Définissez la propriété **Ifc Type** de la dalle sur **slab** (dalle). Ceci n'est pas nécessaire dans FreeCAD, mais il est important pour l'exportation IFC car il garantira que l'objet est exporté avec le type IFC correct.



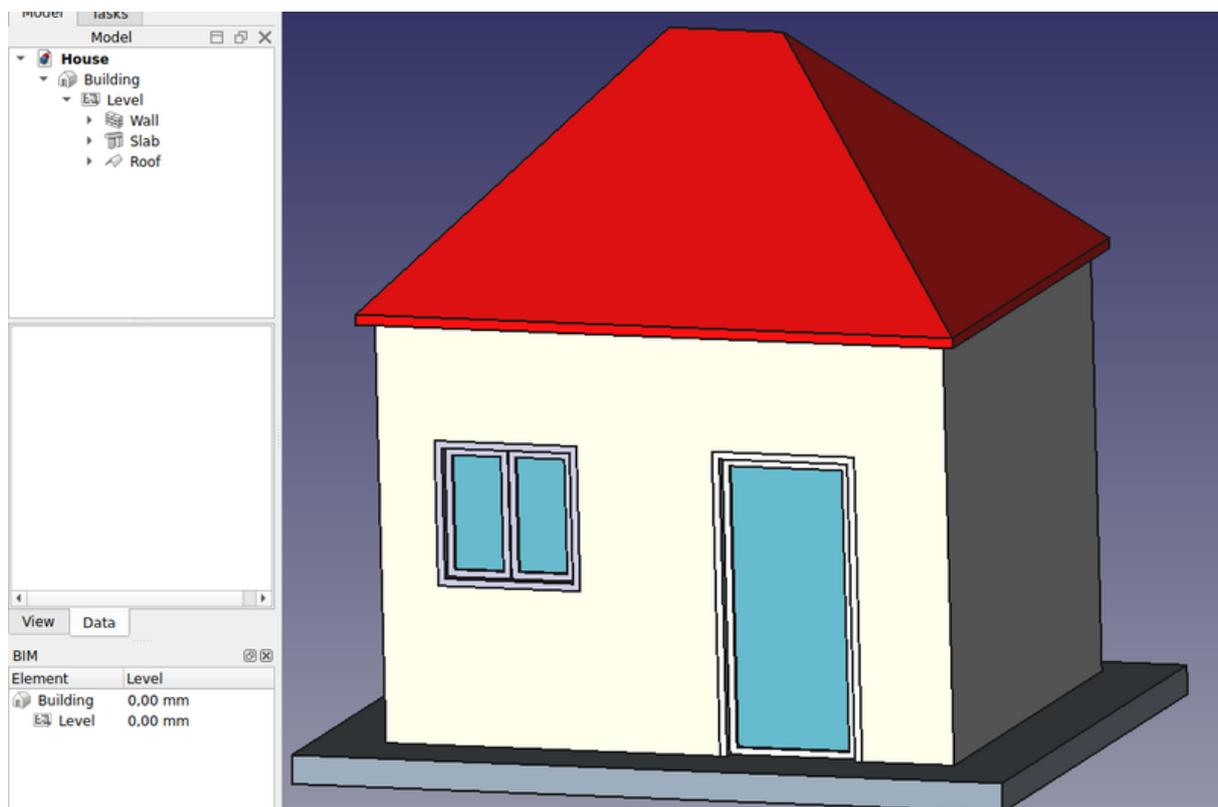
- Mettons un toit au-dessus de nos têtes. Nous pouvons facilement le faire en utilisant l'outil  [Toit](#).
- Appuyez sur l'outil  [Aimantation Plan de travail](#) pour activer l'ébauche sur tous les plans.
- En choisissant l'une des faces supérieures de notre maison, appuyez sur l'option  [Sélectionner un plan](#) pour activer le dessin sur tous les plans. Le plan de travail est maintenant défini sur cette face.
- Créer un  [rectangle](#) s'accrochant à deux points opposés des murs :
- Dans l'onglet **Données** du toit, mettez le paramètre **Runs** à 1600.
- Si vous souhaitez changer la couleur du toit, vous pouvez le faire dans l'onglet *Vue*.



Notre modèle est maintenant complet. L'étape suivante consiste à l'organiser correctement pour s'assurer qu'il s'exporte correctement au format IFC. Les fichiers IFC exigent que tous les éléments de construction soient regroupés au sein d'un objet **bâtiment** et, éventuellement, au sein d'un **étage** spécifique. En outre, tous les bâtiments doivent être situés sur un « site ». Cependant, l'exportateur IFC de FreeCAD génère automatiquement un site par défaut s'il n'y en a pas, de sorte que nous n'avons pas besoin de l'ajouter manuellement. Il est important de structurer correctement le modèle pour respecter les normes IFC, afin d'assurer une collaboration fluide et la compatibilité avec d'autres logiciels BIM. Une bonne organisation permet également d'éviter toute perte de données ou erreur lors du processus d'exportation.

- Sélectionnez les murs, la dalle et le toit.
- Appuyez sur le bouton  [Arch Niveaux](#).
- Sélectionnez le niveau que nous venons de créer.
- Appuyez sur le bouton  [Arch Bâtiment](#).

Notre modèle est maintenant prêt à exporter :

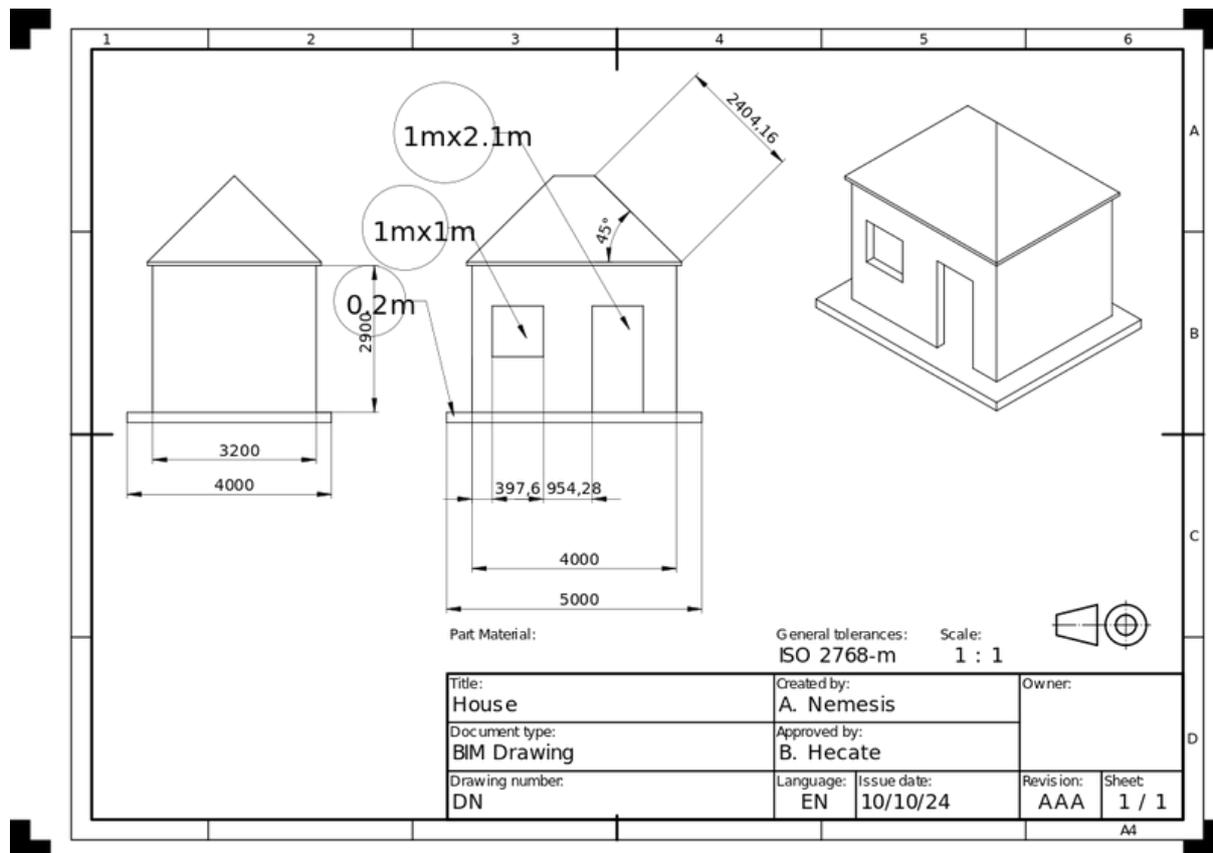


Le [format IFC](#) est l'un des atouts les plus précieux dans un monde BIM gratuit, car il permet l'échange de données entre toute application et acteur du monde de la construction, de manière ouverte (le format est ouvert, gratuit et maintenu par un consortium indépendant). L'exportation de vos modèles BIM comme IFC garantit que tout le monde peut les voir et les analyser, peu importe l'application utilisée.

- Sélectionnez l'objet de niveau supérieur que vous souhaitez exporter, c'est-à-dire l'objet Bâtiment.
- Sélectionnez le menu **Fichier -> Exporter -> Industry Foundation Classes** et enregistrez votre fichier.
- Le fichier IFC résultant peut maintenant être ouvert dans un large éventail d'applications et de visionneuses (l'image ci-dessous montre le fichier ouvert dans la visionneuse gratuite [IfcPlusPlus](#)). Il est important de vérifier le fichier exporté dans une telle application de visualisation avant de le distribuer à d'autres personnes afin de vérifier que toutes les données contenues dans le fichier sont correctes. FreeCAD lui-même peut également être utilisé pour rouvrir le fichier IFC résultant.



Nous pouvons utiliser l' [atelier TechDraw](#) pour créer un dessin de notre bâtiment. Le processus est similaire à ce qui a été montré dans la section précédente, nous n'entrerons donc pas trop dans les détails ici. Il suffit de créer une nouvelle vue en utilisant le . Insérez [une page standard](#), puis sélectionnez la vue que vous souhaitez afficher dans le dessin et ajoutez des cotes au besoin. Cela nous permettra de créer une représentation 2D professionnelle du modèle 3D à des fins de documentation ou de présentation.



Notre page est maintenant prête, et nous pouvons l'exporter vers les formats SVG ou DXF, ou l'imprimer. Le format SVG vous permet d'ouvrir le fichier en utilisant des applications d'illustration telles que [Inkscape](https://inkscape.org/), avec lesquelles vous pouvez rapidement améliorer les dessins techniques et les transformer en dessins de présentation beaucoup plus agréables. Il offre beaucoup plus de possibilités que le format DXF.

Téléchargements

- Le fichier produit lors de cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/house.FCStd>
- Le fichier IFC exporté à partir du fichier ci-dessus: <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/house.ifc>
- Le fichier SVG exporté à partir du fichier ci-dessus: <https://github.com/yorikvanhavre/FreeCADmanual/Blob/master/files/house.svg>

En relation

- [Atelier BIM](#)
- [Atelier Arch](#)
- [Draft Plan de travail](#)
- [Draft Aimantation](#)
- [Expressions](#)
- [Le format IFC](#)
- [IfcOpenShell](#)

- [IfcPlusPlus](#)
- [Inkscape](#)

Manuel : Utilisation de feuilles de calcul

L' [atelier Spreadsheet](#) dans FreeCAD permet aux utilisateurs de créer et de gérer des [feuilles de calcul](#), telles que celles faites avec [Excel](#) ou [Calc de LibreOffice](#), directement dans leurs projets de conception. Il permet de saisir, d'organiser et de manipuler des données sous forme de tableau, qui peuvent ensuite être liées à divers paramètres et modèles dans le projet.

L'un de ses principaux avantages est son utilisation dans la modélisation paramétrique. Les feuilles de calcul peuvent être reliées aux dimensions et aux propriétés des modèles 3D, ce qui en fait un outil essentiel pour les modifications dynamiques de la conception. Par exemple, l'ajustement d'une valeur dans la feuille de calcul mettra automatiquement à jour la dimension correspondante dans le modèle.

Outre la gestion des valeurs, l'atelier est excellent pour la gestion des données, en stockant des informations essentielles telles que les propriétés des matériaux, les dimensions et les paramètres de l'ensemble du projet. Cela s'avère particulièrement utile dans les projets complexes où plusieurs valeurs doivent être référencées ou ajustées.

Les feuilles de calcul permettent également aux utilisateurs de saisir des formules pour les calculs et la gestion des données. Ces formules peuvent faire référence à d'autres cellules du tableur ou à des paramètres du modèle 3D, ce qui rend l'ensemble du processus de conception adaptable et réactif aux changements.

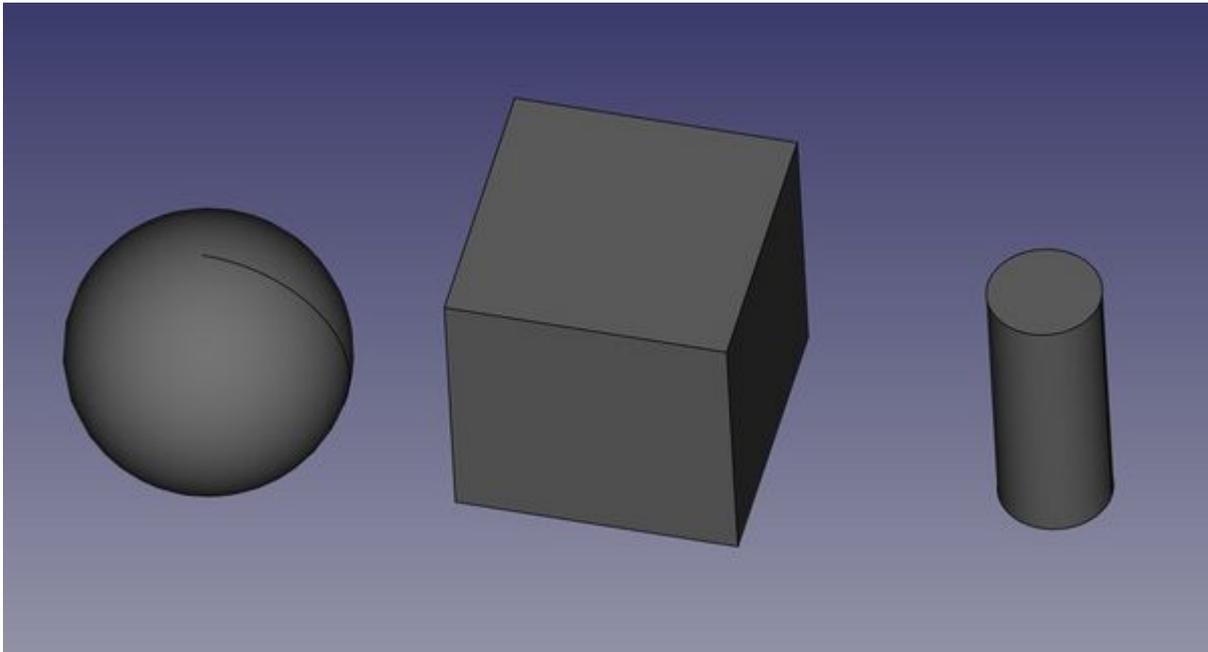
Il peut s'intégrer de manière transparente à d'autres ateliers FreeCAD, ce qui permet une interaction entre les données et les composants du modèle. Cette intégration centralise le contrôle des différents aspects du projet, ce qui en facilite la gestion. L'interface est simple, ressemblant à un tableur traditionnel, ce qui la rend familière et facile à utiliser pour ceux qui sont déjà habitués à des programmes comme Excel ou LibreOffice Calc.

En pratique, l'atelier Spreadsheet est polyvalent pour différents cas d'utilisation, notamment la définition de paramètres à l'échelle du projet, la gestion des nomenclatures et l'exécution de calculs personnalisés qui influencent les décisions de conception. Il simplifie les projets complexes en centralisant le contrôle des paramètres en un seul endroit.

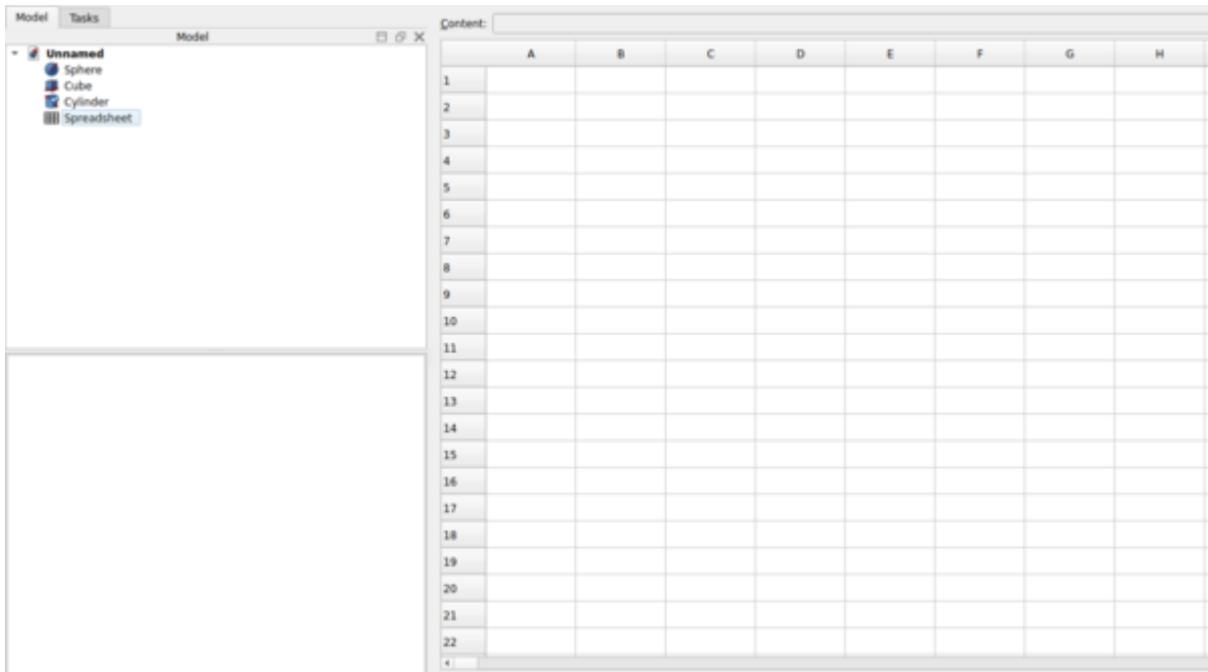
Dans l'exemple suivant, nous allons créer quelques objets, récupérer certaines de leurs propriétés dans une feuille de calcul, puis utiliser la feuille de calcul pour piloter directement les propriétés d'autres objets.

Lire des propriétés

- Commencez par passer à l' [atelier Part](#) et créez quelques objets : un  [cube](#), un  [cylindre](#) et une  [sphère](#).
- Modifiez leur propriété **Placement** (ou utilisez l'outil  [Draft Déplacer](#)) pour les séparer un peu, afin que nous puissions mieux observer les effets de ce que nous allons faire :



- Maintenant, on va extraire des informations sur ces objets. Passez à l'[atelier Spreadsheet](#).
- Appuyez sur le bouton **Créer une feuille de calcul**.
- Double-cliquez sur l'objet Spreadsheet dans la vue en arborescence. L'éditeur de feuille de calcul s'ouvre :



Bien que l'éditeur de feuilles de calcul de FreeCAD ne soit pas aussi riche en fonctionnalités que des applications dédiées comme Excel ou LibreOffice Calc, il fournit des outils essentiels pour la plupart des tâches de conception. Les utilisateurs peuvent ajuster les propriétés des cellules telles que la taille, la couleur et l'alignement, et fusionner ou diviser les cellules pour une meilleure organisation. Les formules de base ou les références à d'autres cellules sont prises en charge, ce qui permet une manipulation simple des données. Ce qui le distingue, c'est son intégration profonde avec l'environnement de modélisation de FreeCAD, où les modifications apportées à la feuille de calcul peuvent automatiquement mettre à jour

les dimensions du modèle en temps réel. Bien qu'il ne dispose pas de fonctions avancées telles que les tableaux croisés dynamiques ou les graphiques, son orientation vers la conception paramétrique en fait un outil puissant pour la gestion des données de conception directement dans FreeCAD.

Dans FreeCAD, au-delà des fonctionnalités standard des tableurs, il existe une fonction particulièrement utile : la possibilité de référencer non seulement d'autres cellules, mais aussi des objets au sein du document et d'extraire des valeurs de leurs propriétés. Par exemple, vous pouvez extraire les propriétés des objets 3D qui sont visibles dans l'onglet **Données de l'Éditeur de propriétés** lorsqu'un objet est sélectionné. Cela permet une intégration transparente entre la feuille de calcul et le modèle 3D, facilitant la liaison et l'automatisation des modifications basées sur les paramètres des objets de la conception, offrant ainsi un flux de travail plus dynamique et interconnecté.

- Commençons par saisir quelques textes dans les cellules A1, A2 et A3, afin de nous souvenir de ce qui est quoi par la suite, par exemple **Cube Length**, **Cylinder Radius** et **Sphere Radius**. Pour saisir du texte, il suffit d'écrire dans le champ « Contenu » au-dessus de la feuille de calcul, ou de double-cliquer sur une cellule.
- Récupérons maintenant la longueur réelle de notre cube. Dans la cellule B1, tapez **=Cube.Length**. Vous remarquerez que le tableur dispose d'un mécanisme d'auto-complétion, qui est en fait le même que l'éditeur d'expressions que nous avons utilisé dans le chapitre précédent.
- Faites de même pour les cellules B2 (**=Cylinder.Radius**) et B3 (**=Sphere.Radius**).

	A	B	C	D
1	Cube Length	10,00 mm		
2	Cylinder Radius	2,00 mm		
3	Sphere Radius	=Sph		
4		Sphere <<Sphere>>		
5				
6				
7				
8				
9				
10				
11				

- Bien que ces résultats soient exprimés avec leurs unités, les valeurs peuvent être manipulées comme n'importe quel nombre, essayez, par exemple, d'entrer dans la cellule C1 : **=B1*2**.
- Nous pouvons maintenant modifier une de ces valeurs dans l'éditeur de propriétés, et la modification sera immédiatement reflétée dans la feuille de calcul. Par exemple, modifions la longueur de notre cube à **20mm** :

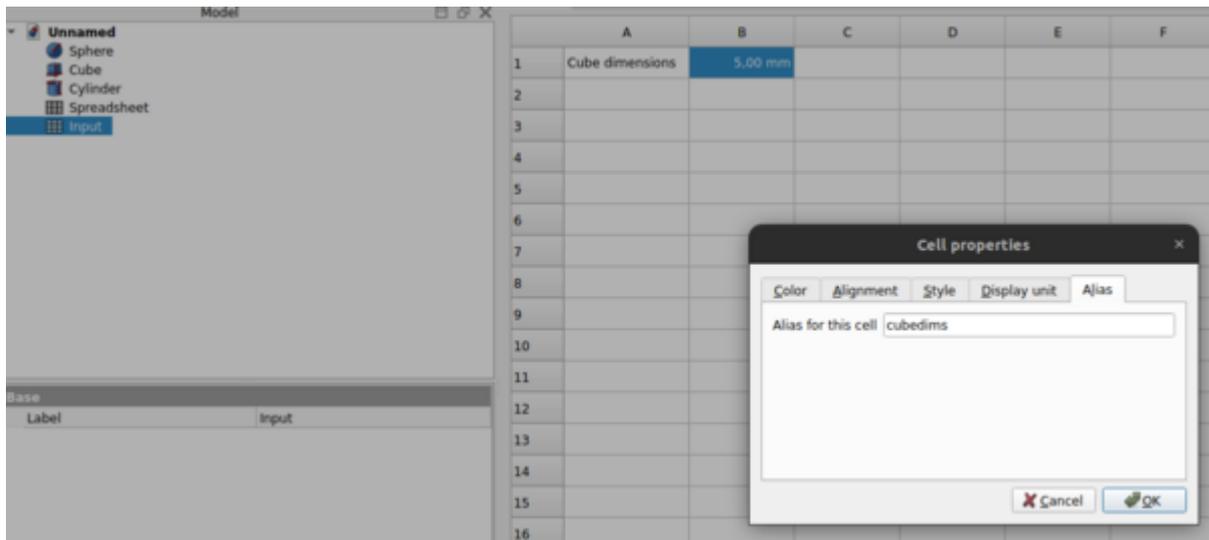
	A	B	C	D
1	Cube Length	10,00 mm	20,00 mm	
2	Cylinder Radius	2,00 mm		
3	Sphere Radius	5,00 mm		
4				
5				
6				

La page de l'[atelier Spreadsheet](#) décrit plus en détail toutes les opérations et fonctions disponibles que vous pouvez utiliser dans les feuilles de calcul.

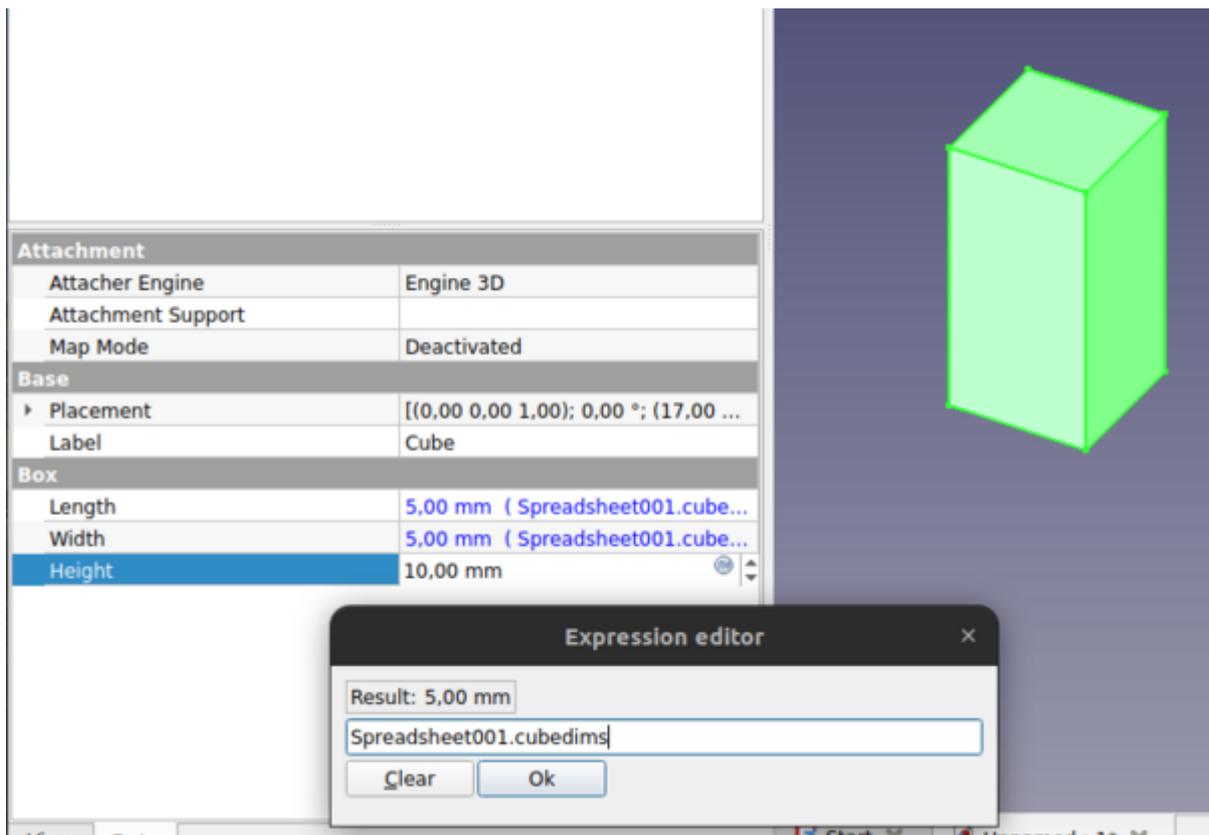
Écrire des propriétés

Une autre fonctionnalité puissante de l'atelier Spreadsheet de FreeCAD est la possibilité non seulement de lire les valeurs des propriétés des objets 3D, mais aussi de leur attribuer des valeurs. Cela permet de contrôler les dimensions et les attributs des objets directement à partir de la feuille de calcul. Cependant, l'une des règles fondamentales de FreeCAD est que les dépendances circulaires sont interdites, ce qui signifie qu'une feuille de calcul ne peut pas à la fois lire et écrire sur le même objet. Cela créerait une situation où l'objet dépendrait de la feuille de calcul tandis que la feuille de calcul dépendrait également de l'objet, ce qui conduirait à une configuration invalide. Pour éviter cela, une deuxième feuille de calcul est généralement créée pour gérer l'écriture des valeurs, ce qui garantit une séparation claire entre les processus de lecture et d'écriture.

- Nous pouvons maintenant fermer l'onglet du tableur (sous la vue 3D). Ce n'est pas obligatoire, il n'y a aucun problème à garder plusieurs fenêtres de feuilles de calcul ouvertes.
- Appuyez à nouveau sur le bouton **Nouvelle feuille de calcul**.
- Changez le nom de la nouvelle feuille de calcul en quelque chose de plus significatif, comme **Input** (faites-le en cliquant avec le bouton droit de la souris sur le nouvel objet feuille de calcul et en choisissant **Renommer**).
- Double-cliquez sur la feuille de calcul Input pour ouvrir l'éditeur de feuille de calcul.
- Dans la cellule A1, mettons un texte descriptif, par exemple : "Cube dimensions"
- Dans la cellule B1, écrivons **=5mm** (l'utilisation du signe = permet de s'assurer que la valeur est interprétée comme une valeur unitaire, et non comme un texte).
- Pour pouvoir utiliser cette valeur en dehors de la feuille de calcul, nous devons donner un nom, ou un alias, à la cellule B1. Cliquez avec le bouton droit de la souris sur la cellule, cliquez sur **Propriétés** et sélectionnez l'onglet **Alias**. Donnez-lui un nom, par exemple **cubedims** :

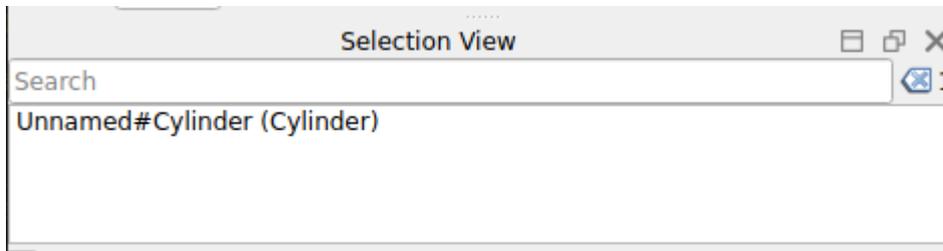


- Appuyez sur **OK**, puis fermer l'onglet de la feuille de calcul
- Sélectionnez l'objet cube.
- Dans l'éditeur de propriétés, cliquez sur l'icône  **expression** située sur le côté droit du champ **Length**. Cela ouvrira l'éditeur d'expressions, où vous pouvez écrire **Spreadsheet001.cubedims**. Répétez ceci pour **Height** et **Width** :



La raison pour laquelle nous utilisons « Spreadsheet001 » au lieu de « Input » dans l'expression est que chaque objet dans un document FreeCAD a un nom interne unique et une étiquette plus conviviale. Alors que l'étiquette est ce qui apparaît dans l'arborescence, le nom interne est utilisé pour identifier de manière unique les objets dans le document. FreeCAD vous permet d'attribuer la même étiquette à plusieurs objets si vous réglez vos préférences, mais le nom interne reste unique. Pour toute

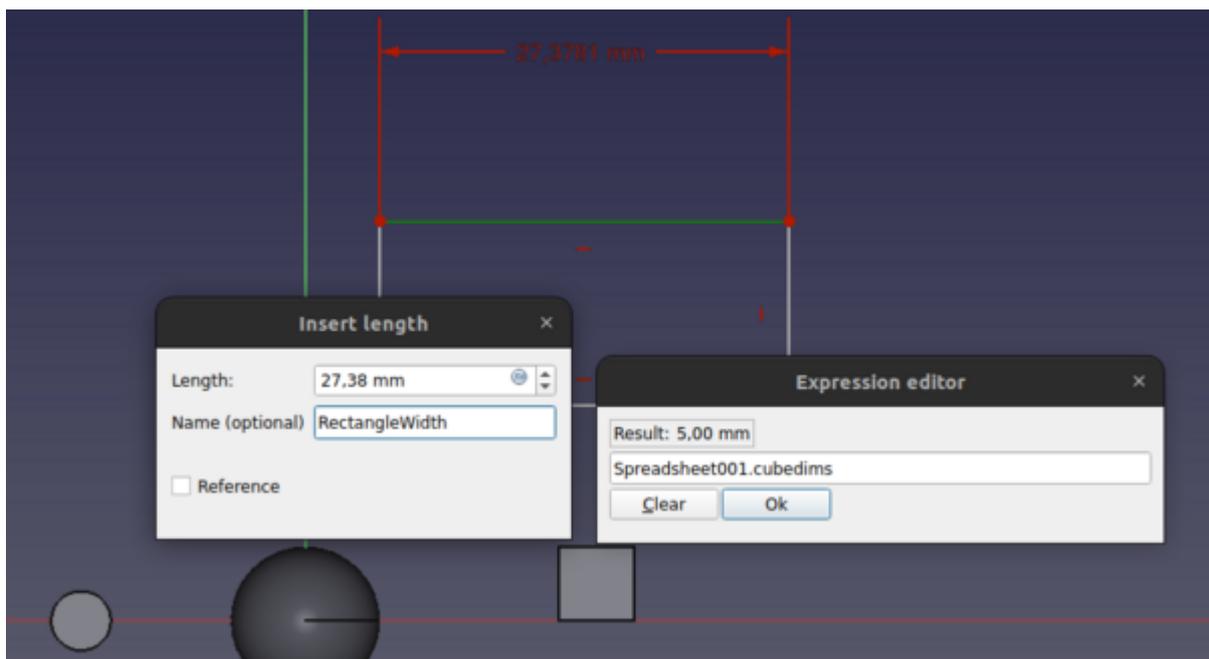
opération nécessitant l'identification d'un objet sans ambiguïté, FreeCAD utilise le nom interne plutôt que l'étiquette, car l'étiquette peut se référer à plusieurs objets. Pour trouver le nom interne d'un objet, il est utile de garder le panneau de sélection (accessible via Affichage → Panneaux) ouvert. Ce panneau affichera toujours le nom interne de l'objet sélectionné, ce qui vous permettra d'utiliser la bonne référence dans vos expressions.



En utilisant des alias de cellules dans l'atelier Spreadsheet de FreeCAD, il est possible de stocker des « valeurs de référence » dans le document, ce qui facilite la gestion et la modification des paramètres clés. Par exemple, une feuille de calcul peut contenir les dimensions d'un modèle, ce qui permet de référencer ces valeurs tout au long de la conception. Cette approche simplifie le processus de mise à jour du modèle. Si de nouvelles dimensions sont nécessaires, il suffit d'ouvrir la feuille de calcul, d'ajuster les valeurs et le modèle sera automatiquement mis à jour pour refléter ces changements. Cette méthode rationalise la gestion des versions et améliore l'efficacité, en particulier dans la modélisation paramétrique, où les dimensions changent fréquemment en fonction des exigences du projet.

Enfin, notez que les contraintes dans une esquisse peuvent également recevoir la valeur d'une cellule de tableur :

Vous pouvez également donner des alias aux contraintes dimensionnelles (horizontales, verticales ou distances) dans une esquisse (vous pouvez ensuite utiliser ces valeurs à l'extérieur de l'esquisse) :



Télécharger

- Le fichier produit dans cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/blob/master/files/spreadsheet.FCStd>

Lire plus d'informations

- [Atelier Spreadsheet](#)
- [Expressions](#)

Manuel : Création d'analyses FEM

[Méthode des éléments finis](#) (FEM en anglais) est une technique de calcul puissante utilisée pour résoudre des problèmes complexes en ingénierie, en physique et en mathématiques appliquées. Elle consiste à décomposer un objet ou une structure vaste et complexe en parties plus petites et plus simples, appelées éléments finis. Ces éléments sont analysés individuellement et leur comportement est combiné pour prédire comment l'ensemble de la structure réagira aux influences extérieures, telles que les forces, la chaleur ou les vibrations.

FEM est largement utilisée dans des domaines tels que l'ingénierie structurelle, la conception mécanique, l'aérodynamique et l'électromagnétisme pour simuler la façon dont les objets se déforment sous la contrainte, comment la chaleur circule à travers les matériaux et comment les champs électromagnétiques interagissent avec différents objets. En fournissant des informations détaillées sur ces interactions, FEM permet aux ingénieurs et aux concepteurs d'optimiser leurs produits en termes de performance, de sécurité et d'efficacité sans avoir recours à des prototypes physiques.

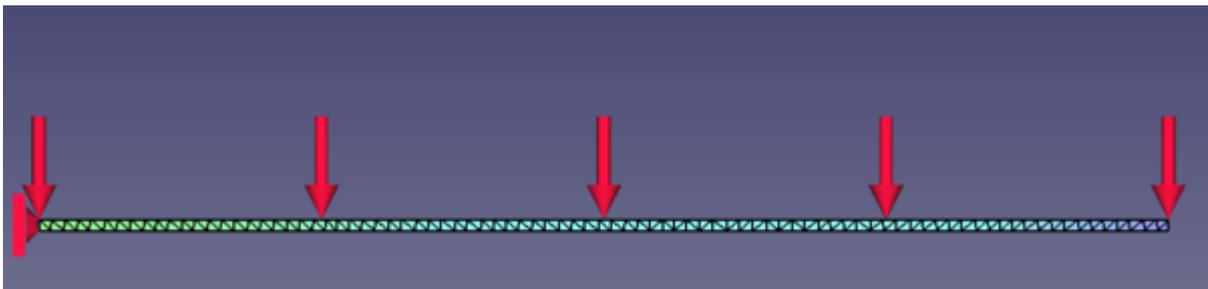
L'obtention de telles simulations dans FreeCAD se fait à l'aide de l'atelier  [FEM](#), qui est spécifiquement conçu pour effectuer des analyses par éléments finis (FEA). Il fournit un ensemble complet d'outils pour la préparation du modèle, l'attribution des propriétés des matériaux, le maillage et l'exécution des simulations. L'atelier FEM est polyvalent et prend en charge un large éventail de simulations telles que les analyses structurelles, thermiques et dynamiques, avec des solveurs tels que [CalculiX](#) et d'autres disponibles.

Cet atelier permet l'intégration d'autres ateliers de FreeCAD, ce qui permet une préparation et une analyse transparentes des modèles. Il fournit également de puissants outils de post-traitement pour visualiser et interpréter les résultats de la simulation, tels que les contraintes, les déformations et les distributions thermiques. Le flux de travail suit les étapes suivantes :

- **Préparation de la géométrie** : le modèle doit être simplifié ou optimisé pour l'analyse FEM. Il s'agit souvent de supprimer les détails ou les fonctions inutiles qui ne contribuent pas à la simulation, mais qui pourraient la rendre coûteuse en termes de calcul. Vous pouvez utiliser des outils d'autres ateliers, comme  [PartDesign](#) ou  [Part](#), pour préparer votre géométrie 3D. La page [FEM Préparation de la géométrie et maillage](#) décrit comment préparer correctement la géométrie pour son utilisation dans l'atelier FEM.
- **Attribution des propriétés aux matériaux** : les définitions des matériaux sont essentielles pour des simulations précises. Des propriétés telles que le module d'Young, le coefficient de Poisson et la densité sont attribuées pour les simulations structurelles, ou la conductivité thermique et la capacité thermique spécifique pour l'analyse thermique. Les matériaux peuvent être sélectionnés dans la bibliothèque de matériaux de FreeCAD ou personnalisés selon les besoins.
- **Maillage** : le maillage divise la géométrie en éléments finis, ce qui permet au solveur d'analyser l'objet. La qualité du maillage est cruciale, car des mailles plus fines permettent d'obtenir des simulations plus précises, mais nécessitent une plus grande puissance de calcul. Des outils sont disponibles pour affiner le

maillage localement, en se concentrant sur les zones où les contraintes ou les déformations devraient être plus importantes. Consultez les [bases du maillages](#) pour plus d'informations.

- **Application de charges et de contraintes** : dans cette étape, des conditions physiques telles que des forces, des pressions, des moments ou des charges thermiques sont appliquées au modèle. Des conditions limites sont également définies, telles que des points de fixation, des contraintes de symétrie ou des restrictions de mouvement, en fonction du scénario simulé.
- **Exécution du solveur** : une fois la configuration terminée, le solveur calcule la réponse du modèle aux conditions appliquées. Les solveurs comme CalculiX calculent les déplacements, les contraintes et d'autres quantités, en fonction du type d'analyse effectué. Le processus peut prendre plus ou moins de temps en fonction de la densité du maillage et de la complexité du modèle.
- **Post-traitement** : après la simulation, les résultats sont visualisés à l'aide d'outils dans l'atelier FEM. Les champs de contrainte, de déformation et de déplacement sont représentés sous forme de cartes de couleur et des tracés de déformation peuvent être générés. Ces visualisations permettent une analyse approfondie des performances du modèle, en mettant en évidence les zones de forte contrainte ou de déformation.



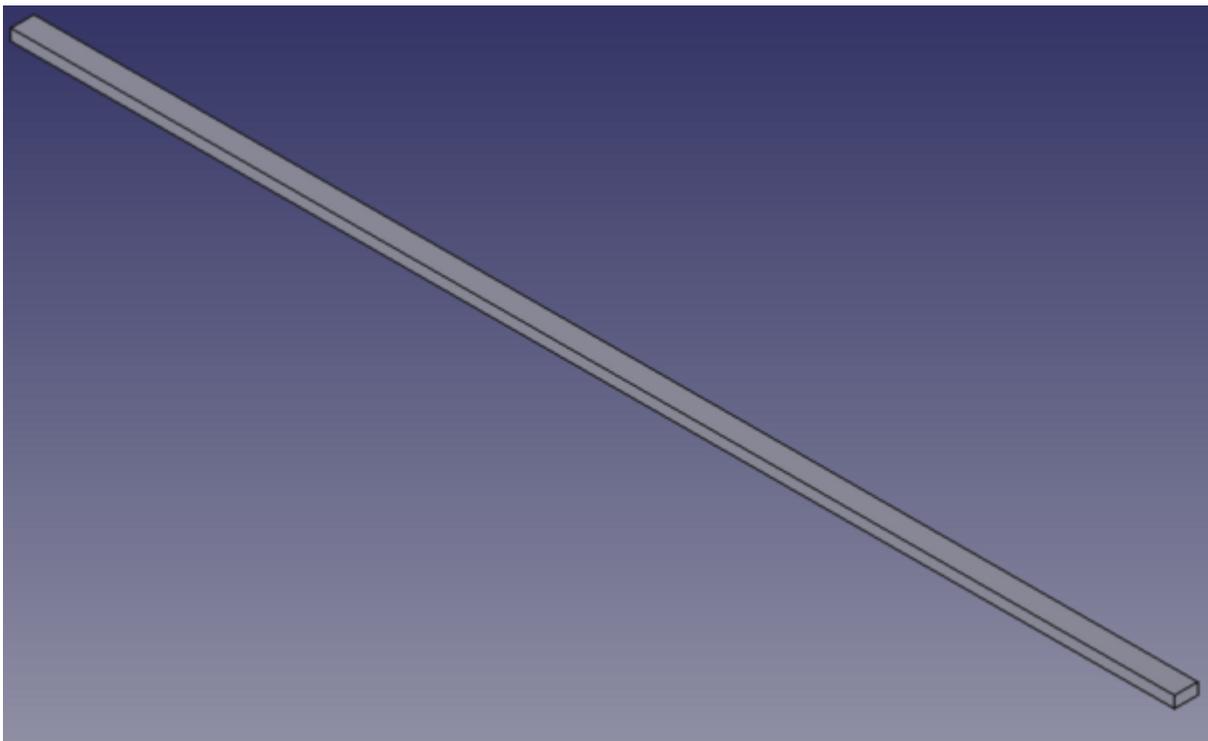
Préparation de FreeCAD

Dans cette section, nous démontrerons la procédure générale d'analyse FEM à l'aide d'un exemple simple. Bien que le sujet des éléments finis soit vaste, nous nous concentrerons sur une géométrie simple : une poutre en porte-à-faux. Notre objectif est de déterminer le déplacement vertical maximal de cette poutre sous l'effet d'une charge appliquée, et nous comparerons les résultats numériques à la solution analytique. En mécanique numérique, la vérification des résultats numériques par rapport aux données expérimentales ou aux solutions analytiques est essentielle pour garantir la précision et la fiabilité de la simulation. En outre, nous utiliserons des packages qui sont déjà inclus dans l'installation de FreeCAD, de sorte qu'aucune installation supplémentaire ne sera nécessaire pour cette analyse.

Préparation de la géométrie

Tout d'abord, nous allons créer notre géométrie simple. Pour cela, nous utiliserons l'atelier [PartDesign](#).

- Créez un nouveau document et accédez à l'atelier [PartDesign](#).
- Appuyez sur [Nouvelle esquisse](#) pour créer une nouvelle esquisse sur le plan YZ.
- Créez un [rectangle centré](#) autour du point d'origine.
 - En utilisant la commande [Contrainte de dimension](#), définissez la dimension verticale à **20 mm** et la dimension horizontale à **10 mm**
- Quittez le mode esquisse.
- En ayant sélectionné notre esquisse nouvellement créée appliquez une [protrusion](#) avec une longueur de **1000 mm**.
- Notre géométrie est maintenant prête. Dans cet exemple, nous avons réduit la hauteur (h) et la largeur (b) de la poutre par rapport à sa longueur (L) afin de nous concentrer sur la façon dont elle se plie. Ce faisant, nous pouvons nous assurer que la poutre se comporte comme un objet long et fin typique où la flexion est l'effet principal lorsqu'une force est appliquée. Cette configuration facilite également la comparaison de nos résultats avec des formules simples que nous pouvons calculer à la main.

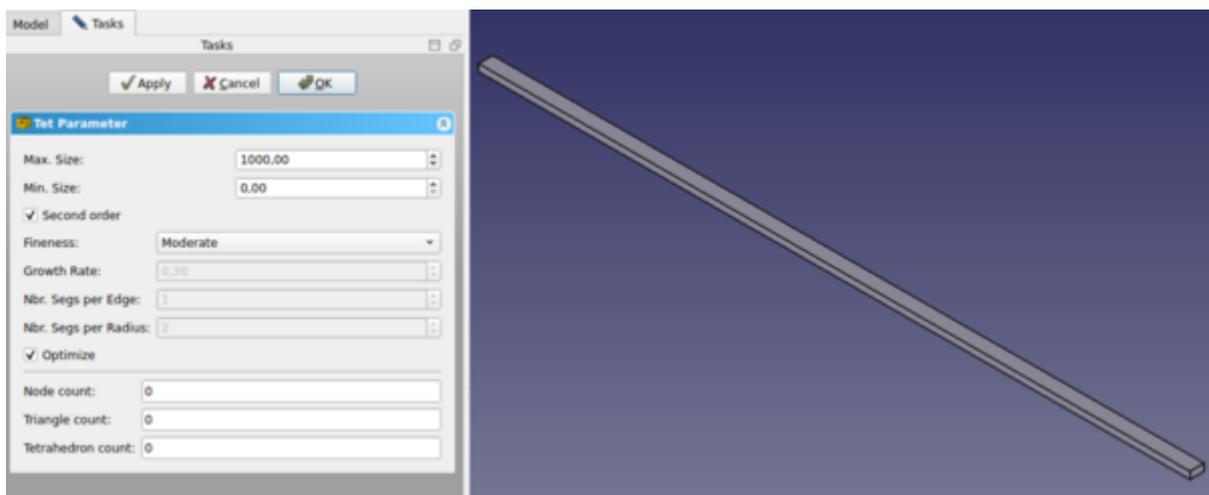


Création de l'analyse

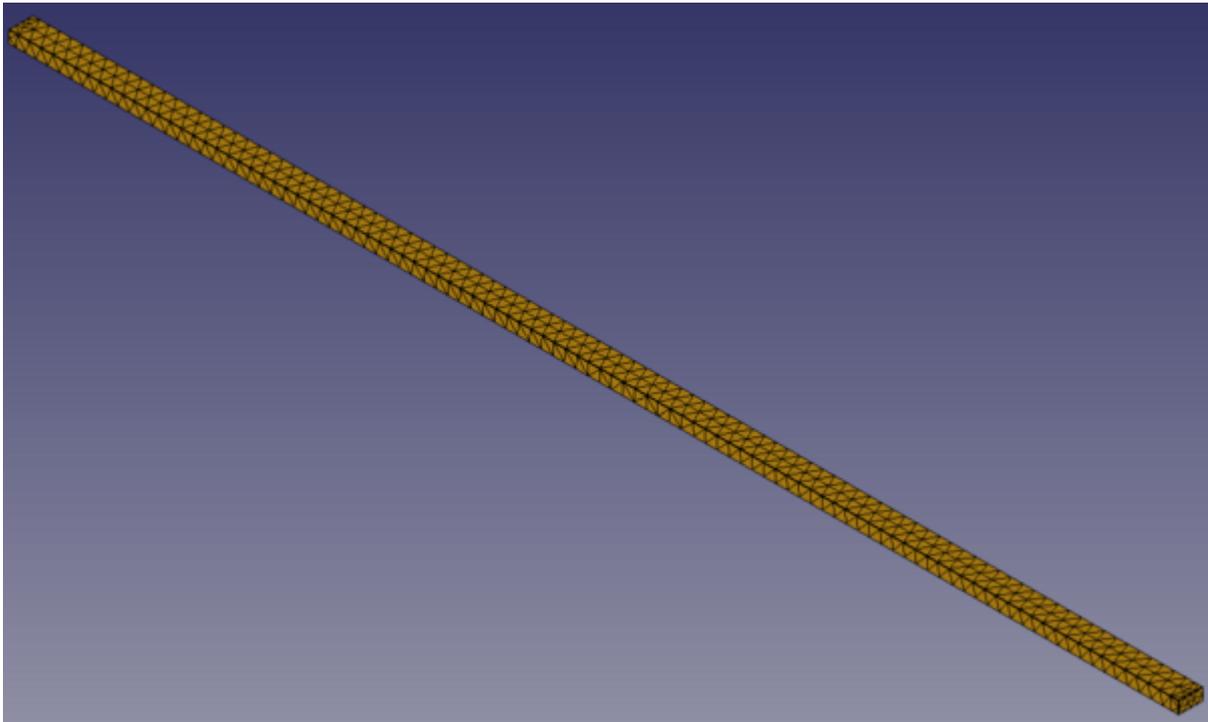
- Nous sommes maintenant prêts à commencer une analyse FEM. Passons à l'[atelier FEM](#).
- Appuyez sur le bouton [Conteneur d'analyse](#)
- Une nouvelle analyse sera créée et un panneau de configuration sera ouvert. Le bouton **Créer une analyse** met en place le cadre d'exécution d'une analyse

par éléments finis. Il crée un conteneur d'analyse qui organise les éléments clés tels que le maillage, les propriétés des matériaux, les contraintes (par exemple, les points fixes), les charges appliquées et le solveur. Ce bouton prépare essentiellement tout ce qui est nécessaire à la simulation, ce qui permet de passer à d'autres étapes comme le maillage et l'exécution du solveur pour analyser le comportement de l'objet dans les conditions définies.

- Nous allons commencer par créer le **maillage**. Pour cela, après avoir sélectionné notre corps, appuyez sur le bouton  [Mailler avec le mailleur Netgen](#). Cette option utilise le maillage de Netgen, un outil open-source utilisé pour créer des maillages tétraédriques de haute qualité, particulièrement adaptés aux géométries complexes dans l'analyse par éléments finis.
- Dans la fenêtre des paramètres du mailleur, nous allons rester simples et ne modifier que la taille maximale des cellules. L'option **Taille maximum** définit la plus grande taille autorisée pour les éléments de maillage individuels. Elle contrôle la grosseur ou la finesse du maillage. Une taille maximale plus grande entraînera un maillage plus grossier avec moins d'éléments, ce qui peut accélérer les calculs mais peut réduire la précision. Une taille maximale plus petite crée un maillage plus fin avec plus d'éléments, ce qui augmente la précision mais nécessite également plus de ressources de calcul. Réglez cette valeur sur **10** et appuyez sur **appliquer**.



- Notre maillage est prêt.



- Nous pouvons maintenant définir le matériau à appliquer à notre maillage en appuyant sur l'option  [Nouveau matériau](#). Le choix du matériau est crucial dans toute analyse, car différents matériaux aux propriétés variées se comporteront différemment dans les mêmes conditions. Des facteurs tels que la résistance, l'élasticité et la densité jouent un rôle important dans la manière dont un matériau réagit aux forces, aux pressions ou aux températures. La sélection du matériau approprié garantit des résultats de simulation précis, reflétant la manière dont l'objet réagirait dans des scénarios réels.
- Un panneau de tâches s'ouvre pour nous permettre de choisir un matériau. Dans la liste déroulante Matériau, choisissez le matériau **Steel-1C22** et appuyez sur **OK**.

F FEM material
⌵

Material

Category: Solid

Material card: Steel-1C22

Material name: Steel-1C22

Case hardened alloy steel

Editing material

use FreeCAD material editor use this task panel

Basic Properties

Density:

Mechanical Properties

Young's Modulus:

Poisson Ratio:

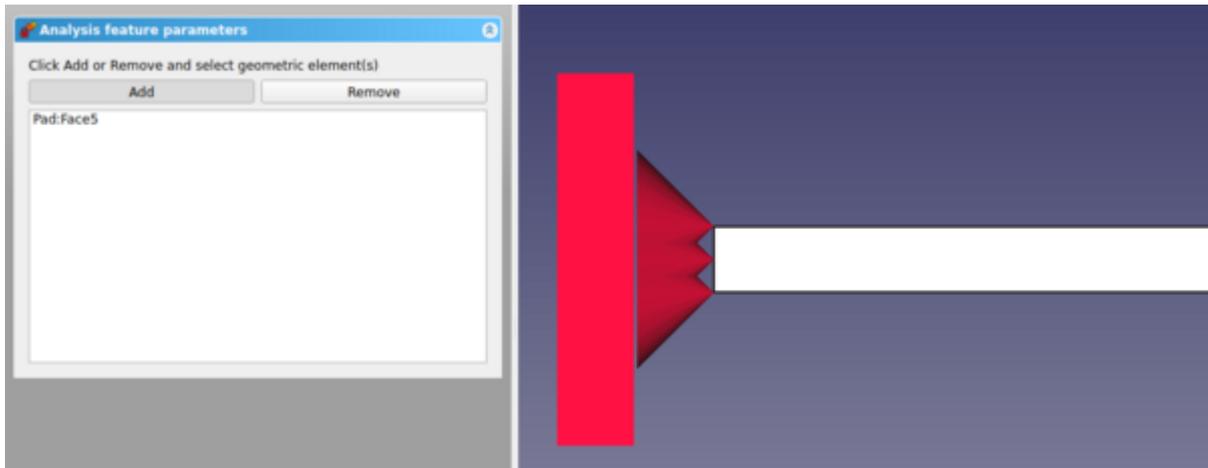
Thermal Properties

Thermal Conductivity:

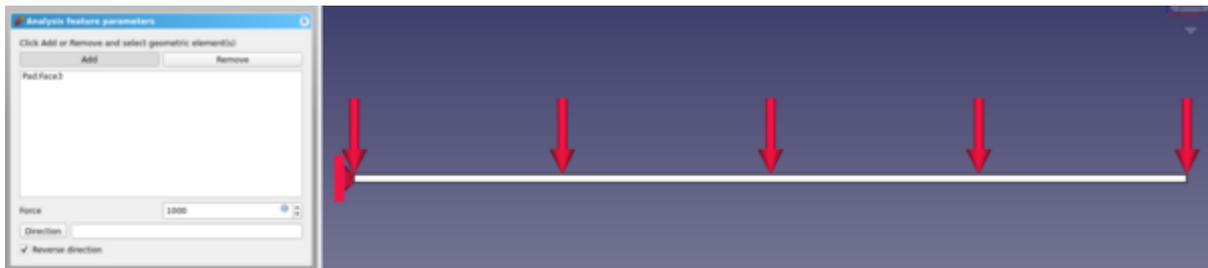
Expansion Coefficient:

Specific Heat:

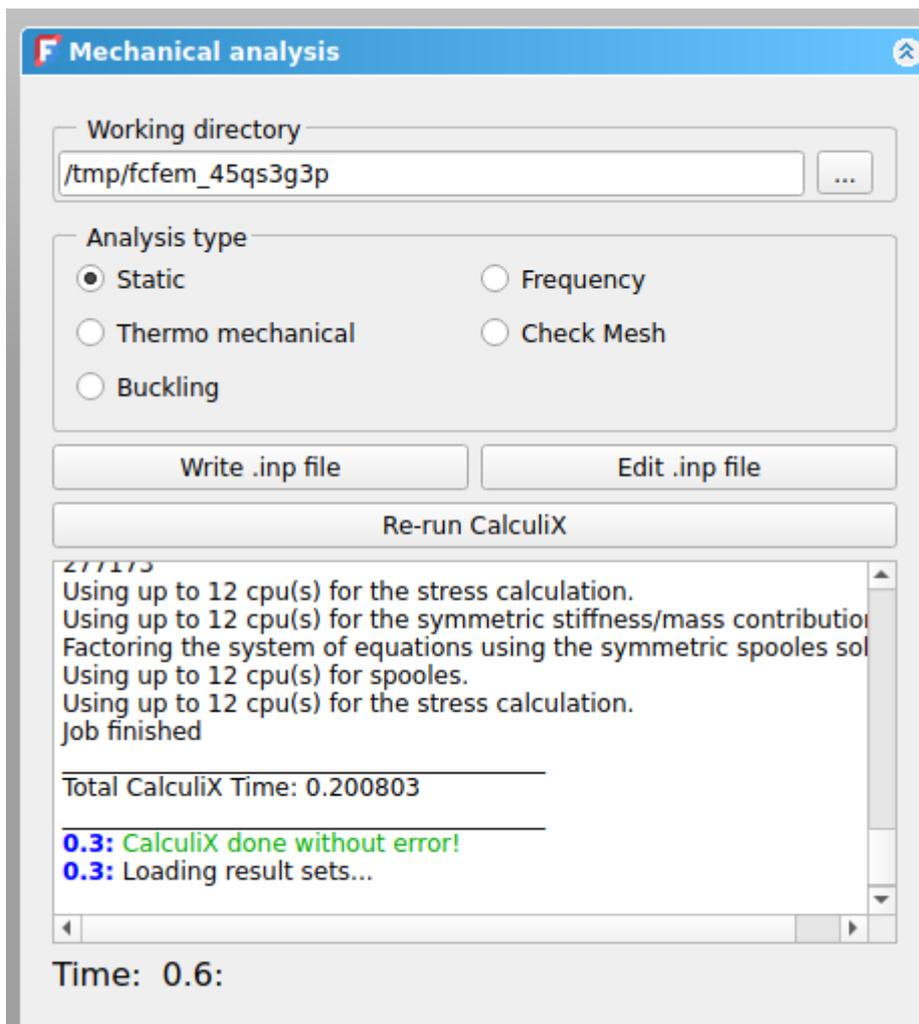
- L'étape finale consiste à appliquer des forces et des contraintes, traduisant les conditions physiques dans l'analyse FEM. Dans ce cas simple, nous avons une poutre qui est fixée d'un côté (représentant la fixation à un mur), tandis que l'autre côté est libre de se déplacer. Une force répartie est appliquée sur toute la longueur de la poutre, simulant la charge qu'elle subit dans des conditions réelles. Commençons par spécifier la face fixée au mur et qui ne peut donc pas bouger. Appuyez sur le bouton  [Condition de limite fixe](#).
- Appuyez sur le bouton **Ajouter** et sélectionnez la face gauche de notre poutre (celle à l'origine). Cliquez sur **Appliquer**. Cette face est maintenant désignée comme inamovible :



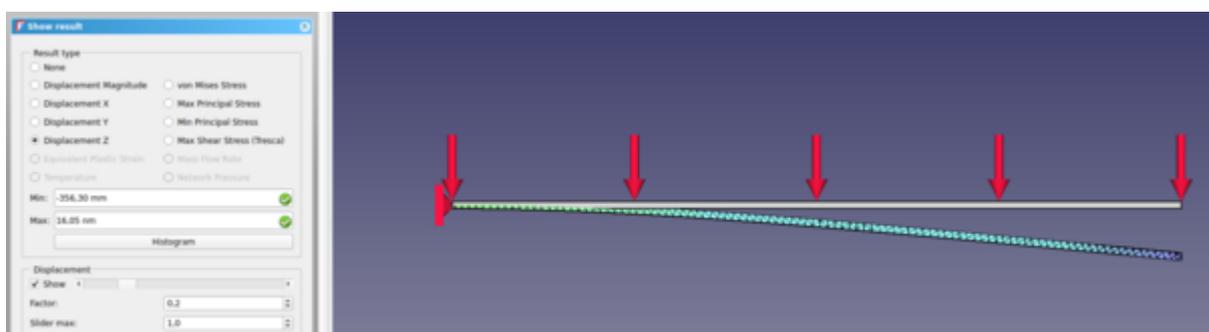
- Nous allons maintenant ajouter une charge répartie sur la face supérieure, qui pourrait représenter, par exemple, un poids massif placé sur la poutre. Pour cela, nous utiliserons une pression sur l'option  [Charge d'effort](#).
- Cliquez sur la face supérieure de la poutre, réglez la force sur **1000 N** et cliquez sur l'option **Inverser la direction**. Appuyez ensuite sur le bouton **OK**. Notre force est maintenant appliquée :



- Nous sommes maintenant prêts à commencer le calcul. Sélectionnez le  [solveur Calculix](#).
- Sélectionnez l'analyse **Statique** et cliquez sur **Générer le fichier .inp** pour créer le fichier d'entrée pour CalculiX. Cliquez ensuite sur **Lancer CalculiX**. La simulation va maintenant s'exécuter.



- Nous pouvons maintenant regarder les résultats. Cliquez sur l'option  [Afficher les résultats](#).
- Cochez l'option **Déplacement en Z**, qui est la coordonnée verticale pour notre cas.
- Vous pouvez voir les valeurs minimales et maximales du déplacement vertical. D'après l'analyse, le déplacement vertical maximal est de **-356,30 mm**. Cela correspond bien à notre solution analytique de **-357.14 mm**.
- Vous pouvez déplacer le curseur à côté. Vous pourrez voir la déformation augmenter au fur et à mesure que vous appliquerez une force plus importante :



Les résultats affichés par l'atelier FEM ne sont évidemment pas suffisantes pour prendre des décisions dans la vie réelle sur le dimensionnement des structures et les matériaux. Ils peuvent par contre donner déjà des informations intéressantes sur la

façon dont les forces s'appliquent sur une structure et quelles sont les zones faibles qui seront le plus stressées.

Lire plus d'informations

- [L'atelier FEM](#)
- [Installation des composants requis pour l'atelier FEM](#)
- [CalculiX](#)
- [NetGen](#)

Manuel : Création de rendus réalistes

Le [rendu photoréaliste](#) est le processus de création d'images très réalistes à partir de modèles 3D en simulant l'éclairage, les matériaux et les textures. Il est couramment utilisé dans des secteurs tels que le cinéma, les jeux vidéo et la conception de produits, où des visualisations photoréalistes sont nécessaires pour présenter des conceptions ou des concepts. Si le rendu permet de créer des images très proches de photographies réelles, il nécessite des outils spécialisés pour contrôler des éléments tels que l'éclairage, les reflets et les ombres.

FreeCAD, cependant, se concentre principalement sur la modélisation technique plutôt que sur les effets artistiques ou visuels. Son objectif premier est de créer des modèles 3D précis pour l'ingénierie, la conception et la fabrication. Par conséquent, FreeCAD ne dispose pas d'outils de rendu intégrés avancés pour le photoréalisme.

Cependant, FreeCAD propose l'atelier [Render](#), qui peut être installé en tant que module complémentaire (ce n'est pas l'un des ateliers par défaut). Cet atelier permet aux utilisateurs de connecter les modèles FreeCAD à des moteurs de rendu externes tels que Blender Cycles, LuxCoreRender ou POV-Ray. Grâce à l'atelier Render, les utilisateurs peuvent utiliser leurs modèles et tirer parti de ces puissants outils externes pour rendre leurs conceptions avec un éclairage et des textures réalistes. Cette approche permet à FreeCAD de rester léger et concentré, tout en offrant la flexibilité nécessaire à un rendu photoréaliste lorsque cela est nécessaire.

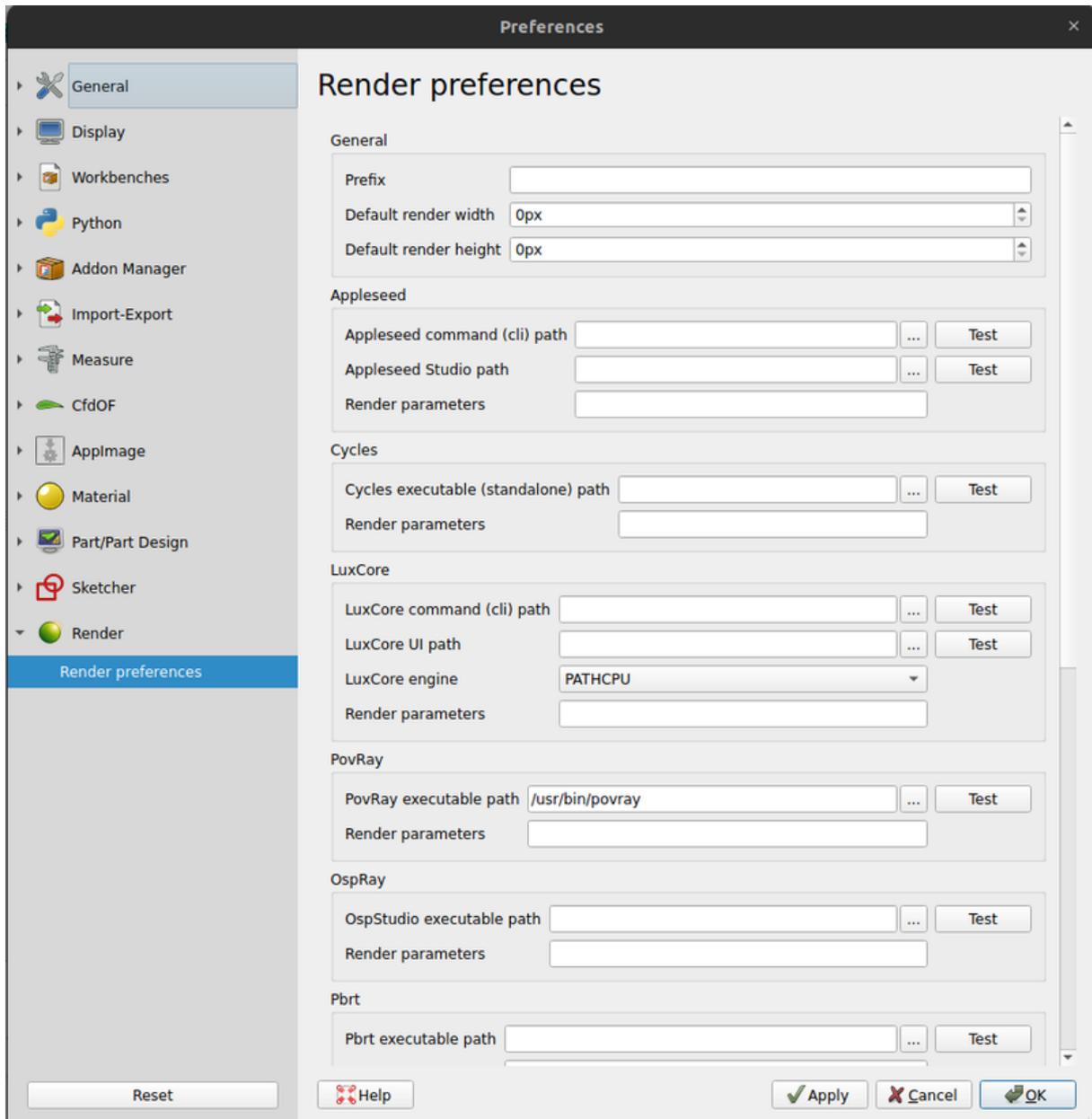
L'atelier Render de FreeCAD s'intègre à plusieurs moteurs de rendu externes, notamment [LuxCorerender](#), [POV-Ray](#) et [Blender Cycles](#). LuxCoreRender est un moteur de rendu moderne, basé sur la physique, qui produit des images photoréalistes, mais qui nécessite une puissance de calcul importante, en particulier pour les scènes de grande taille. POV-Ray, bien que plus ancien, reste un moteur [raytracing](#) fiable et moins gourmand en ressources, bien qu'il n'ait pas le réalisme des technologies plus récentes. Blender Cycles, accessible via FreeCAD lorsque Blender est installé, offre une solution de rendu puissante avec prise en charge du GPU et du CPU, produisant efficacement des images de haute qualité. Cependant, il faut installer Blender séparément et exporter les modèles vers Blender pour le rendu. Chaque moteur de rendu offre des avantages qui dépendent de l'équilibre entre le réalisme, les performances et les capacités du système de l'utilisateur. Chaque option a ses forces et ses faiblesses, en fonction du type d'image que l'on veut rendre. La meilleure façon de s'en rendre compte est de regarder des exemples sur le site web de chaque moteur.

Installation

Avant d'utiliser l'atelier Render dans FreeCAD, vous devrez installer à la fois l'atelier lui-même (comme indiqué dans [cette section](#)) et l'une des applications de rendu externes telles que LuxCoreRender, POV-Ray, ou Blender Cycles (avec Blender installé). L'installation de ces applications est généralement simple, car elles fournissent des installateurs pour différentes plateformes et sont souvent incluses dans les dépôts de logiciels des distributions Linux. Une fois ces outils installés, vous pourrez connecter FreeCAD à ces moteurs de rendu pour produire des images de haute qualité.

Une fois que POV-Ray ou LuxCorerender est installé, nous devons définir le chemin d'accès à son exécutable principal dans les préférences de FreeCAD. Cela n'est

généralement nécessaire que sur Windows et Mac. Sur Linux, FreeCAD va le choisir parmi les emplacements standard. L'emplacement des exécutables de Povray ou LuxRender peut être trouvé en recherchant simplement dans votre système les fichiers appelés Povray (ou Povray.exe sur Windows) et LuxRender (ou Luxrender.exe sur Windows). Dans l'onglet **Préférences**, vous pouvez désigner le chemin et définir certains paramètres.

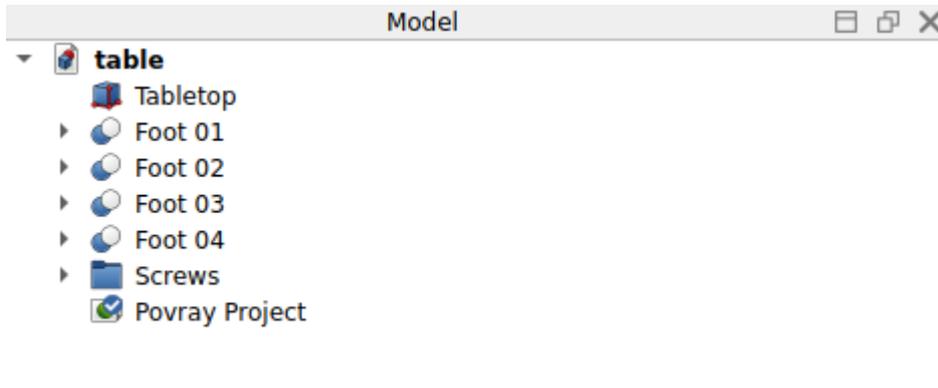


Rendu avec PovRay

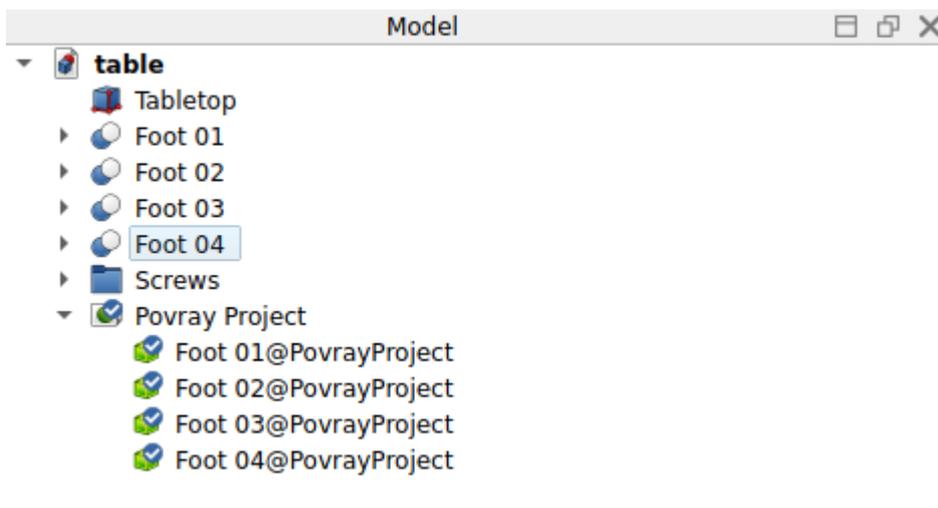
Nous allons utiliser la table que nous avons modélisée dans le chapitre sur la [Modélisation traditionnelle](#) pour produire des rendus avec PovRay.

- Commencez par charger le fichier table.FCStd que nous avons modélisé plus tôt ou à partir du lien au bas de ce chapitre et entrez dans l' [Render](#).

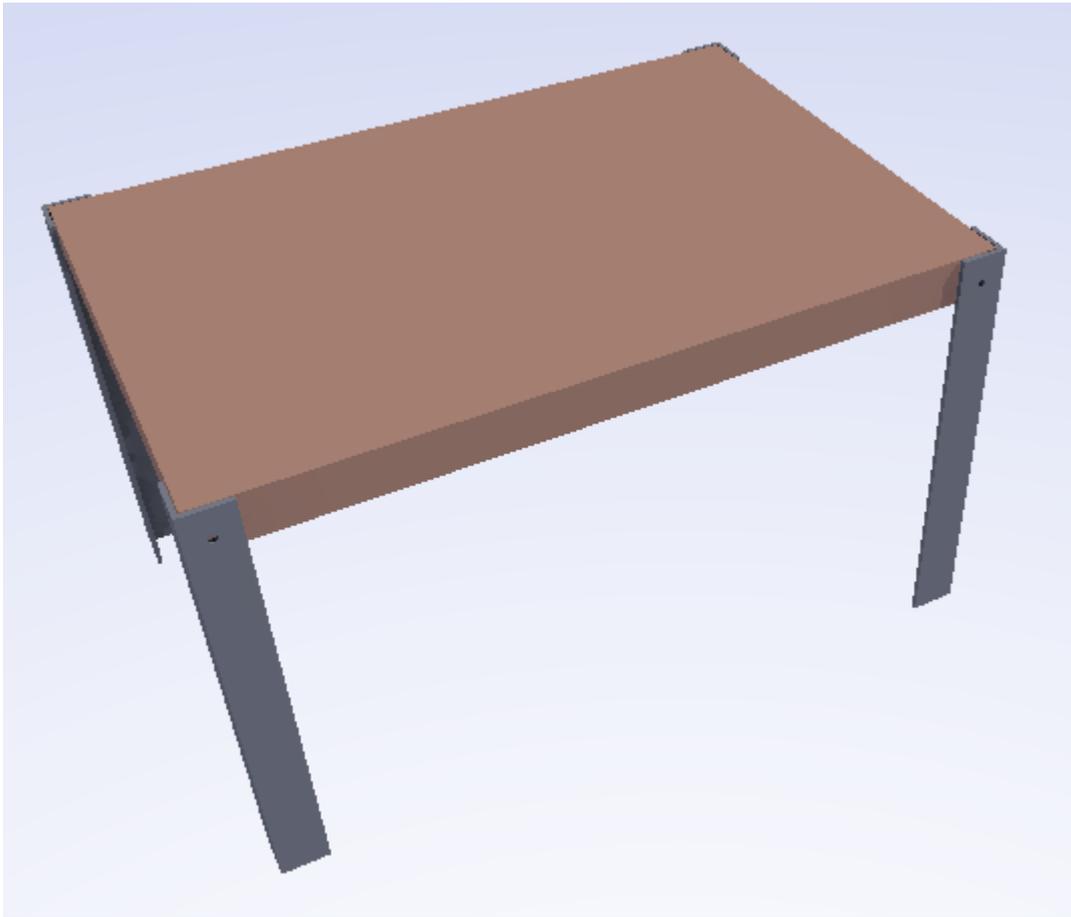
- Créez un projet de rendu en appuyant sur le bouton de la barre d'outils correspondant à votre moteur de rendu. Pour notre exemple, nous choisirons le moteur de rendu povray .
- Sélectionnez un modèle adapté à votre projet. Nous choisirons le modèle **povray_sunlight.pov**.
- Vous pouvez également essayer d'autres modèles après avoir créé un nouveau projet, simplement en modifiant sa propriété **Template**.
- Un nouveau projet a maintenant été créé :



- Vous pouvez ajouter les objets souhaités au projet en les sélectionnant et en appuyant sur l'option  [vue de rendu](#).



- Si nous le souhaitons, nous pouvons appliquer un matériau à nos corps en appuyant sur l'option  [Matériau](#). Dans notre cas, nous choisirons l'option matte.
- Nous pouvons maintenant appuyer sur le bouton  et notre résultat apparaîtra dans une fenêtre séparée.



À vrai dire, le résultat final n'est pas très impressionnant. Le processus de rendu est itératif et il faut du temps et de la patience pour obtenir des résultats de haute qualité. En outre, comme indiqué plus haut, POV-Ray est quelque peu limité en termes de réalisme. N'hésitez pas à expérimenter d'autres outils de rendu. La procédure reste en grande partie la même, la seule différence étant la sélection d'un autre moteur de rendu au début du processus.

Téléchargements

- Le modèle de table: <https://github.com/yorikvanhavre/FreeCADmanual/blob/master/files/table.FCStd>
- Le fichier produit lors de cet exercice: <https://github.com/yorikvanhavre/FreeCADmanual/blob/master/files/render.FCStd>

Lire plus d'informations

- [Blender](#)
- [POV-Ray](#)

Manuel : petite introduction à Python

[Python](#) est un langage de programmation open-source largement utilisé, reconnu pour sa simplicité, sa polyvalence et sa lisibilité. Il est souvent intégré dans les applications en tant que langage de script, et FreeCAD ne fait pas exception. Cette intégration permet aux utilisateurs d'automatiser des tâches, de personnaliser des flux de travail et d'étendre les fonctionnalités du logiciel bien au-delà de son interface graphique.

Python offre plusieurs avantages qui le rendent particulièrement adapté aux utilisateurs de FreeCAD. Il est convivial pour les débutants, avec une syntaxe claire et intuitive qui le rend facile à apprendre, même pour les personnes n'ayant aucune expérience préalable de la programmation. Cette accessibilité est particulièrement précieuse dans la communauté FreeCAD, où de nombreux utilisateurs viennent d'horizons divers, tels que l'ingénierie, l'architecture et le design, et n'ont pas de connaissances approfondies en matière de codage.

De plus, l'adoption généralisée de Python dans d'autres applications, telles que [Blender](#) pour la modélisation 3D, [Inkscape](#) pour les graphiques vectoriels, et [GRASS GIS](#) pour l'analyse géospatiale, en fait une compétence essentielle pour tous ceux qui cherchent à étendre leurs capacités de script. La maîtrise de Python dans FreeCAD améliore non seulement votre capacité à créer des outils et des macros personnalisés, mais fournit également des compétences transférables qui peuvent être appliquées à diverses plates-formes logicielles.

Dans FreeCAD, les scripts Python vous permettent de :

- Automatiser les tâches répétitives pour gagner du temps et réduire les erreurs.
- Créer des objets paramétriques personnalisés qui s'adaptent dynamiquement aux changements.
- Développer des macros et des outils personnalisés adaptés à des flux de travail spécifiques.
- Interagir avec l'interface de programmation d'applications (API) de FreeCAD pour accéder à la géométrie, aux scènes et aux éléments de l'interface utilisateur et les manipuler par programme.

En utilisant Python, les utilisateurs de FreeCAD peuvent exploiter tout le potentiel du logiciel, le transformant en un outil puissant et flexible adapté à leurs besoins uniques.

FreeCAD comprend une console Python avancée, accessible via **Affichage** → **Panneaux** → **Console Python**. Cet outil permet aux utilisateurs d'effectuer des opérations au-delà de l'interface graphique, telles que l'accès aux fonctionnalités avancées, le dépannage de la géométrie et l'automatisation des tâches. Il enregistre également les commandes Python pour les actions de l'interface graphique si l'option *Afficher les commandes des scripts dans la console Python* est activée sous **Édition** → **Préférences** → **Python** → **Macro**). En gardant la console ouverte, vous pouvez regarder le code Python se dérouler au fur et à mesure que vous travaillez, ce qui constitue un moyen intuitif d'apprendre le langage tout en explorant les capacités de FreeCAD. Enfin, FreeCAD dispose également d'un [système de macros](#), qui vous permet d'enregistrer des actions pour les rejouer plus tard. Ce système utilise également la console Python, en enregistrant simplement tout ce qui y est fait.

```
Python console
Python 3.11.9 | packaged by conda-forge | (main, Apr 19 2024, 18:53:30) [GCC 12.3.0] on linux
Type 'help', 'copyright', 'credits' or 'license' for more information.
>>> # Gui.runCommand('Start_Start',0)
>>> # Gui.runCommand('Std_Workbench',9)
>>> # Gui.runCommand('Std_ViewStatusBar',1)
>>> ### Begin command Std_New
>>> App.newDocument()
>>> # App.setActiveDocument("Unnamed")
>>> # App.ActiveDocument=App.getDocument("Unnamed")
```

Dans ce chapitre, nous allons découvrir de façon très générale le langage Python. Si vous souhaitez en apprendre plus, la documentation wiki de FreeCAD possède une section pour utilisateurs avancés en [programmation Python](#).

Ecrire du code Python

Dans FreeCAD, vous pouvez écrire du code Python de deux manières principales : à travers la console Python (**Affichage** → **Panneaux** → **Console Python**) ou en utilisant l'éditeur de macros (**Macro** → **Macros** → **Créer**). La console Python vous permet de saisir des commandes une par une, qui sont exécutées immédiatement après avoir appuyé sur la touche Retour, ce qui la rend idéale pour les tests rapides ou l'exploration interactive. L'éditeur de macros, quant à lui, est utilisé pour écrire et enregistrer des scripts plus complexes composés de plusieurs lignes de code. Ces macros peuvent être exécutées dans leur ensemble plus tard à partir de la fenêtre Macros, ce qui constitue un moyen puissant d'automatiser les tâches répétitives et d'étendre les fonctionnalités de FreeCAD. Dans ce chapitre, vous pourrez utiliser les deux méthodes, mais il est fortement recommandé d'utiliser la console Python, car elle vous informera immédiatement de toutes les erreurs que vous faites en tapant.

Si c'est la première fois que vous programmez en Python, pensez à lire cette courte [introduction au langage Python](#) avant d'aller plus loin. Cela rendra les concepts de base de Python plus clairs.

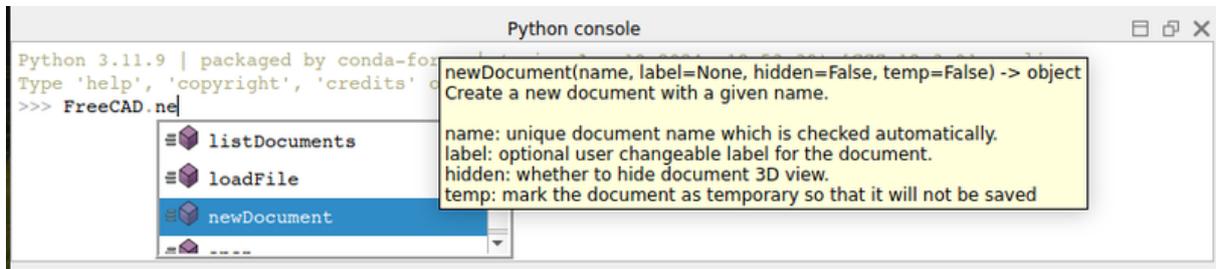
Manipuler des objets FreeCAD

Commençons pas créer un nouveau document vide:

```
doc = FreeCAD.newDocument()
```

Dans la console FreeCAD Python, dès que vous tapez FreeCAD. (le mot « FreeCAD » suivi d'un point), une fenêtre d'autocomplétion apparaît. Cette fonction n'accélère pas seulement votre flux de travail en suggérant des commandes disponibles, mais vous aide également à découvrir de nouvelles fonctions et caractéristiques dans FreeCAD. Chaque entrée de la liste est accompagnée d'une infobulle expliquant sa fonction, ce qui facilite la compréhension et l'exploration des fonctionnalités disponibles. Cette fonction d'autocomplétion est particulièrement utile

pour les débutants qui apprennent les scripts Python et pour les utilisateurs avancés qui naviguent efficacement dans la vaste API de FreeCAD. Prenez le temps d'explorer les options de la fenêtre d'autocomplétion, vous découvrirez peut-être des commandes qui simplifieront votre flux de travail ou ouvriront de nouvelles possibilités.



Taper **FreeCAD.newDocument()** crée un nouveau document vide dans FreeCAD, tout comme cliquer sur le bouton **Nouveau document** dans la barre d'outils. Lorsque vous exécutez **doc = FreeCAD.newDocument()**, le nouvel objet document est assigné à la variable **doc**, ce qui vous permet de le manipuler par programme. En utilisant **doc**, vous pouvez ajouter des objets, modifier des propriétés ou enregistrer le document.

En Python, le point (.) est utilisé pour indiquer qu'un élément est contenu dans un autre. Par exemple, **newDocument** est une fonction du **module FreeCAD**, c'est pourquoi nous écrivons **FreeCAD.newDocument**. La fenêtre d'autocomplétion qui apparaît montre tout ce qui est disponible dans le module **FreeCAD**. Si vous tapez un point après **newDocument** (sans ajouter les parenthèses), cela affichera tout ce qui appartient à la fonction **newDocument**. Ceci illustre la façon dont Python organise et accède aux objets et à leurs composants. Il est important de noter que les parenthèses sont obligatoires lors de l'appel d'une fonction Python, car elles signalent l'exécution de la fonction.

Revenons maintenant à notre document. Voyons ce que nous pouvons en tapant ce qui suit et explorons les options disponibles:

```
doc.
```

Dans FreeCAD, les conventions d'appellation des commandes Python peuvent vous aider à comprendre leur objectif :

- Les noms commençant par une lettre majuscule sont généralement des attributs, qui stockent des valeurs ou des données, telles que les propriétés d'un objet.
- Les noms commençant par une lettre minuscule sont généralement des fonctions (également appelées méthodes), qui effectuent des actions ou des opérations, telles que la création ou la modification d'objets.
- Les noms commençant par un trait de soulignement (_) sont généralement destinés à un usage interne au module et peuvent être ignorés.

Cette distinction vous aide à naviguer dans l'API de FreeCAD et à sélectionner la commande appropriée à vos besoins. Par exemple, vous pouvez utiliser une

méthode pour ajouter un nouvel objet à un document. La méthode exécute l'action de création et d'ajout de l'objet, tandis que les attributs stockent les propriétés de l'objet. La compréhension de cette structure facilite l'exploration des fonctionnalités disponibles, en particulier lors de l'utilisation de la fonction d'autocomplétion ou de la lecture d'infobulles. Ajoutons une boîte.

```
box = doc.addObject("Part::Box", "myBox")
```

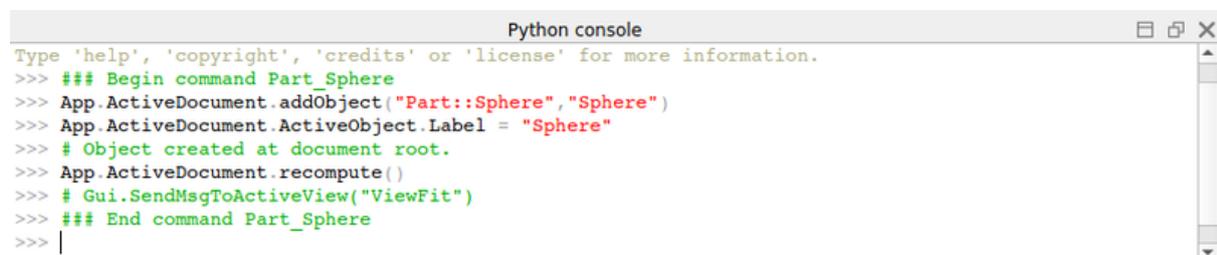
La commande **box = doc.addObject("Part::Box", "myBox")** ajoute un nouvel objet box 3D au document. Voici une explication simple :

- **doc.addObject** : indique à FreeCAD d'ajouter un nouvel objet au document.
- **Part::Box** : spécifie le type d'objet à créer, dans ce cas, une boîte 3D.
- **myBox** : cette commande attribue un nom au nouvel objet, ce qui permet de l'identifier et de le référencer plus facilement par la suite.

Le résultat de cette commande est qu'une boîte apparaît dans le document actif et que la variable boîte stocke une référence à cette boîte afin que vous puissiez la modifier ou interagir avec elle ultérieurement. Notre boîte est ajoutée dans l'arborescence, mais rien ne se passe encore dans la vue 3D, car lorsqu'on travaille à partir de Python, le document n'est jamais recalculé automatiquement. Nous devons le faire manuellement, chaque fois que cela est nécessaire :

```
doc.recompute()
```

Notre boîte est maintenant apparue dans la vue 3D. De nombreux boutons de la barre d'outils qui ajoutent des objets dans FreeCAD font en fait deux choses : ajouter l'objet et recalculer. Essayez maintenant d'ajouter une sphère avec le bouton approprié dans l'atelier Part, et vous verrez les deux lignes de code Python s'exécuter l'une après l'autre.



```
Python console
Type 'help', 'copyright', 'credits' or 'license' for more information.
>>> ### Begin command Part_Sphere
>>> App.ActiveDocument.addObject("Part::Sphere", "Sphere")
>>> App.ActiveDocument.ActiveObject.Label = "Sphere"
>>> # Object created at document root.
>>> App.ActiveDocument.recompute()
>>> # Gui.SendMsgToActiveView("ViewFit")
>>> ### End command Part_Sphere
>>> |
```

Vous pouvez obtenir une liste de tous les types d'objets possibles, comme Part :: Box:

```
doc.supportedTypes()
```

Explorons maintenant le contenu de notre boîte :

```
box.
```

Vous verrez immédiatement un certain nombre de choses très intéressantes telles que :

```
box.Height
```

Cela affichera la hauteur actuelle de notre boîte. Maintenant, essayons de changer cela :

```
box.Height = 5
```

Si vous sélectionnez votre boîte avec la souris, vous verrez cela dans le panneau des propriétés, sous l'onglet **Données**, notre propriété **Hauteur** apparaître avec la nouvelle valeur. Toutes les propriétés d'un objet FreeCAD qui apparaissent dans les onglets **Données** et **Affichage** sont directement accessibles par Python aussi, par leur noms, comme nous l'avons fait avec la propriété Height. Les propriétés des données sont accessibles directement depuis l'objet lui-même, par exemple :

```
box.Length
```

Dans FreeCAD, les propriétés visuelles sont gérées par **ViewObject**. Chaque objet FreeCAD a un ViewObject associé, qui stocke des informations sur la façon dont l'objet est affiché dans l'interface graphique, comme la couleur, la transparence ou la visibilité. Cependant, le ViewObject n'est accessible que lorsque FreeCAD fonctionne avec son interface graphique. Si FreeCAD est lancé dans un mode non graphique, comme à partir d'un terminal avec l'option de ligne de commande -c ou lorsqu'il est utilisé comme bibliothèque Python dans un script externe, le ViewObject n'est pas disponible. Ceci est dû au fait qu'il n'y a pas de représentation visuelle de l'objet dans ces modes, puisque l'interface graphique n'est pas chargée.

Essayez l'exemple suivant pour accéder à la couleur de trait de notre boîte :

```
box.ViewObject.LineColor
```

Vecteurs et emplacements

Les vecteurs sont un concept fondamental dans toute application 3D. Un vecteur est essentiellement une liste de trois nombres (x, y et z) qui décrivent un point, une position ou une direction dans l'espace 3D. Les vecteurs sont essentiels pour définir la géométrie, les transformations et les interactions dans l'environnement 3D. Ils servent d'éléments de base pour des opérations telles que les translations, les rotations et les mises à l'échelle.

Dans FreeCAD, les vecteurs sont largement utilisés pour créer et manipuler des objets. Ils permettent un large éventail d'opérations mathématiques, telles que l'addition, la soustraction, les produits en croix, les produits en points et les projections. Ces opérations vous permettent de calculer des distances, des angles et des directions ou de définir des relations entre des objets dans l'espace.

Il est essentiel de comprendre les vecteurs et leur fonctionnement pour pouvoir créer des scripts et des personnalisations efficaces dans FreeCAD. Par exemple, les vecteurs sont utilisés pour positionner des objets, définir leur orientation, ou même calculer les trajectoires pour des opérations complexes telles que les balayages ou les lissages.

Dans FreeCAD, les vecteurs sont représentés à l'aide de la classe **Vector**, qui fournit diverses méthodes et propriétés pour effectuer des opérations et accéder à leurs composants. La maîtrise de ces capacités améliorera considérablement votre capacité à interagir avec l'environnement 3D de FreeCAD par le biais de programmes. Dans FreeCAD, les vecteurs fonctionnent comme suit :

```
myvec = FreeCAD.Vector(2, 0, 0)
print(myvec)
print(myvec.x)
print(myvec.y)
othervec = FreeCAD.Vector(0, 3, 0)
sumvec = myvec.add(othervec)
```

Voici une brève description des commandes ci-dessus :

- **myvec = FreeCAD.Vector(2,0,0)** : crée un vecteur myvec avec X = 2, Y = 0, Z = 0.
- **print(myvec)** : affiche le vecteur myvec avec ses composantes (X, Y, Z).
- **print(myvec.x)** : affiche la composante X de myvec.
- **print(myvec.y)** : affiche la composante Y de myvec.
- **othervec = FreeCAD.Vector(0,3,0)** : crée un second vecteur othervec avec X = 0, Y = 3, Z = 0.
- **sumvec = myvec.add(othervec)** : additionne myvec et othervec pour créer un nouveau vecteur sumvec contenant la somme de leurs composantes.

Une autre caractéristique commune des objets FreeCAD est leur **Positionnement** (placement). Comme nous l'avons vu dans les chapitres précédents, chaque objet a une propriété de placement qui contient la position (Base) et l'orientation (Rotation)

de l'objet. Ces propriétés sont faciles de manipuler avec Python, par exemple pour déplacer notre objet :

```
print(box.Placement)
print(box.Placement.Base)
box.Placement.Base = sumvec
otherpla = FreeCAD.Placement()
otherpla.Base = FreeCAD.Vector(5, 5, 0)
box.Placement = otherpla
```

Voici une brève description des commandes ci-dessus :

- **print(box.Placement)** : affiche le placement de la boîte, ce qui inclut sa position, sa rotation et son orientation dans l'espace 3D.
- **print(box.Placement.Base)** : affiche la position (Base) de la boîte, qui est un vecteur représentant son emplacement dans l'espace 3D.
- **box.Placement.Base = sumvec** : définit la position de base (emplacement) de la boîte au vecteur sumvec, déplaçant ainsi la boîte vers cette nouvelle position.
- **otherpla = FreeCAD.Placement()** : crée un nouvel objet de placement nommé otherpla.
- **otherpla.Base = FreeCAD.Vector(5,5,0)** : définit la position de base de otherpla à un vecteur aux coordonnées (5, 5, 0).
- **box.Placement = otherpla** : applique otherpla à la boîte, en mettant à jour son placement à la nouvelle position et à l'orientation définies par otherpla.

Lire plus d'informations

- [Python](#)
- [Macros](#)
- [Introduction à Python](#)
- [Tutoriel de scripts Python](#)
- [Portail d'utilisation avancée](#)

Manuel : Création et manipulation de la géométrie

Dans les chapitres précédents, nous avons exploré les différents ateliers de FreeCAD et comment chacun d'entre eux présente son propre ensemble d'outils et de types de géométrie. Les mêmes principes s'appliquent lorsque l'on travaille avec FreeCAD par le biais de scripts Python.

Nous avons également observé que la plupart des ateliers de FreeCAD s'appuient sur un atelier fondamental : l' [atelier Part](#). De nombreux autres ateliers, tels que l' [atelier Draft](#), utilisent les outils et la géométrie de l'atelier Part, ce qui est exactement ce que nous ferons dans ce chapitre - utiliser Python pour créer et manipuler la géométrie de la pièce.

Pour commencer à travailler avec la géométrie Part en Python, nous devons effectuer l'équivalent en script du passage à l'atelier Part : importer le module Part.

```
import Part
```

Prenez le temps d'explorer le module Part en tapant **Part.** dans la console Python et en parcourant les méthodes et attributs disponibles dans la fenêtre d'autocomplétion. C'est un excellent moyen de se familiariser avec les fonctionnalités offertes par le module. Vous trouverez une variété de fonctions pratiques, telles que `makeBox` et `makeCircle`, qui vous permettent de créer rapidement des formes géométriques et des objets à l'aide d'une seule commande. La plupart de ces fonctions proposent également des paramètres facultatifs, ce qui vous permet de contrôler précisément les dimensions et l'emplacement.

Passez un peu de temps à parcourir le contenu du module vous aidera non seulement à comprendre quels sont les outils à votre disposition, mais vous donnera également un aperçu du fonctionnement de l'atelier Part sous le capot. Ces connaissances fondamentales s'avéreront précieuses lorsque nous commencerons à créer et à manipuler des géométries par programme. Tapez la commande suivante

```
Part.makeBox(3, 5, 7)
```

Cette commande crée une boîte 3D, également connue sous le nom de prisme rectangulaire, avec des dimensions spécifiques. Le premier paramètre, 3, définit la longueur de la boîte sur l'axe X. Le deuxième paramètre, 5, définit la largeur sur l'axe Y. Le troisième paramètre, 7, spécifie la hauteur sur l'axe Z. Le deuxième paramètre, 5, définit la largeur le long de l'axe Y et le troisième paramètre, 7, spécifie la hauteur le long de l'axe Z. Bien que cette fonction génère la géométrie de la boîte, elle ne l'ajoute pas automatiquement au document FreeCAD actif. Dans la console Python, vous verrez ce qui suit :

```
<Solid object at 0x5f43600>
```

La sortie **<Solid object at 0x5f43600>** indique qu'une Part Shape a été créée en mémoire. Il s'agit d'un objet géométrique stocké à une adresse mémoire spécifique, comme le montre la valeur hexadécimale (0x5f43600). Cependant, il est important de comprendre que ce que nous avons créé ici n'est pas encore un objet document FreeCAD. Il existe seulement en tant que forme géométrique brute dans la mémoire.

Cette distinction met en évidence un concept fondamental de FreeCAD : les objets et leur géométrie sont indépendants. Un objet document FreeCAD sert de conteneur qui héberge une forme. Ces objets document peuvent avoir des propriétés supplémentaires, telles que la longueur, la largeur et la hauteur, et ils peuvent être paramétriques. Les objets paramétriques recalculent leur géométrie (ou leur forme) de manière dynamique chaque fois qu'une de leurs propriétés est modifiée. Par exemple, la modification de la longueur d'une boîte paramétrique régénérera automatiquement sa forme avec la valeur mise à jour.

Dans ce cas, nous avons créé manuellement une forme à l'aide de la fonction **Part.makeBox()**. Cette forme est un objet non paramétrique, ce qui signifie qu'elle ne sera pas mise à jour automatiquement en fonction des propriétés - elle est statique à moins que nous ne la manipulions par programme. Pour que cette forme fasse partie du document FreeCAD actif, il faudrait qu'elle soit assignée à un objet document (comme un **Part::Feature**), ce qui la relierait à l'interface graphique et la rendrait visible et gérable dans l'environnement FreeCAD.

Cette séparation entre les formes et les objets documentaires est ce qui rend FreeCAD très polyvalent, permettant aux utilisateurs de manipuler les formes de manière programmatique et de les intégrer dans un flux de travail de modélisation paramétrique selon les besoins.

Nous pouvons maintenant créer facilement un document objet "générique" dans le document actuel (assurez-vous que vous avez au moins un nouveau document ouvert) et donnez-lui une forme de boîte comme celle que nous venons de faire :

```
boxShape = Part.makeBox(3, 5, 7)
myObj =
FreeCAD.ActiveDocument.addObject("Part::Feature", "MyNewBox")
myObj.Shape = boxShape
FreeCAD.ActiveDocument.recompute()
```

Voici le détail des commandes précédentes :

- **boxShape = Part.makeBox(3,5,7)** : crée une boîte 3D de dimensions 3x5x7 (longueur, largeur et hauteur) et la stocke en tant que Part Shape dans la

variable `boxShape`. Cette forme n'existe qu'en mémoire et ne fait pas encore partie du document FreeCAD.

- **`myObj = FreeCAD.ActiveDocument.addObject("Part::Feature", "MyNewBox")`** : ajoute un nouvel objet `Part::Feature` nommé "MyNewBox" au document FreeCAD actif et l'affecte à la variable `myObj`. Le nouvel objet apparaîtra dans l'arborescence du document FreeCAD.
- **`myObj.Shape = boxShape`** : lie la géométrie `boxShape` à la propriété `Shape` de `myObj`, intégrant la géométrie dans le document FreeCAD.
- **`FreeCAD.ActiveDocument.recompute()`** : met à jour le document pour refléter les changements, en s'assurant que le nouvel objet et sa géométrie apparaissent dans l'interface graphique.

Remarquez comment nous avons traité **`myObj.Shape`**. Cela a été fait de la même manière que dans le chapitre précédent, où nous avons modifié d'autres propriétés d'un objet, comme **`box.Height = 5`**. En fait, **`Shape`** est aussi une propriété, tout comme **`Height`**. Cependant, au lieu de prendre un nombre, **`Shape`** nécessite une `Part Shape`. Dans le prochain chapitre, nous verrons de plus près comment ces objets paramétriques sont construits.

Pour l'instant, explorons les formes de pièces plus en détail. Dans le chapitre sur la modélisation traditionnelle avec l'atelier Part, nous avons présenté un tableau expliquant comment les Part Shapes sont construites et les différents composants dont elles sont constituées, tels que les **sommets**, les **arêtes** et les **faces**. Ces mêmes composants sont disponibles lorsque l'on travaille avec des formes partielles en Python, ce qui permet une exploration et une manipulation détaillées de la géométrie. Les Part Shapes dans FreeCAD ont toujours les attributs suivants :

- **Vertex** : points dans l'espace 3D qui définissent les coins ou les extrémités de la géométrie.
- **Edges** : lignes droites ou courbes reliant deux sommets.
- **Wires** : boucles fermées ou ouvertes formées par une ou plusieurs arêtes connectées.
- **Faces** : surfaces entourées par un ou plusieurs polygones.
- **Shells** : groupes de faces connectées, formant une surface continue.
- **Solids** : volumes 3D entourés d'une ou plusieurs coques.

Tous ces attributs sont représentés sous forme de listes en Python. Chaque liste peut contenir un nombre quelconque d'éléments ou être vide, en fonction de la forme explorée. Par exemple, une boîte aura huit **Vertexes**, **douze Edges**, **six Faces**, **une Shell et un Solid**, **tandis qu'une ligne n'aura que deux Vertexes** et une **Edge**, tous les autres attributs étant vides. Ces composants sont des éléments fondamentaux de la géométrie d'une pièce et sont accessibles et manipulables par programme. Comprendre comment ils interagissent permet d'exercer un contrôle puissant sur la création et la modification des modèles 3D. Nous pouvons accéder à ces listes comme suit :

```
print(boxShape.Vertexes)
print(boxShape.Edges)
print(boxShape.Wires)
print(boxShape.Faces)
```

```
print(boxShape.Shells)
print(boxShape.Solids)
```

Trouvons l'aire de chaque face de notre boîte ci-dessus : (Veillez à indenter la deuxième ligne, comme elle apparaît ci-dessous. Appuyez deux fois sur Entrée après la dernière ligne pour exécuter la commande Python).

```
for f in boxShape.Faces:
    print(f.Area)
```

Ou pour chaque arête, son point de départ et son point final:

```
for e in boxShape.Edges:
    print("New edge")
    print("Start point:")
    print(e.Vertexes[0].Point)
    print("End point:")
    print(e.Vertexes[1].Point)
```

Comme vous le voyez, si notre `boxShape` a un attribut "Vertexes", chaque bord de la `boxShape` a également un attribut "Vertexes". Comme on peut s'y attendre, la `boxShape` aura 8 sommets, tandis que l'arête n'en aura que 2, qui font tous deux partie de la liste de 8.

Nous pouvons toujours vérifier quel est le type de forme :

```
print(boxShape.ShapeType)
print(boxShape.Faces[0].ShapeType)
print(boxShape.Vertexes[2].ShapeType)
```

Voici une brève explication des commandes ci-dessus :

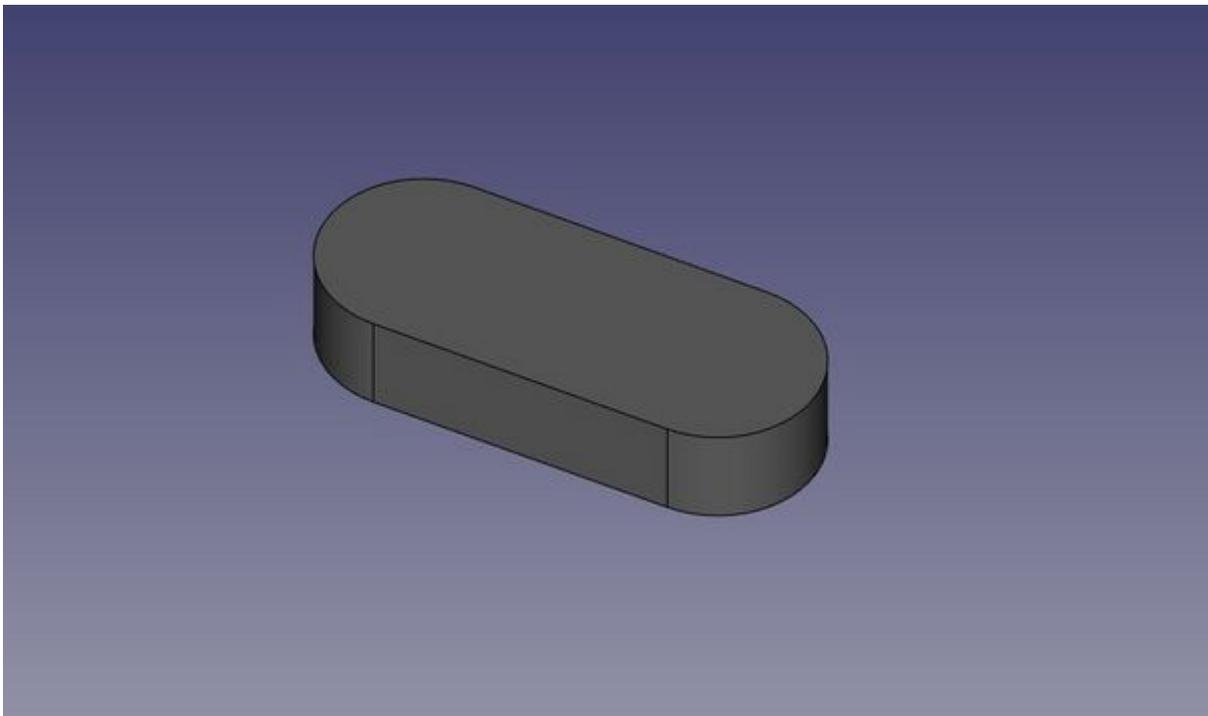
- **print(boxShape.ShapeType)** : affiche le type de la forme de premier niveau représentée par **boxShape**. Dans ce cas, puisque **boxShape** a été créée

comme une boîte à l'aide de **Part.makeBox**, la sortie sera "Solid", indiquant que la forme est un objet solide en 3D.

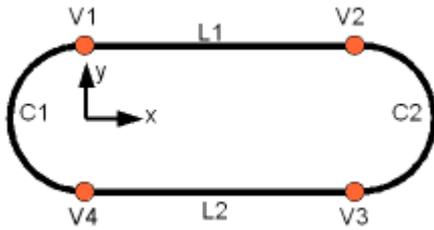
- **print(boxShape.Faces[0].ShapeType)** : accède à la première face de la liste **Faces** de **boxShape** (index 0) et affiche son type de forme. Pour une boîte, chaque face est une surface plane, la sortie sera donc "Face".
- **'print(boxShape.Vertexes[2].ShapeType)** : accède au troisième sommet de la liste **Vertexes** de **boxShape** (index 2) et affiche son type de forme. Comme il s'agit d'un point spécifique dans l'espace 3D, la sortie sera "Vertex".

Pour résumer le concept des Part Shapes : tout commence par des **Vertex** (sommet ou point), les éléments les plus basiques de la géométrie. En utilisant un ou deux **Vertex**, vous pouvez créer un **Edge** (arête) (notez que les cercles complets ne nécessitent qu'un seul **Vertex**). Une ou plusieurs **Arêtes** peuvent alors former une **Polyligne**, qui peut être ouverte ou fermée. Lorsque vous avez un ou plusieurs **Polylignes** fermées, vous pouvez créer une **Face**. Les **Polylignes** supplémentaires à l'intérieur de la **Polyligne** principale agiront comme des **trous** dans la **Face**. La combinaison d'une ou de plusieurs **Faces** permet de construire une **Coque**, qui est essentiellement une collection de surfaces connectées. Si une **Coque** est entièrement fermée et étanche, elle peut alors être utilisée pour former un **Solide**, un objet 3D avec un volume. Enfin, un nombre quelconque de formes de n'importe quel type, y compris des **Sommets**, des **Arêtes**, des **Polylignes**, des **Faces**, des **Coques** ou des **Solides**, peuvent être regroupées en un **Composé**, qui agit comme un conteneur pour de multiples formes.

Nous pouvons maintenant essayer de créer des formes complexes à partir de zéro, en construisant tous leurs composants un par un. Par exemple, essayons de créer un volume comme celui-ci:



Nous commencerons par créer une forme planeire comme celle-ci :



D'abord, créons les quatre points de base :

```
V1 = FreeCAD.Vector(0, 10, 0)
V2 = FreeCAD.Vector(30, 10, 0)
V3 = FreeCAD.Vector(30, -10, 0)
V4 = FreeCAD.Vector(0, -10, 0)
```

Ensuite, nous pouvons créer les deux segments de droites :



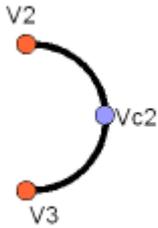
```
L1 = Part.LineSegment(V1, V2)
L2 = Part.LineSegment(V4, V3)
```

Remarquez que nous n'avons pas besoin de créer des **Vertexes** explicitement. Au lieu de cela, nous pouvons directement créer des **Part.LineSegments** en utilisant des **FreeCAD Vectors**. En effet, à ce stade, nous travaillons avec une géométrie de base, et non avec des **arêtes** réelles. Un **Part.LineSegment**, tout comme **Part.Circle**, **Part.Arc**, **Part.Ellipse**, ou **Part.BSpline**, définit la géométrie sous-jacente mais ne génère pas d'arête en soi. Dans FreeCAD, les arêtes sont toujours construites à partir d'une telle géométrie de base, qui est stockée dans l'attribut **Curve** de l'**Arête**. Cela signifie qu'une arête est essentiellement une enveloppe autour de la géométrie de base, héritant de ses propriétés. Si vous avez une arête, vous pouvez accéder à sa géométrie sous-jacente en vous référant à l'attribut de la courbe. La commande suivante :

```
print(Edge.Curve)
```

vous permet de comprendre la structure sous-jacente de l'arête et la manière dont elle a été construite. Revenons maintenant à notre exercice et construisons les segments d'arc. Pour créer un arc, nous avons besoin de trois points : un point de

départ, un point d'arrivée et un point central qui détermine la courbure. Pour ce faire, nous pouvons utiliser la fonction **Part.Arc**, qui prend ces trois points en entrée et génère la géométrie de base d'un arc.



```
VC1 = FreeCAD.Vector(-10,0,0)
C1 = Part.Arc(V1,VC1,V4)
VC2 = FreeCAD.Vector(40,0,0)
C2 = Part.Arc(V2,VC2,V3)
```

Maintenant, nous avons 2 lignes (L1 et L2) et 2 arcs (C1 et C2). Nous devons les transformer en arêtes (edges) :

```
E1 = Part.Edge(L1)
E2 = Part.Edge(L2)
E3 = Part.Edge(C1)
E4 = Part.Edge(C2)
```

Les géométries de base disposent également d'une fonction **toShape()** qui fait exactement la même chose :

```
E1 = L1.toShape()
E2 = L2.toShape()
...
```

Une fois que nous avons une série d'arêtes, nous pouvons maintenant former une **polyligne**, en lui donnant une liste d'arêtes. Nous devons faire attention à l'ordre. Remarquez également les parenthèses.

```
W = Part.Wire([E1,E4,E2,E3])
```

Et nous pouvons vérifier si notre polyligne a été correctement compris, et qu'il est correctement fermé :

```
print( W.isClosed() )
```

Ce qui imprimera "True" ou "False". Pour faire une **face**, nous avons besoin de **polylignes fermées**, donc c'est toujours une bonne idée de vérifier cela avant de créer la face. Maintenant, nous pouvons créer une face, en lui donnant une seule polyligne (ou une liste de polylignes si nous voulons des trous) :

```
F = Part.Face(W)
```

Ensuite, nous l'extrudons :

```
P = F.extrude(FreeCAD.Vector(0, 0, 10))
```

Notez que P est déjà un **solide** :

```
print(P.ShapeType)
```

En effet, lorsque nous extrudons une seule face, nous obtenons toujours un solide. Ce ne serait pas le cas, par exemple, si nous avions extrudé la polyligne suivante à la place :

```
S = W.extrude(FreeCAD.Vector(0, 0, 10))  
print(S.ShapeType)
```

Ce qui, bien sûr, nous donnera une coquille creuse, les faces supérieure et inférieure manquantes.

Maintenant que nous avons notre forme définitive, nous sommes impatients de la voir à l'écran! Créons donc un objet générique et affectons-lui notre nouveau solide:

```
myObj2 =  
FreeCAD.ActiveDocument.addObject("Part::Feature", "My_Strange_Solid")  
myObj2.Shape = P  
FreeCAD.ActiveDocument.recompute()
```

Alternativement, le module Part fournit également un raccourci qui fait l'opération ci-dessus plus rapidement (mais vous ne pouvez pas choisir le nom de l'objet) :

```
Part.show(P)
```

Tout ce qui précède, et bien plus encore, est expliqué en détail sur la page [Programmation dans Part](#) en même temps que des exemples.

Lire plus d'informations :

- [Atelier Part](#)
- [Programmation dans Part](#)

Manuel : Création d'objets paramétriques

Dans le [chapitre précédent](#), nous avons exploré la manière de créer une géométrie de **Part** et de l'afficher à l'écran en l'attachant à un objet document **muet** (non paramétrique). Bien qu'efficace, cette approche devient fastidieuse lorsqu'il s'agit de modifier la forme. Chaque modification nécessite la création d'une nouvelle forme et sa réaffectation à l'objet, ce qui entraîne des flux de travail répétitifs et inefficaces.

Tout au long de ce manuel, nous avons vu comment les objets paramétriques répondent à ce problème en permettant des mises à jour dynamiques. En modifiant une seule propriété, la forme est recalculée automatiquement, ce qui élimine le besoin de mises à jour manuelles. Ce processus de recalcul permet une modélisation plus efficace et introduit une certaine adaptabilité dans les conceptions. En interne, les objets paramétriques fonctionnent de manière similaire à ce que nous avons déjà fait : ils recalculent le contenu de leur propriété **Forme** chaque fois qu'une de leurs propriétés change. Ce processus itératif est transparent et garantit que l'objet reste cohérent avec ses paramètres définis.

FreeCAD propose un système convivial de création d'objets paramétriques entièrement en Python. Ces objets sont définis à l'aide d'une classe Python, qui :

- Déclare les propriétés nécessaires pour l'objet.
- Définit le comportement en cas de modification de l'une de ces propriétés.

La structure d'un tel objet paramétrique est aussi simple que cela :

```
class myParametricObject:

    def __init__(self, obj):
        obj.Proxy = self
        obj.addProperty("App::PropertyFloat", "MyLength")
        ...

    def execute(self, obj):
        print ("Recalculating the shape...")
        print ("The value of MyLength is:")
        print (obj.MyLength)
        ...
```

Toutes les classes Python ont généralement une méthode `__init__`. Cette méthode est exécutée lorsqu'une classe est instanciée, c'est-à-dire lorsqu'un objet Python est créé à partir de la classe. Vous pouvez considérer une classe comme un «modèle» ou un plan utilisé pour créer des instances vivantes d'elle-même. Chaque instance de la classe devient un objet indépendant doté de ses propres attributs et méthodes. Dans notre méthode `__init__`, nous effectuons deux tâches cruciales :

- Enregistrer la classe elle-même dans l'attribut **Proxy** de l'objet document FreeCAD :

En assignant **self** à l'attribut **Proxy** de l'objet document FreeCAD, nous lions la logique de notre classe Python à l'objet FreeCAD. Cela signifie que l'objet document

« portera » le code de la classe Python à l'intérieur de lui-même, lui permettant de se comporter selon la logique définie dans la classe. Cette connexion permet à FreeCAD de savoir comment interagir avec l'objet paramétrique et le recalculer.

- Créer toutes les propriétés dont l'objet a besoin :

En utilisant la méthode **addProperty**, nous définissons les propriétés personnalisées requises par l'objet. Les propriétés agissent comme des paramètres ou des variables pour l'objet et peuvent être consultées, modifiées et affichées dans l'éditeur de propriétés de FreeCAD. Dans l'exemple, nous ajoutons une propriété à virgule flottante nommée **MyLength**. Cette propriété influencera ultérieurement la forme ou le comportement de l'objet.

class myParametricObject:

```
def __init__(self,obj):
    obj.Proxy = self
    obj.addProperty("App::PropertyFloat", "MyLength")
    ...

def execute(self,obj):
    print ("Recalculating the shape...")
    print ("The value of MyLength is:")
    print (obj.MyLength)
    ...
```

```
FreeCAD.ActiveDocument.addObject("Part::FeaturePython",
"dummy").supportedProperties()
```

La deuxième partie clé de notre classe est la méthode **execute**. Cette méthode est automatiquement déclenchée chaque fois que l'objet est marqué pour un nouveau calcul, ce qui se produit lorsque l'une de ses propriétés est modifiée. C'est dans la méthode **execute** que s'effectuent tous les recalculs de l'objet, garantissant que sa forme et son comportement sont mis à jour pour refléter tout changement. Dans la méthode **execute**, vous effectuez toutes les opérations nécessaires pour générer la nouvelle géométrie de votre objet. En règle générale, il s'agit de recalculer la forme en fonction des valeurs en cours de ses propriétés, puis d'affecter la forme mise à jour à l'attribut **Shape** de l'objet à l'aide d'une instruction telle que **obj.Shape = myNewShape**. La méthode **execute** prend un seul argument, **obj**, qui représente l'objet document FreeCAD associé à votre objet paramétrique. Cela vous permet de manipuler directement l'objet dans la méthode, par exemple en accédant à ses propriétés, en mettant à jour sa géométrie ou en effectuant d'autres opérations.

```
FreeCAD.ActiveDocument.addObject("Part::FeaturePython","dummy").supportedProperties()
```

- La méthode **execute** est appelée chaque fois que l'objet doit être mis à jour.
- Elle est responsable de recalculer la forme et de l'assigner à l'attribut **Shape** de l'objet.
- L'argument **obj** donne accès à l'objet document de FreeCAD, ce qui permet d'effectuer des modifications par programme.

Avec ce système, FreeCAD s'occupe du reste en s'assurant que l'objet est correctement mis à jour dans le document et affiché correctement dans l'interface graphique.

Une chose essentielle à retenir est que lorsque vous créez des objets paramétriques dans un document FreeCAD, le code Python utilisé pour les définir n'est pas sauvegardé dans le fichier. Il s'agit d'une mesure de sécurité intentionnelle. Si les fichiers FreeCAD étaient autorisés à stocker du code Python, des acteurs malveillants pourraient distribuer des fichiers contenant des scripts nuisibles susceptibles d'endommager l'ordinateur de quelqu'un. Par conséquent, lorsque vous partagez un fichier FreeCAD contenant des objets paramétriques créés avec du code Python personnalisé, le destinataire doit également avoir accès au code utilisé pour définir ces objets. Sans ce code, FreeCAD ne saura pas comment recalculer ou interagir correctement avec les objets.

La façon la plus simple d'y parvenir est d'enregistrer le code Python dans un fichier macro. Lorsque vous distribuez votre fichier FreeCAD, vous pouvez inclure la macro avec celui-ci. Vous pouvez également partager la macro sur le [dépôt des macros de FreeCAD](#), ce qui permet aux autres de la télécharger et de l'utiliser facilement. Cette approche garantit que vos objets paramétriques personnalisés restent fonctionnels sur d'autres systèmes tout en respectant les meilleures pratiques en matière de sécurité.

Ci-dessous, nous ferons un petit exercice, en construisant un objet paramétrique qui est une simple Face rectangulaire paramétrique. Des exemples plus complexes sont disponibles sur Exemple d'objet paramétrique ([parametric object example](#)) et dans le code source de FreeCAD lui-même ([FreeCAD source code](#)).

Nous allons donner à notre objet deux propriétés : Longueur et Largeur, que nous utiliserons pour construire un rectangle. Puis, puisque notre objet aura déjà une propriété de placement prédéfinie (tout objet géométrique en a une par défaut, il n'est pas nécessaire de l'ajouter nous-mêmes), nous décalerons notre rectangle à l'emplacement / orientation définis dans le Placement, de sorte que l'utilisateur pourra déplacer le rectangle n'importe où en éditant la propriété Placement.

```
class ParametricRectangle:

    def __init__(self, obj):
        obj.Proxy = self
        obj.addProperty("App::PropertyFloat", "Length")
        obj.addProperty("App::PropertyFloat", "Width")

    def execute(self, obj):
        # We need to import the FreeCAD module here too, because
        we might be running out of the Console
        # (in a macro, for example) where the FreeCAD module has
        not been imported automatically:
        import FreeCAD
        import Part

        # First we need to make sure the values of Length and
        Width are not 0
        # otherwise Part.LineSegment will complain that both
        points are equal:
        if (obj.Length == 0) or (obj.Width == 0):
```

```

        # If yes, exit this method without doing anything:
        return

    # We create 4 points for the 4 corners:
    v1 = FreeCAD.Vector(0, 0, 0)
    v2 = FreeCAD.Vector(obj.Length, 0, 0)
    v3 = FreeCAD.Vector(obj.Length,obj.Width, 0)
    v4 = FreeCAD.Vector(0, obj.Width, 0)

    # We create 4 edges:
    e1 = Part.LineSegment(v1, v2).toShape()
    e2 = Part.LineSegment(v2, v3).toShape()
    e3 = Part.LineSegment(v3, v4).toShape()
    e4 = Part.LineSegment(v4, v1).toShape()

    # We create a wire:
    w = Part.Wire([e1, e2, e3, e4])

    # We create a face:
    f = Part.Face(w)

    # All shapes have a Placement too. We give our shape the
    value of the placement
    # set by the user. This will move/rotate the face
    automatically.
    f.Placement = obj.Placement

    # All done, we can attribute our shape to the object!
    obj.Shape = f

```

Au lieu de coller le code ci-dessus dans la console Python, nous devrions plutôt l'enregistrer quelque part, alors nous pourrions le réutiliser et le modifier ultérieurement. Par exemple dans une nouvelle macro (menu **Macro -> Macros -> Créer**). Nommez-le, par exemple, "ParamRectangle". Cependant, les macros FreeCAD sont enregistrées avec une extension .FCMro, que Python ne reconnaît pas lors de l'importation. Donc, avant d'utiliser le code ci-dessus, nous devons renommer le fichier ParamRectangle.FCMacro en ParamRectangle.py. Cela se fait simplement à partir de votre explorateur de fichiers, en naviguant vers le dossier Macros indiqué dans le menu Outils -> Macros.

Une fois cela fait, nous pouvons maintenant l'utiliser dans la console Python :

```
import ParamRectangle
```

En explorant le contenu de ParamRectangle, nous pouvons vérifier qu'il contient notre Classe ParametricRectangle.

Pour créer un nouvel objet paramétrique à l'aide de notre classe ParametricRectangle, nous utiliserons le code suivant. Observez que nous utilisons **Part::FeaturePython** au lieu de **Part::Feature** que nous avons utilisé dans les chapitres précédents (la version Python permet de définir notre propre comportement paramétrique):

```
myObj = FreeCAD.ActiveDocument.addObject("Part::FeaturePython",
"Rectangle")
ParamRectangle.ParametricRectangle(myObj)
myObj.ViewObject.Proxy = 0 # This is mandatory unless we code the
ViewProvider too.
FreeCAD.ActiveDocument.recompute()
```

```
myObj = FreeCAD.ActiveDocument.addObject("Part::FeaturePython","Rectangle")
```

```
ParamRectangle.ParametricRectangle(myObj)
myObj.ViewObject.Proxy = 0 # this is mandatory unless we code the
ViewProvider too
FreeCAD.ActiveDocument.recompute()
```

- **myObj = FreeCAD.ActiveDocument.addObject("Part::FeaturePython", "Rectangle")** : crée un nouvel objet **Part::FeaturePython** nommé **Rectangle** dans le document FreeCAD actif. Cet objet est spécifiquement conçu pour un comportement paramétrique personnalisé, permettant à la logique définie par Python de gérer ses propriétés et son comportement.
- **ParamRectangle.ParametricRectangle(myObj)** : initialise l'objet en l'associant à la classe **ParametricRectangle** du module ou du script **ParamRectangle**. Cela permet de lier la logique personnalisée définie par Python à l'objet, ce qui lui permet d'agir comme un objet paramétrique.
- **myObj.ViewObject.Proxy = 0** : réinitialise l'attribut **ViewObject.Proxy** à 0, garantissant que l'objet utilise la gestion de la vue par défaut de FreeCAD. Cette étape est nécessaire à moins que vous ne définissiez un ViewProvider personnalisé pour gérer la représentation visuelle de l'objet.
- **FreeCAD.ActiveDocument.recompute()** : recalcule le document pour mettre à jour la géométrie et refléter les changements dans l'interface graphique de FreeCAD, rendant le nouvel objet entièrement visible et fonctionnel.

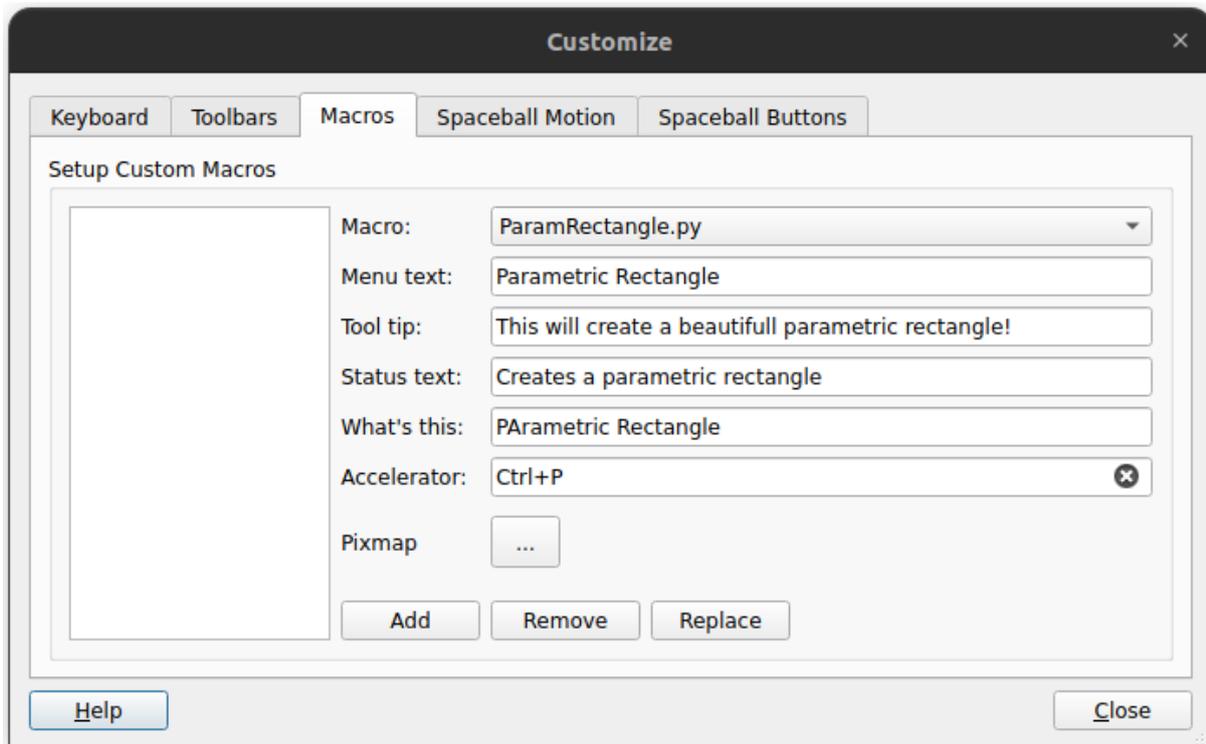
Rien ne s'affiche encore à l'écran, car les propriétés **Length** et **Width** sont à 0, ce qui déclenchera notre condition de "ne rien faire" à l'intérieur de l'exécution. Nous devons juste changer les valeurs de Longueur et Largeur, et notre objet apparaîtra magiquement et sera recalculé à la volée.

Bien sûr, il serait fastidieux de devoir taper ces 4 lignes de code Python chaque fois que nous voulons créer un nouveau rectangle paramétrique. Une façon très simple de résoudre ceci est de placer les 4 lignes ci-dessus dans notre fichier **ParamRectangle.py**, à la fin, après la fin de la classe **ParametricRectangle** (nous pouvons le faire à partir de l'éditeur Macro).

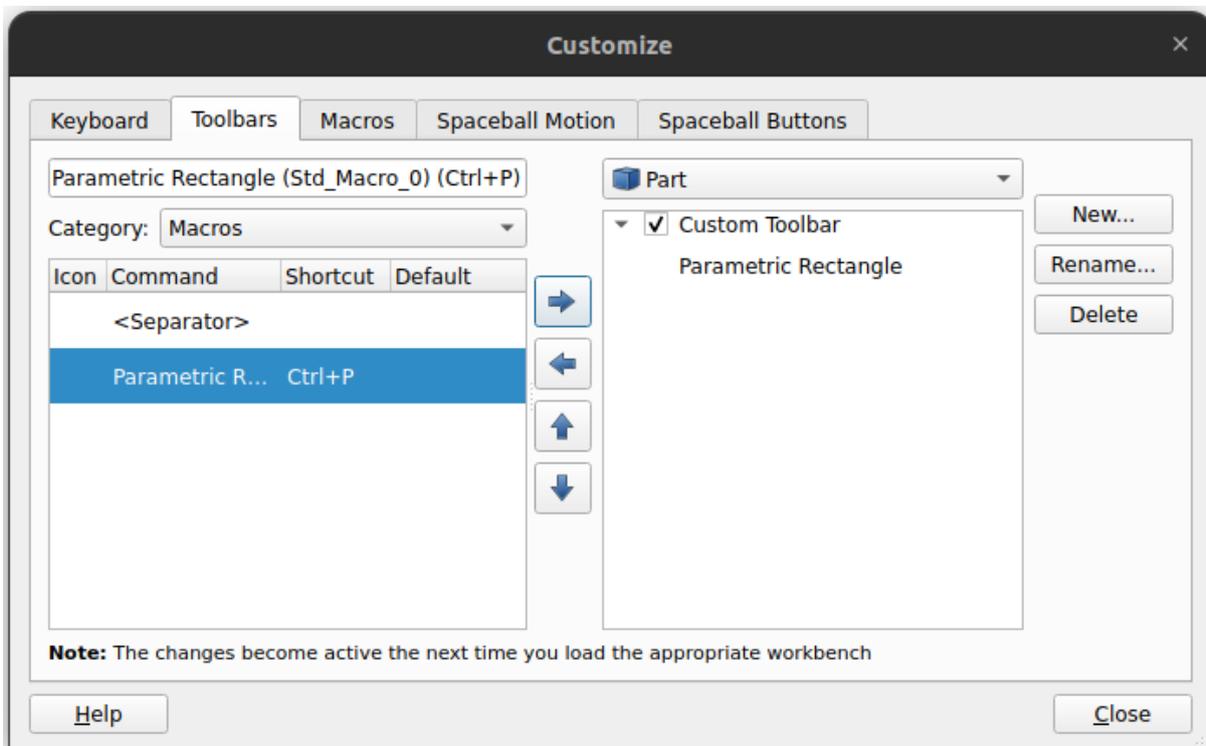
Maintenant, lorsque nous tapons import ParamRectangle, un nouveau rectangle paramétrique sera automatiquement créé. Encore mieux, nous pouvons ajouter un bouton de barre d'outils qui fera exactement cela :

- Ouvrir le menu **Outils -> Personnaliser...**

- Sous l'onglet "Macros", sélectionnez notre macro ParamRectangle.py, complétez les détails au fur et à mesure que vous le souhaitez, et appuyez sur "Ajouter" :



Sous l'onglet Barres d'outils, créez une nouvelle barre d'outils personnalisée dans l'atelier de votre choix (ou globalement), sélectionnez votre macro et ajoutez-la à la barre d'outils :



- Nous avons maintenant un nouveau bouton de barre d'outils qui, lorsqu'il sera cliqué, créera un rectangle paramétrique.

N'oubliez pas que si vous souhaitez distribuer des fichiers créés avec ce nouvel outil à d'autres personnes, celles-ci doivent également avoir la macro **ParamRectangle.py** installée sur leur ordinateur.

Lire plus d'informations

- [Le dépôt de macros FreeCAD](#)
- [Exemple d'objet paramétrique](#)
- [Plus d'exemples dans le code FreeCAD](#)

Manuel : Création d'outils d'interface

Dans les deux derniers chapitres, nous avons vu comment créer la géométrie Part ([Création et manipulation de la géométrie](#)) et créer des objets paramétriques ([Création d'objets paramétriques](#)). Une dernière pièce manque pour avoir un contrôle total sur FreeCAD: créer des outils qui interagissent avec l'utilisateur.

Dans de nombreuses situations, il n'est pas très convivial de construire un objet avec des valeurs nulles, comme nous avons fait avec le rectangle dans le chapitre précédent, puis de demander à l'utilisateur de remplir la hauteur et la largeur dans le panneau Propriétés. Cela fonctionne pour un très petit nombre d'objets, mais devient très fastidieux si vous avez beaucoup de rectangles à réaliser. Une meilleure façon serait d'être capable de donner déjà la hauteur et la largeur lors de la création du rectangle.

Python offre un outil de base permettant à l'utilisateur de saisir du texte à l'écran :

```
text = raw_input("Height of the rectangle?")
print("The entered height is ",text)
```

Cependant, cela nécessite une console Python en cours d'exécution et, lors de l'exécution de notre code à partir d'une macro, nous ne sommes pas toujours sûrs que la console Python sera activée sur la machine de l'utilisateur.

L'[interface utilisateur graphique](#), ou GUI en anglais, qui comprend les menus, les barres d'outils, la vue 3D et d'autres composants visuels de FreeCAD - est conçue pour rendre le logiciel intuitif et accessible. Elle sert de pont entre l'utilisateur et les fonctionnalités sous-jacentes de FreeCAD, permettant aux utilisateurs occasionnels comme aux experts d'interagir efficacement avec le programme.

L'interface graphique de FreeCAD est construite à l'aide de [Qt](#), une boîte à outils d'interface graphique puissante et libre qui offre un large éventail de fonctionnalités. Qt offre des éléments essentiels à la conception d'une interface, tels que des boîtes de dialogue, des boutons, des étiquettes, des champs de saisie de texte et des menus déroulants, collectivement connus sous le nom de « widgets ». Ces widgets constituent la base de l'expérience utilisateur de FreeCAD.

L'un des principaux avantages de Qt est sa compatibilité multiplateforme, qui permet à FreeCAD de fonctionner de manière transparente sur différents systèmes d'exploitation tels que Windows, macOS et Linux. En outre, la flexibilité de Qt permet aux développeurs d'étendre ou de personnaliser facilement l'interface de FreeCAD, soit en créant de nouvelles barres d'outils et de nouveaux menus, soit en construisant des modules entièrement nouveaux qui s'intègrent au logiciel. Cette adaptabilité garantit que FreeCAD reste à la fois convivial et hautement extensible.

Pour les utilisateurs intéressés par l'écriture de scripts ou le développement de nouveaux outils, l'API Python de FreeCAD permet également d'accéder à de nombreuses fonctionnalités de Qt. Cela signifie que vous pouvez non seulement automatiser des tâches, mais aussi créer des widgets ou des boîtes de dialogue personnalisés qui s'intègrent directement dans l'environnement FreeCAD.

Les outils Qt sont très faciles à utiliser à partir de Python, grâce à un module Python appelé [PySide](#). PySide est la liaison Python officielle pour la bibliothèque Qt, qui permet de créer des widgets et d'interagir avec eux de manière programmatique. Il permet aux développeurs de concevoir des interfaces, de gérer les entrées de l'utilisateur (comme la lecture de texte dans des boîtes de saisie) et de définir des actions basées sur les interactions de l'utilisateur, comme la réponse à la pression d'un bouton. En utilisant PySide, vous pouvez créer des boîtes de dialogue, des menus et des barres d'outils personnalisés directement dans FreeCAD, en étendant ses fonctionnalités d'une manière qui s'intègre harmonieusement à l'interface existante.

PySide facilite la connexion des actions de l'utilisateur à des fonctions spécifiques de votre code. Par exemple, vous pouvez configurer un bouton de sorte que lorsqu'il est pressé, il déclenche un script pour exécuter une commande ou modifier un objet dans la vue 3D. Cette capacité interactive ouvre des possibilités infinies de personnalisation des flux de travail et d'automatisation des tâches répétitives.

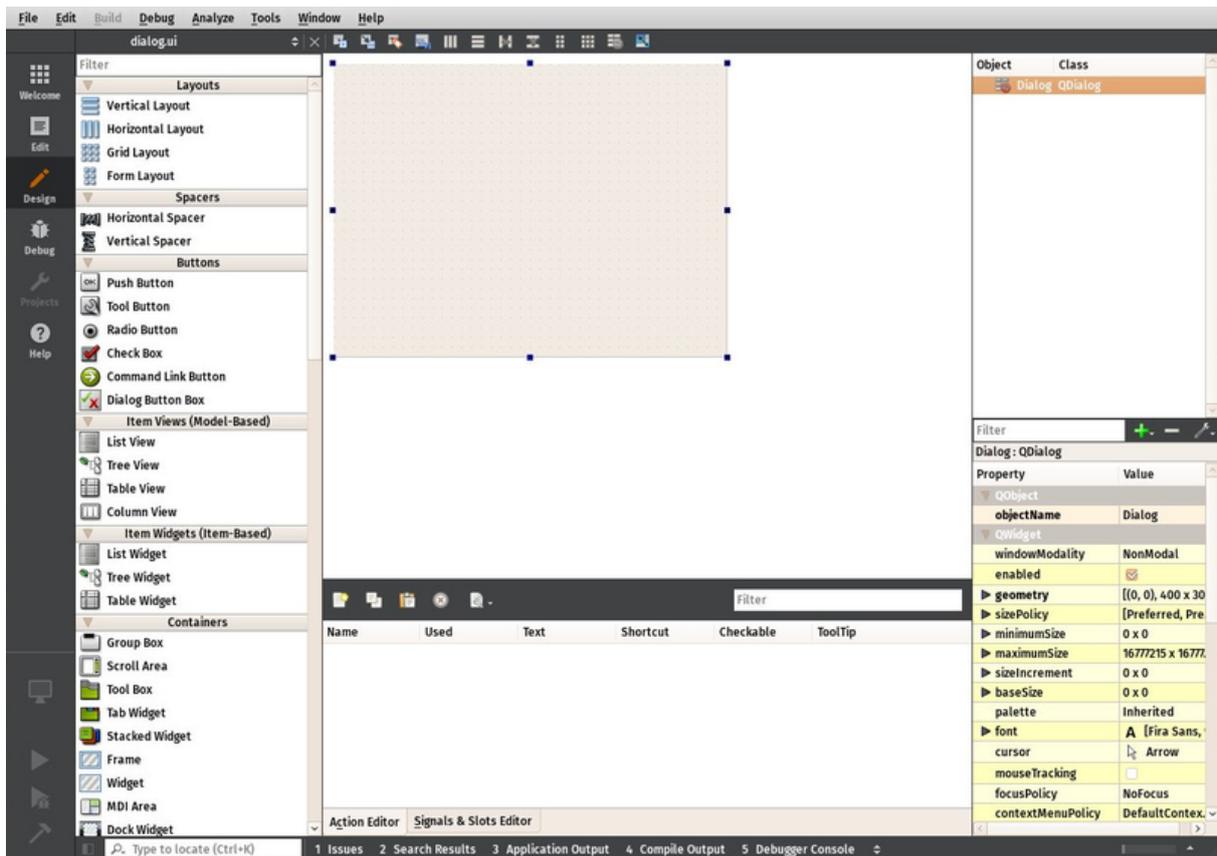
Qt fournit également un autre outil intéressant appelé [Qt Designer](#), aujourd'hui incorporé à l'intérieur d'une plus grande application appelée [Qt Creator](#). Il permet de concevoir des boîtes de dialogue et des panneaux d'interface graphiquement au lieu d'avoir à les coder manuellement. Dans ce chapitre, nous utiliserons Qt Creator pour dessiner un widget de panneau que nous utiliserons dans le **panneau des tâches** de FreeCAD. Vous devrez télécharger et installer Qt Creator à partir de sa page officielle ([official page](#)) si vous êtes sur Windows ou Mac. Sur Linux, il est généralement disponible auprès de votre application de gestionnaire de logiciels.

Dans l'exercice suivant, nous allons d'abord créer un panneau avec Qt Creator qui demande les valeurs de longueur, de largeur et de hauteur, alors nous allons créer une classe Python autour de lui, qui lira les valeurs entrées par l'utilisateur à partir du panneau, et créer une boîte avec les dimensions données. Cette classe Python sera ensuite utilisée par FreeCAD pour afficher et contrôler le panneau des tâches :

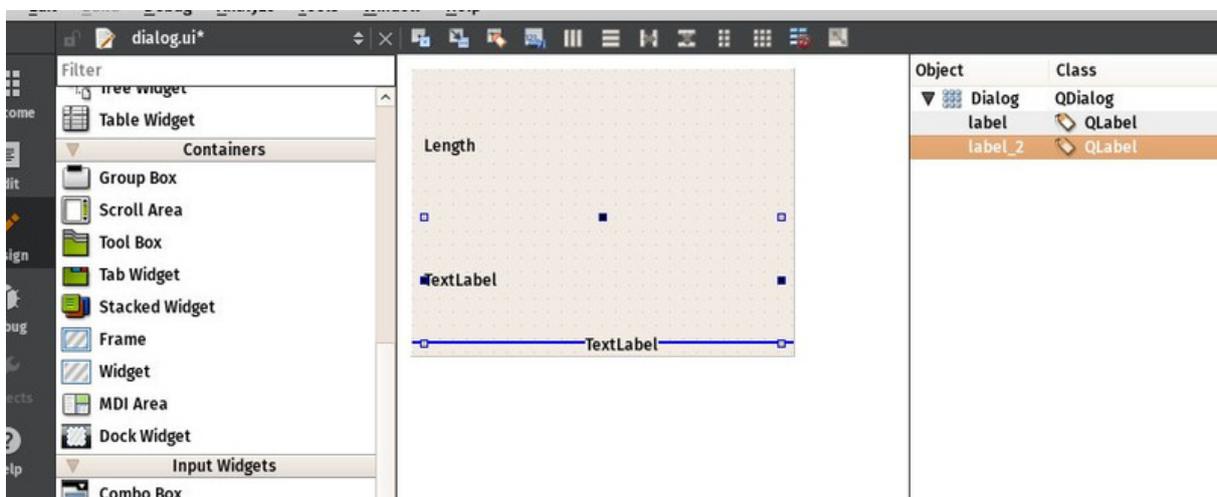


The image shows a Qt Designer dialog box with three input fields. The first field is labeled 'Length' and contains the value '0.00mm'. The second field is labeled 'Width' and contains the value '0.00mm'. The third field is labeled 'Height' and contains the value '0.00mm'. Each field has a small arrow icon on the right side, indicating it is a spin box.

Commençons par créer le widget. Démarrez Qt Creator, puis menu **Fichier** → **Nouveau fichier ou projet** → **Qt** → **Formulaire Qt Designer** → **Boîte de dialogue sans boutons**. Cliquez sur **Suivant**, donnez lui un nom de fichier pour l'enregistrer, cliquez sur **Suivant**, laissez tous les champs de projet à leur valeur par défaut ("**<none>**") et **Créer**. Le système de tâches de FreeCAD ajoutera automatiquement les boutons OK/Annuler, c'est pourquoi nous avons choisi ici une boîte de dialogue sans boutons.

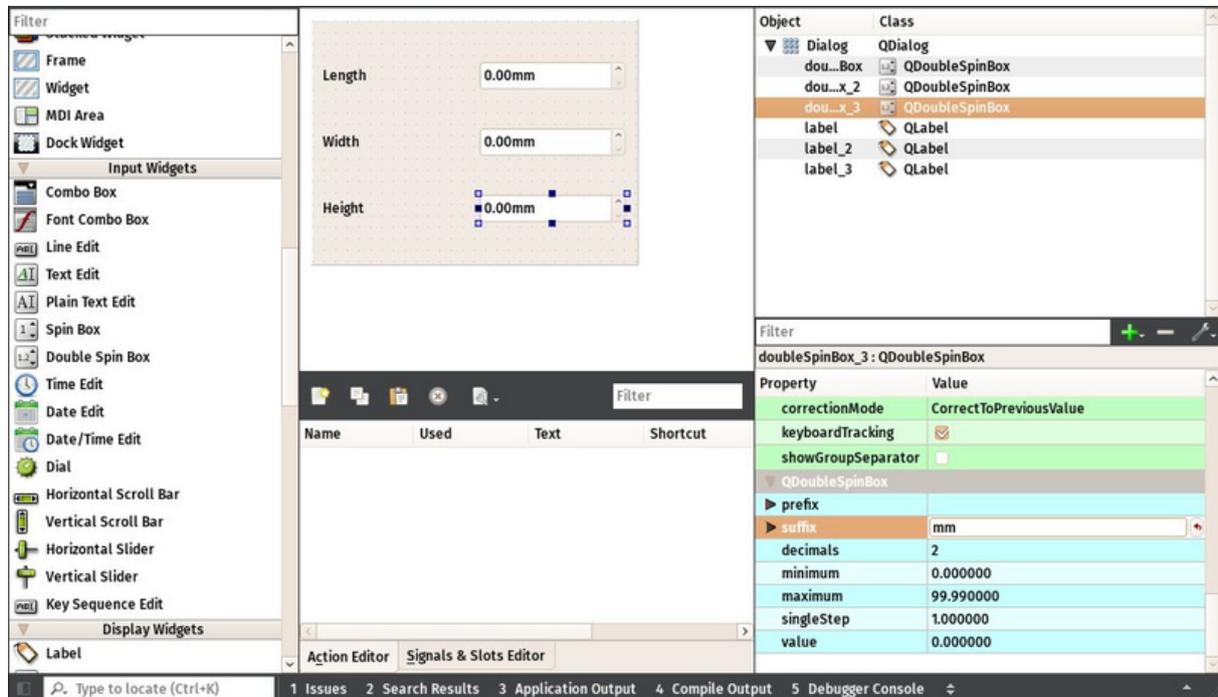


- Trouvez le **Label** (étiquette) dans la liste du panneau de gauche (dans la section affichage des widgets) et faites-le glisser sur le canevas de notre widget. Double-cliquez sur l'étiquette juste placée et modifiez son texte en **Longueur**.
- Cliquez avec le bouton droit de la souris sur la toile du widget, puis choisissez **Lay out** → **Lay out in a Grid**. Cela mettra notre widget dans une grille avec actuellement une seule cellule, occupée par notre première étiquette. Nous pouvons maintenant ajouter les éléments suivants à gauche, à droite, en haut ou en bas de notre première étiquette, et la grille s'étendra automatiquement.
- Ajoutez deux autres étiquettes en dessous de la première et modifiez leur texte en **Largeur** et **Hauteur** :

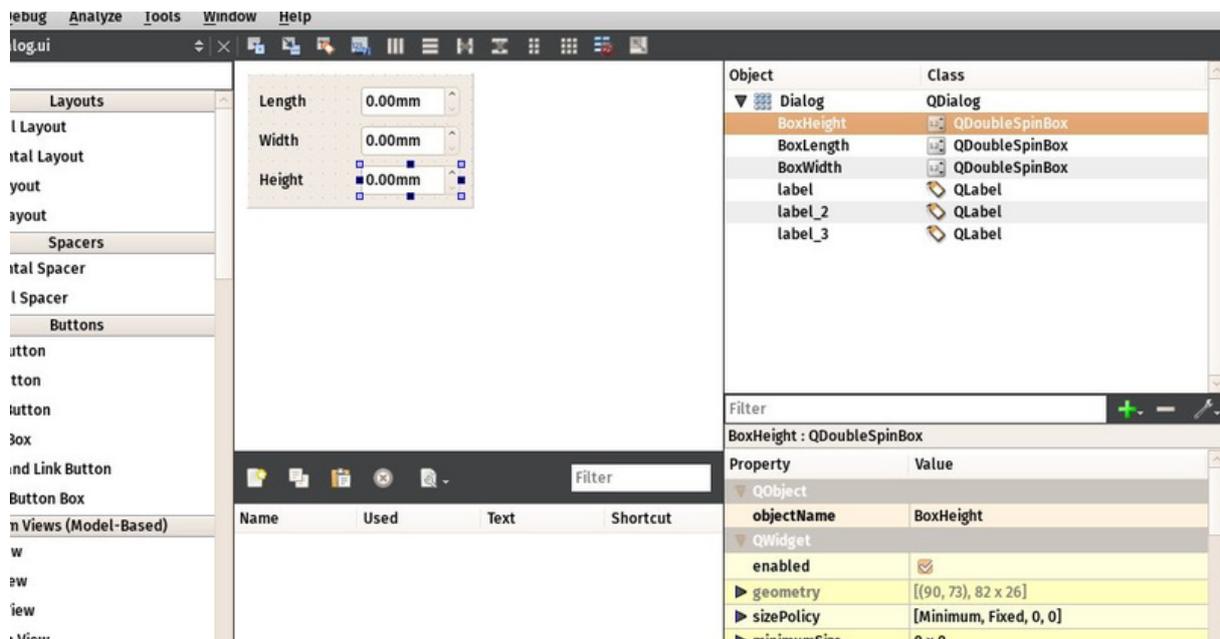


- Maintenant, placez 3 widgets **Double Spin Box** (dans la section entrée des widgets) à côté de nos étiquettes Longueur, Largeur et Hauteur. Pour chacun

d'entre eux, dans le panneau inférieur droit qui affiche tous les paramètres disponibles pour le Widget sélectionné, localisez **Suffix** et définissez leur suffixe en **mm**. FreeCAD a un widget plus avancé qui peut gérer différentes unités mais cela n'est pas disponible dans Qt Creator par défaut (néanmoins il peut être ([compilé](#))), alors, pour l'instant, nous utiliserons une Double Spin Box standard et nous ajouterons le suffixe "mm" pour nous assurer que l'utilisateur sait dans quelle unité ils fonctionnent :



- Maintenant, notre widget est terminé, il suffit de nous assurer d'une dernière chose. Étant donné que FreeCAD devra accéder à ce widget et lire les valeurs Longueur, Largeur et Hauteur, nous devons donner les noms appropriés à ces widgets, afin que nous puissions les récupérer facilement dans FreeCAD. Cliquez sur chacune des boîtes Double Spin, et dans la fenêtre supérieure droite, double-cliquez sur leur nom d'objet, et modifiez-les par quelque chose de facile à retenir, par exemple : BoxLength, BoxWidth et BoxHeight :



- Enregistrez le fichier, vous pouvez maintenant fermer Qt Creator, le reste se fera dans Python.
- Ouvrez FreeCAD et créez une nouvelle macro dans le menu **Macro** → **Macros** → **Créer**
- Collez le code suivant. Assurez-vous de modifier le chemin du fichier pour correspondre à l'endroit où vous avez enregistré le fichier .ui créé dans QtCreator :

```
import FreeCAD,FreeCADGui,Part

# CHANGE THE LINE BELOW
path_to_ui = "C:\Users\yorik\Documents\dialog.ui"

class BoxTaskPanel:
    def __init__(self):
        # this will create a Qt widget from our ui file
        self.form = FreeCADGui.PySideUic.loadUi(path_to_ui)

    def accept(self):
        length = self.form.BoxLength.value()
        width = self.form.BoxWidth.value()
        height = self.form.BoxHeight.value()
        if (length == 0) or (width == 0) or (height == 0):
            print("Error! None of the values can be 0!")
            # we bail out without doing anything
            return
        box = Part.makeBox(length,width,height)
        Part.show(box)
        FreeCADGui.Control.closeDialog()

panel = BoxTaskPanel()
FreeCADGui.Control.showDialog(panel)
```

Dans le code ci-dessus, nous avons utilisé une fonction commode **PySideUic.loadUi** à partir du module **FreeCADGui**. Cette fonction charge un fichier `.ui`, crée un widget Qt à partir de celui-ci et écrit les noms, afin que nous puissions accéder facilement au sous-espace par leurs noms (ex: `self.form.BoxLength`).

La fonction "accept" est également une commodité offerte par Qt. Lorsqu'il existe un bouton "OK" dans une boîte de dialogue (ce qui est le cas par défaut lors de l'utilisation du panneau des Tâches FreeCAD), toute fonction appelée "accept" sera automatiquement exécutée lorsque vous appuyez sur le bouton "OK". De même, vous pouvez également ajouter une fonction de "reject" qui s'exécute lorsque vous appuyez sur le bouton "Annuler". Dans notre cas, nous avons supprimé cette fonction, alors appuyer sur "Annuler" provoquera le comportement par défaut (ne rien faire et fermer la boîte de dialogue).

Si nous implémentons une des fonctions d'acceptation ou de rejet, leur comportement par défaut (ne rien faire et fermer) ne se produira plus. Nous devons donc fermer le panneau de tâches nous-mêmes. Cela se fait avec :

```
FreeCADGui.Control.closeDialog()
```

Une fois que nous avons notre `BoxTaskPanel` qui a 1- un widget appelé "self.form" et 2- si nécessaire, les fonctions `Accepter` et `Rejeter`, nous pouvons ouvrir le panneau de tâches avec lui, ce qui est fait avec ces deux dernières lignes :

```
panel = BoxTaskPanel()  
FreeCADGui.Control.showDialog(panel)
```

Notez que le widget créé par **PySideUic.loadUi** n'est pas spécifique à FreeCAD, c'est un widget Qt standard qui peut être utilisé avec d'autres outils Qt. Par exemple, nous aurions pu afficher une boîte de dialogue distincte. Essayez ceci dans la console Python de FreeCAD (en utilisant le chemin correct pour votre fichier `.ui` bien sûr) :

```
from PySide import QtGui  
w = FreeCADGui.PySideUic.loadUi("C:  
\Users\yorik\Documents\dialog.ui")  
w.show()
```

Bien sûr, nous n'avons ajouté aucun bouton "OK" ou "Annuler" à notre fenêtre de dialogue, car il a été fait pour être utilisé à partir du panneau de tâches FreeCAD, qui fournit déjà de tels boutons. Il n'y a donc aucun moyen de fermer la fenêtre de dialogue (autrement que de presser son bouton Fermer la fenêtre). Mais la fonction

`show ()` crée une fenêtre de dialogue non modale, ce qui signifie qu'elle ne bloque pas le reste de l'interface. Ainsi, alors que notre fenêtre de dialogue est toujours ouverte, nous pouvons lire les valeurs des champs :

```
w.BoxHeight.value()
```

Ceci est très utile pour les tests.

Enfin, n'oubliez pas qu'il existe beaucoup plus de documentation sur l'utilisation des widgets Qt dans le Wiki FreeCAD, dans la section [Python Scripting](#), qui contient un didacticiel de création de dialogue ([dialog creation tutorial](#)), un tutoriel spécial en 3 parties de PySide ([PySide tutorial](#)) qui couvre le sujet en profondeur.

Liens pertinents

- [Qt Creator Documentation](#)
- [Installing Qt Creator](#)
- [Documentation pour utilisateurs avancés](#)
- [Création de dialogue](#)
- [PySide](#)
- [Documentation PySide](#)

Manuel : La communauté

Aucun manuel traitant des logiciels libres ne serait complet sans un chapitre sur la communauté. Comme la grande majorité des projets de logiciels libres, FreeCAD est créé par une communauté et maintenu par cette communauté. Au lieu d'une société opaque, inconnue, impersonnelle et inaccessible qui se trouve le plus souvent derrière les logiciels commerciaux, les communautés de logiciels libres sont des espaces ouverts, où vous êtes le bienvenu en tant qu'utilisateur, et où vous pouvez obtenir des réponses très rapidement, et même avoir votre mot à dire dans le développement du logiciel lui-même. Vous êtes également les bienvenus pour aider, il y a des tâches pour tout le monde.

La communauté est un groupe croissant et éclectique de toutes sortes de personnes unies par leur passion pour FreeCAD. Tous travaillent sur FreeCAD volontairement, pendant leur temps libre (bien que parfois des entreprises ou des individus se rassemblent pour payer quelques heures de programmation à un développeur pour mettre en œuvre une fonction spécifique). Certains sont des programmeurs professionnels, certains sont des utilisateurs de FreeCAD de longue date (certains d'entre eux sont de vrais gourous FreeCAD, qui connaissent presque tout, et beaucoup d'entre eux finissent par connaître beaucoup de programmation FreeCAD) et beaucoup sont de nouveaux utilisateurs de FreeCAD. Il n'y a rien à faire pour faire partie de la communauté. Utilisez simplement FreeCAD !

Le principal endroit où la communauté se rencontre et discute est le [forum FreeCAD](#). Tout ce que vous devez faire pour participer aux discussions est d'enregistrer un compte sur le forum (votre premier post devra être approuvé par un modérateur avant de poster plus, afin d'éviter tout spam). Le forum est un excellent endroit pour poser des questions lorsque vous êtes nouveau sur FreeCAD. Si vous avez posé une bonne question (assurez-vous de lire les [règles du forum](#) car elles contiennent des informations utiles pour transformer votre question en une bonne question), vous recevrez habituellement plusieurs réponses dans la même heure. Si vous pensez que quelqu'un a déjà posé votre question, n'oubliez pas de rechercher, votre réponse pourrait déjà être là.

Le forum est également un endroit idéal pour montrer ce que vous avez réalisé avec FreeCAD, pour aider les nouveaux venus lorsque vous êtes plus expérimentés, et pour suivre et donner votre avis dans les discussions plus techniques sur le développement. Tout le [développement de FreeCAD](#) est discuté sur le forum, et tout le monde est libre de lire ou de participer.

Il existe également des communautés FreeCAD qui se forment en dehors du forum FreeCAD, par exemple sur [Facebook](#).

Si vous devenez aussi enthousiaste à propos de FreeCAD que nous le sommes, vous voudrez peut-être aider le projet. Cela peut se faire de différentes façons, et il existe des tâches pour tous, les programmeurs et les non-programmeurs, par exemple :

- **Aidez à diffuser le mot** : beaucoup de gens tireraient un énorme bénéfice de l'utilisation d'un modèleur 3D gratuit et open source comme FreeCAD, mais ne savent tout simplement pas qu'il existe. Publier le travail que vous faites avec FreeCAD, en parler sur les réseaux sociaux, etc... aide ces personnes à découvrir FreeCAD.

- **Aidez les nouveaux arrivants** : la grande majorité des discussions sur le forum sont des questions posées par de nouveaux utilisateurs. Vous pourriez avoir de bonnes réponses à leur donner.
- **Aidez à signaler les bugs** : la stabilité de FreeCAD provient en grande partie de la correction des bugs. Comme il n'est pas possible pour les développeurs de FreeCAD de tester tous les cas d'utilisation possibles, il est important que les utilisateurs signalent les problèmes lorsqu'ils les détectent. Assurez-vous de lire les [directives](#) si vous pensez avoir trouvé un bug, puis écrivez un rapport sur le [traqueur de bogues](#).
- **Aidez-nous à écrire de la documentation** : la [documentation wiki de FreeCAD](#) est également rédigée par les membres de la communauté. Certaines de ses sections sont encore incomplètes, ou leurs informations sont incorrectes ou obsolètes. Vous pouvez peut-être aider à corriger cela. Pour pouvoir travailler sur le wiki, vous devrez vous familiariser avec l'[édition de wiki](#) et [demander la permission](#) sur le forum pour éditer le wiki de FreeCAD .
- **Aidez-nous à traduire FreeCAD** : la traduction de FreeCAD est faite en ligne par les membres de la communauté, sur [crowdin](#). Si vous n'y voyez pas votre langue, demandez à l'un des administrateurs de l'ajouter.
- **Aidez-nous à traduire la documentation wiki** : chaque page du wiki est traduisible, et nécessite très peu de connaissances de la syntaxe du wiki. Aider à la traduction est aussi un excellent moyen d'apprendre FreeCAD.
- **Écrivez des scripts et des macros** : FreeCAD a une liste croissante des [Macros](#). Si vous avez écrit une fonctionnalité intéressante, pensez à la partager ici.
- **Programmation** : Pour cela, vous devez savoir comment programmer en Python ou C++, et avoir une bonne connaissance de FreeCAD lui-même.

Le code source de FreeCAD se trouve sur le compte [Github](#) du projet FreeCAD. Tout le monde peut télécharger, utiliser et modifier le code. Vous pouvez publier vos modifications (sur Github ou tout autre service d'hébergement Git). Si vous avez fait des modifications intéressantes que vous souhaitez voir incluses dans le code source de FreeCAD, vous devez demander à la communauté de les inclure. Cela peut être fait en utilisant le mécanisme de demandes de pull de Github, mais la meilleure façon est de discuter de ce que vous avez l'intention de faire d'abord sur le forum, puis de poster une demande officielle dans la section [Pull requests](#) du forum lorsque votre code est prêt. Cela vous évite de travailler sur quelque chose sur lequel quelqu'un d'autre travaille déjà aussi, et garantit que les autres sont d'accord avec la façon dont vous le faites, de sorte qu'il n'y a pas de risque que votre travail soit refusé pour une raison que vous n'avez pas prévue.

Nous espérons que nous avons réussi à vous donner un bon aperçu de FreeCAD dans ce manuel, et vous êtes déjà notre plus récent membre de la communauté. Bienvenue !

Lire plus d'informations

- [Le forum FreeCAD](#)
- [Le code source de FreeCAD](#)
- [La communauté Facebook FreeCAD](#)
- [Le wiki de documentation FreeCAD](#)
- [Traduction de FreeCAD sur crowdin](#)
- [Le tracker des bugs FreeCAD](#)