

FreeCAD [使用方法]

利用 Python 进行实体建模

Foreword by Yorik van Havre, FreeCAD developer

布拉德·科莱特 (Brad Collette)
丹尼尔·法尔克 (Daniel Falck)

FreeCAD[使用方法]

FreeCAD[How-to]

Copyright©2012PacktPublishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotation embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: September 2012
Production Reference: 1070912
Published by Packt Publishing Ltd., Livery Place
35 Livery Street, Birmingham B3 2PB, UK. ISBN 978-1-84951-886-4

感谢

作者

Brad Collette

Daniel Falck

审稿人

Yorik van Havre

Adrian Przekwas

组稿编辑

Mary Jasmine Nadar

内容编辑

Meeta Rajani

Priyanka Shah

技术编辑

Joyslita D'Souza

项目协调员

Michelle Quadros

校对员

Aaron Nash

索引

Hemangini Bari

制作协调员

Prachali Bhiwandkar

封面工作

Prachali Bhiwandkar

封面图片

Manu Joseph

汉化翻译

FreeCAD 翻译团队

QQ2416113943

前言

FreeCAD 项目由两名德国工程师 JurgenRiegel 和 WernerMayer 于 2002 年左右开始启动的。这个项目雄心勃勃。计算机辅助设计（CAD）世界曾经是，现在仍然还是由一些高级商业应用程序所主导，这些应用程序背后有大量的开发人员。

使得创建开源专业级 CAD 应用程序成为可能的事件是 OpenCasCade 库的开源，OpenCasCade 库是一个强大的 3D 建模内核，它是 FreeCAD 的核心组件。在此之后，有很多聪明的想法出现，关于如何发展、帮助、保护现代 CAD 应用程序。尽管它仍然无法与商业对手竞争，但它开始对小型 CAD 项目非常有用。

我在 2006 年左右发现了这个项目，关注了一段时间，然后开始为其编写一些脚本，并在 2008 年正式加入开发团队。现在，围绕该项目的开发人员，用户和爱好者社区正在以前所未有的速度增长。这有助于项目达到更高的开发速度和质量水平，并且很高兴看到 FreeCAD 进入专业领域的第一步。

我认识 DanFalck 很长一段时间了，从旧的邮件列表开发，当时我们都在拼命寻找在 Linux 平台上进行 CAD 工作的方法。。Dan 是 Linux，CAD 和 CNC 世界的知名人物，并且在 HeeksCAD 上做了很多工作，HeeksCAD 是 FreeCAD 的近亲，也是基于 OpenCasCade 内核的。在此过程中，Dan 也越来越多地参与 FreeCAD，为 FreeCAD 项目做出了许多贡献，很多想法还初步的设计阶段。不久之后，来自 HeeksCAD 和 CNC 社区还有著名的 BradCollette（在开源 CAD/CAM 世界中被称为 Sliptonic）。这是开源 CAD 世界的两个重量级人物，没有人会比他们写 FreeCAD 的书，写的更好了。

Yorik van Havre
FreeCAD 开发人员

关于作者

布拉德·科莱特 (BradCollette) 曾经为一家大公司设计软件，但他不喜欢被记住这一点。如今，他是一位企业家，业余爱好者，万事通和绅士的农夫。他从事一个旨在养育两个黑客儿子的多年项目。他为众多开源项目做出了贡献，并且是密苏里州中部最好的黑客空间 ColumbiaGadgetWorks 的组织成员。

丹尼尔·法尔克 (DanielFalck) 一直对事物的运作方式很感兴趣。作为一个男孩，他学会了弹吉他，并决定学习如何制作吉他。这后来发展到学习如何使工具，以帮助建立吉他。他今天仍然表现出这种行为，因为他试图学习如何构建开源的 CAD/CAM 软件，帮助他构建其他工具，以及吉他部件。

过去，他曾在 GibsonGuitarCorporation 工作，担任工具设计师和原型机车间主管，在那里他学习了 CAD 软件。目前，他为 ChrisKingPrecisionComponents 做同样的事情，经营原型机加工车间，设计工具，量具，夹具，机器零件和软件。

多年来，他对开放源代码软件贡献很多，并参与了多个开放源代码 CAD/CAM 项目。Linuxcnc 的灵感激发了他对 Linux 和开源的兴趣。他还参与了 APTOS, HeeksCNC 项目以及现在的 FreeCAD。

我要感谢 FreeCAD 的开发人员和用户，他们共同构建了一个强大的开源 CAD 程序。感谢 JurgenRiegel, WernerMayer 和 YorikvanHavre 创建了如此出色的应用程序。

布拉德和我过去合作过，为《数字机械师》杂志撰写了一篇关于 HeeksCNC 的文章。我要感谢布拉德再次与我合作。

关于审稿人

YorikvanHavre 是一位架构师，3D 艺术家和开源软件爱好者。他在巴西生活和工作，是 FreeCAD 项目的开发人员之一，并且是 Blender 等其他几个开源项目社区的活跃成员。他审查了 Packt 的其他几本书，包括 Blender3D 建筑，建筑物和风光以及 Blender2.5 照明和渲染。他定期在其网站 <http://yorik.uncreated.net> 上发布有关他的工作，体系结构，3D 和开源软件的文章。

AdrianPrzekwas 是机械工程学理学硕士。他是一家波兰中型公司的设计工程师。该公司生产钢结构并为采矿业开发解决方案。他是开源软件和汽车技术爱好者。

前言

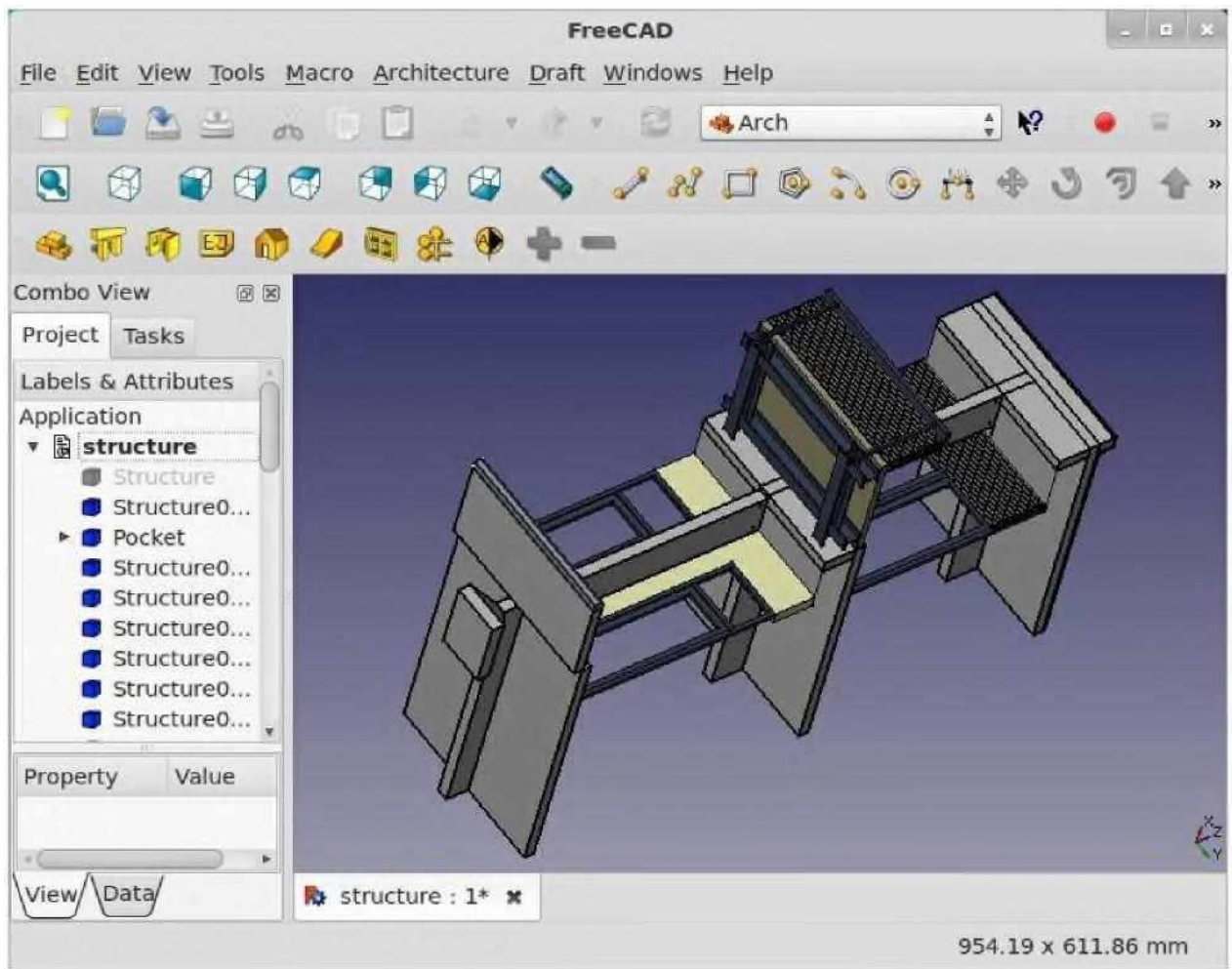
FreeCAD 是面向工程界的通用建模工具。与为动画师和美工设计的其他建模工具（如 Blender 或 Maya）不同，FreeCAD 高度重视基于参数的，基于特征的设计。

FreeCAD 最初是为机械工程和产品设计而设计的，现在正在开发中以添加适用于各种工程学科的功能。

FreeCAD 的基础技术已经有了很多远见和规划。结果是该工具功能强大，易于使用且易于扩展。

功能强大的脚本语言 Python 的广泛使用在一定程度上负责 FreeCAD 的灵活性和快速开发。最终用户也可以通过几种不同方式使用此功能。从自动执行简单任务的录制宏到直接创建和操纵几何图形，几乎做任何事情都是可能的。

实际上，可以通过内置的 Python 解释器访问 FreeCAD 应用程序的各个方面。甚至可以使用 Python 代码访问和扩展用户界面，允许用户创建新的对话框屏幕和扩展核心功能的整个模块。Arch 模块就是一个例子。Yori kvan Havre 正在开发这个工作台，以提供架构设计功能。它完全是用 Python 编写的。



它的开放架构、可扩展的设计、快速的开发速度和热情的社区弥补了它的许多限制。它在所有三个主要平台上均能很好地运行，并已被翻译成多种语言。加上它的免费价格，意味着现在任何人都可以尝试免费 FreeCAD。

本书涵盖的内容

获取和安装 FreeCAD（必知），将逐步介绍在计算机上运行 FreeCAD 的基础知识。最新最好的，还是稳定容易的？Mac，Windows 或 Linux？我们将向您展示您的选择，并讨论权衡问题。

了解 FreeCAD 界面（必须知道），可以广泛地了解 FreeCAD 的组织。该使用说明书将引导您完成用户界面，并轻松自定义它以供自己使用。

零件工作台中的 CSG 建模（必知），向您展示了如何从简单的形状开始并将它们组合以构建复杂的对象。CSG 建模是 FreeCAD 实现的几种重要技术之一，并且是建模工具套件中不可或缺的一部分。

录制和编辑一个宏（应该知道），通过录制和回放一系列经常使用的命令，使工作变得更容易。创建您自己的自定义命令或自动化一个例行任务。

使用 Draft 工作台（必须知道），对简单零件进行建模，介绍了 FreeCAD 中可用的另一种建模技术。许多设计，甚至是复杂的 3D 对象，都从 2D 工程图开始。草稿工作台提供了用于处理圆，圆弧，直线和其他 2D 元素的工具。

旋转和拉伸以创建零件（应该知道），将显示如何将 2D 工程图转换为 3D 模型。旋转和拉伸是每个 FreeCAD 用户都应该知道的另外两个工具。

使用 Python 创建 3D 实体（成为专家），展示了如何使用 Python 创建自定义 3D 对象。Python 是另一个 FreeCAD 工具箱中的强大工具。

创建自定义对话框以自动执行任务（成为专家），将向您展示如何通过创建自定义用户界面使 Python 脚本更易于使用。您可以创建一个漂亮的对话框。

使用约束进行建模（必须知道），涵盖了 FreeCAD 用户可用的最强大的技术之一。需要在不破坏其他部分的情况下更改模型的一个方面吗？带有约束的设计意味着在设计中构建灵活性。现在您可以轻松调整设计并获得可预测的结果。

使用外部约束（应该知道），将通过使用 FreeCAD 中的一些高级草图绘制工具超越简单的草图。您可以在现有对象的面上建模其他特征，并将草图附加到现有对象。

使用 Python 添加或修改约束（成为专家），使 Python 的功能可用于基于约束的建模。您可以使用 Python 脚本更深入地了解草图约束的内部工作原理。

创建一个部分的绘图（应该知道），教你如何通过创建一个可以打印的绘图来向 2D 世界展示你的 3D 设计。

导出其他应用程序的 DXF 文件（应该知道），将向您展示如何使用行业标准的 DXF 文件将您的设计导出到其他应用程序。当你需要在 FreeCAD 之外做一些事情的时候，导出是非常宝贵的。

导入数据（应该知道），将为您提供如何与其他应用程序中的设计进行交互的信息。

这本书需要什么

FreeCAD 是一个独立的设计工具。一旦安装，就不需要额外的东西来开始设计复杂的模型。如果要编译 FreeCAD 的最新版本，Internet 上提供了开放源代码工具，包括下载源代码所需的 Git 版本控制系统。

这本书是给谁的

本书是专为很少或没有经验的 CAD 软件新用户编写的，还将帮助其他 CAD 应用程序的用户快速熟悉 FreeCAD 的独特概念。

约定

在本书中，您会发现许多可以区分不同类型信息的文本样式。以下是这些样式的一些示例，并解释了其含义。

文本中的代码字如下所示：“注意它如何为您提供有关 `makeBox` 类的有用提示。

代码块设置如下：

```
importFreeCAD
importPartApp.ActiveDocument.addObject("Part::Box","Box")
FreeCAD.getDocument("Unnamed").getObject("Box").Wid
```

当我们希望引起您对代码块特定部分的注意时，相关的行或项目将以粗体显示：

```
QtCore.QObject.connect\
(self.okButton,QtCore.SIGNAL("pressed()"),self.box
```

新术语和关键词以粗体显示。您在屏幕上看到的单词，例如在菜单或对话框中，将以如下形式显示在文本中：“选择导线，然后单击升级按钮以将其转换为面。”

警告或重要提示会出现在这样的框中。

提示和技巧如下所示。

读者反馈

我们欢迎读者的反馈。让我们知道你对这本书的看法，你喜欢什么或可能不喜欢什么。读者的反馈对于我们开发出你真正能最大限度利用的书籍是很重要的。

要向我们发送一般性反馈，只需将电子邮件发送至<feedback@packtpub.com>，并通过邮件主题提及书名。

如果您需要一本书，并且希望我们出版，请通过 WWW.packtpub.com 上的 **SUGGESTATITLE** 表格发送给我们说明，或发送电子邮件至 suggest@packtpub.com。

如果有一个你擅长的主题，并且你对写书或为书做贡献感兴趣，请查看我们的作者指南 www.packtpub.com/authors。

客户支持

既然您是 Packtbooks 骄傲的拥有者，我们可以通过很多方法来帮助您从购买中获得最大收益。

下载示例代码

您可以在 <http://www.PacktPub.com> 上从帐户下载所有 Packt 书籍的示例代码文件。

如果您在其他地方购买了此书，则可以访问

<http://www.PacktPub.com/support> 并注册以将文件直接通过电子邮件发送给您。

勘误

尽管我们已尽一切努力确保内容的准确性，但还是会发生错误。如果您发现我们的其中一本书中有错误—可能是文本或代码中的错误—请将此报告给我们，我们将不胜感激。这样，您可以使其他读者免于沮丧，并帮助我们改进本书的后续版本。如果您发现任何勘误，请访问，

<http://www.packtpub.com/support> 进行报告，选择您的书籍，单击勘误提交表单链接，然后输入勘误的详细信息。验证勘误后，您的提交将被接受，并且勘误将上传到我们的网站上，或添加到该标题的“勘误”部分下的任何现有勘误列表中。通过从 <http://www.packtpub.com/support> 选择标题，可以查看任何现有的勘误。

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at <copyright@packtpub.com> with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

问题

您可以通过<questions@packtpub.com>与我们联系。如果您对本书的任何方面都有疑问，我们将尽力解决。

汉化团队留言

您可以通过<xingwozhonghua@126.com>与我们联系。FreeCAD 相关资料匮乏，目前最近版本已经出到 19，支持 python3，本书中有疏漏的地方请联系我们，后面我们也会出一些整理的资料，以便大家推广学习。

第 1 章 FreeCAD[使用方法]

欢迎使用 FreeCAD[操作方法]。本书将介绍此开源应用程序的组织 and 多功能性。本书中使用说明将介绍 FreeCAD 中使用的大多数概念，以便在一个领域中获得的知识将帮助您更有效地探索其他领域。

FreeCAD 是面向工程界的通用建模工具。与为动画师和艺术家设计的其他建模工具（例如 Blender 或 Maya）不同，FreeCAD 着重强调了三维实体基于参数，基于特征的设计。

FreeCAD 的基础技术已经有了很多远见和规划。结果是该工具功能强大，易于使用，并且易于使用功能强大的脚本语言 Python 进行扩展。

本书的第一部分将帮助您熟悉 FreeCAD 界面和原理。我们将研究几种用于对象建模的特定技术。后面的使用说明将更加着重于如何使用 Python 自动化和扩展 FreeCAD。

获取和安装 FreeCAD（必须知道）

本食谱将描述可用于在三个主要平台（Windows，Mac 和 Linux）上安装 FreeCAD 的选项。它将简要介绍安装稳定发行版的步骤。

做好准备

FreeCAD 仍在大量开发中。定期提供经过测试和调试的稳定版本。这些版本将缺少最新功能，并且可能与本书中描述的许多使用方法不匹配。

FreeCAD 站点上提供了在所有三个平台上安装的详细说明，以及删除和故障排除信息。

怎么做...

1. 对于大多数临时用户，稳定版本的二进制文件是开始学习 FreeCAD 的最佳位置。
2. 首先访问 FreeCAD 网站，网址为 <http://free-cad.sourceforge.net/>。
3. FreeCAD 网站的主页上有一个面板，其中包含当前稳定版本的下载链接。



●对于 Windows，稳定版本以 32 位 MicrosoftInstaller 应用程序(.msi 文件). 的形式提供。将文件下载到您的计算机，然后双击以运行安装程序。

●Linux (Ubuntu) 链接会将您定向到个人软件包存档(PPA). 此处的软件包适用于 Ubuntu，但可以在 Debian 衍生的任何最新发行版上使用。PPA 提供 32 位和 64 位版本。将 PPA 添加到您的存储库管理器，然后像安装其他任何应用程序一样安装它。FreeCAD 在主 Ubuntu 存储库中也可用，但是此版本已过时，应避免使用。

●MacOSX(Lion)支持 64 位安装包。安装程序以磁盘映像的形式分发。下载磁盘映像文件。安装它并运行 InstallFreeCAD 软件包。在过去，MacOSX 的打包是由一个小团队支持的，并且打包并不总是最新的。鼓励 OSX 用户考虑从源代码构建。

还有更多...

FreeCAD 的发展速度非常快。新功能和改进几乎每天都在源代码的不稳定和开发分支中出现。在研究了基础知识之后，您可能希望探索稳定版中尚不可用的功能。让我们看一下其他一些安装选项。

每日构建

如果您正在使用 Linux，并且愿意承担损坏特性和偶尔崩溃的风险，那么您可以安装日常构建之一。它们每天由构建脚本自动生成，因此没有经过正式的测试。每日构建包含了开发人员提交给源代码的最新特性和 bug 修复。

每天的构建对于 Windows 是不可用的，但是一个不稳定的二进制安装程序是定期构建的。从此下载

<https://sourceforge.net/projects/free-cad/files/FreeCAD%20Windows/>。

从源代码构建

从源代码构建 FreeCAD 的详细说明超出了指南书的范围，但是对主要步骤的概述可能会有所帮助：

1. **满足先决条件**: 您将需要一个编译器和所有依赖库。FreeCAD 有一长串的依赖项，把它们都放在合适的位置可能是件苦差事。这在 Linux 计算机上要容易得多，因为大多数或所有依赖项都可以从软件管理器下载。

对于 Windows, FreeCAD 站点还提供了 libpack 下载，以简化依赖项需求。

2. **获取源代码**: FreeCAD 源代码保存在 git://freecad.git.sourceforge.net/gitroot/free-cad/free-cad 的 Git 存储库中。

3. **编译**: 编译的具体步骤因平台而异，但是一旦源代码和依赖项就位，通常就很简单了。在 FreeCAD 网站上研究你的平台页面。

第三方库构建

FreeCAD 站点还维护一个备选构建的列表。这些是由 FreeCAD 项目之外的人员和组织创建和维护的二进制安装。它们可以包含自定义特性，也可以针对特定的操作系统进行优化。在撰写本文时，已经有了针对 OpenSuse、Fedora、Slackware、ArchLinux、Gentoo 等的第三方版本。

了解 FreeCAD 界面（必须知道）

FreeCAD 旨在灵活，可扩展且面向任务。

核心功能分为工作台[workbenches]。每个工作台都会收集用于特定任务的最常用工具。本使用说明将指导您完成 FreeCAD 用户界面的介绍，并介绍所有工作台通用的一些概念。。

做好准备

启动 FreeCAD，并使用屏幕快照和界面主要部分的名称来定位自己。本书中的图像和示例来自 0.13 开发快照。



默认视图为开始页面[Startpage]-嵌入式浏览器，显示有趣的视频链接，示例和其他新闻。

用户界面分为三个主要区域：

1. 3D 区域使您可以在空间中查看和浏览设计。
2. 组合视图显示项目的层次结构视图或当前任务的状态。
3. 工具栏和工作台切换器使您可以使用各种工具。

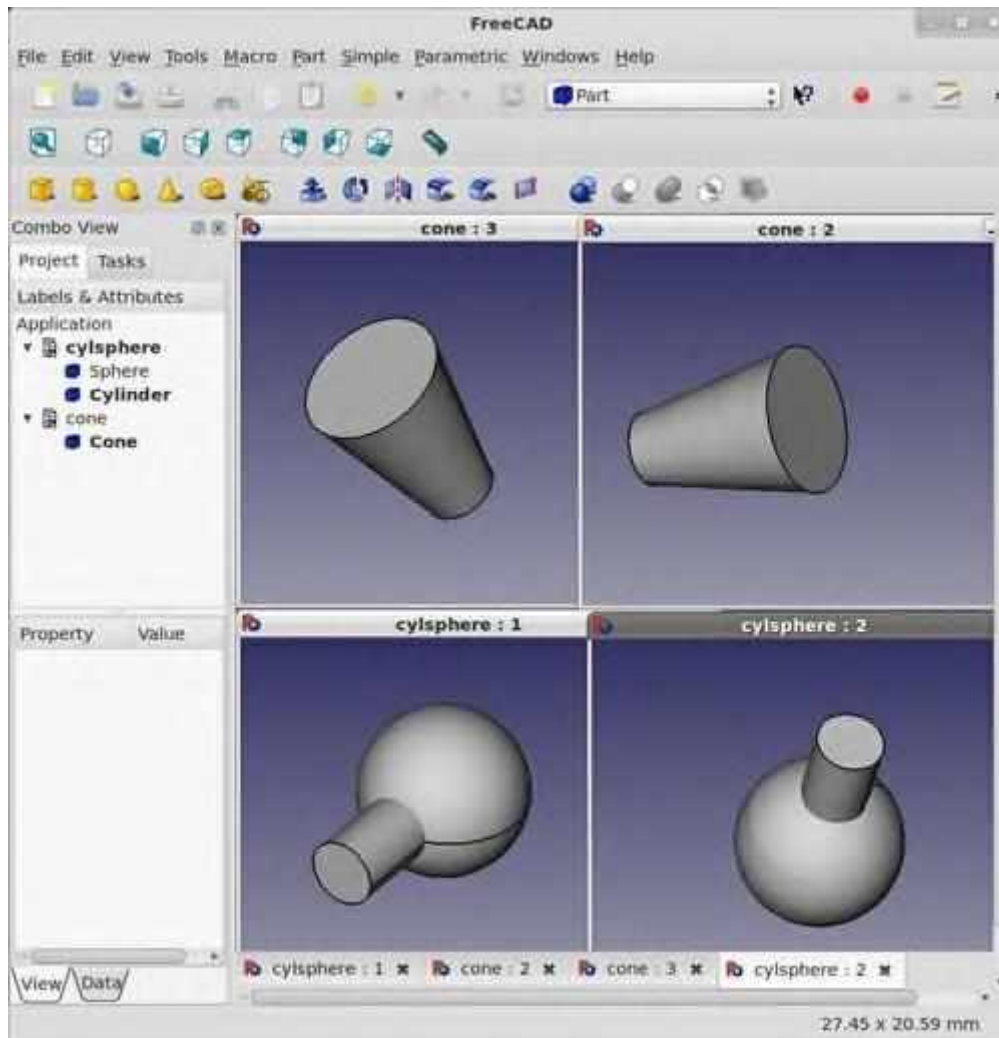
怎么做...

1. 单击菜单**文件**[File] | **新建**[new] 创建一个新文档。3D 视图将切换到显示空白空间。在默认方向上，摄像机位置向下俯视 X 和 Y 轴定义的平面。Z 轴直接指向摄像机。角落中的坐标指示器可帮助您将自己定位在 3D 空间中。



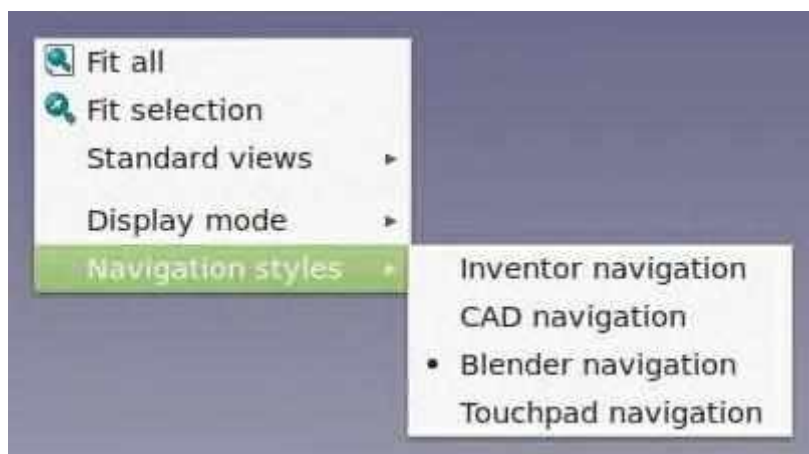
2. 单击工作台切换器，然后选择**零件**[part]工作台。然后单击**创建方框实心**[Createboxsolid]图标以将多维数据集插入到项目中。

3. 您可以同时打开多个项目。您还可以打开同一项目的多个视图。通过单击菜单**视图**[View] | **创建新视图**[Createnewview]来打开项目的第二个视图。您可以使用 3D 窗口下面的选项卡切换视图，也可以将视图平铺为同时具有多个透视图。在下面的截图中，有两个项目是开放的，每个项目都有两个视图：



4. 您可以通过**平移**[panning]，**旋转**[rotating]和**缩放**[zooming]来更改项目的视图。通过单击对象来选择对象以更改其属性。用于执行此操作的特定鼠标和键盘命令称为导航样式。FreeCAD 默认支持四种样式: Inventor, CAD, Blender, 和 Touchpad.

您可以通过右键单击 3D 画布来更改当前视图中使用的样式。



通过**编辑**[Edit] | **首选项**[Preferences] 菜单更改样式将更改当前打开的所有视图的样式，并将选择设置为新视图的默认选项。默认的样式，**CAD 导航** [CADnavigation]，对一些用户来说是不直观的，因为它需要同时点击鼠标左键

和中键来旋转。**鼠标**[Mouse]...**显示**[Display]首选项中的按钮将显示执行每个操作的按钮组合。

5. 选择该框，然后点击空格键以切换对象可见性属性。可见性是屏幕上任何对象的许多属性之一。在仍选中该框的情况下，检查组合视图的下部。所选对象的属性显示在两个选项卡（**视图**[View]和**数据**[Data]）上。.

6. 单击**测量距离**图标，然后在 3D 窗口中的框上单击两个点。将显示一个尺寸，并将距离对象添加到树中。



7. 您可以在树中选择距离对象并浏览其属性。

8. 随着项目复杂度的增加，您可能需要同时对多个对象执行操作。用鼠标选择时，按住 *Ctrl* 键可以选择多个对象。

怎么运行的...

“查看/数据”选项卡反映了 FreeCAD 的深刻设计特征。构造对象所需的数据（大小，类型，位置和旋转）与在屏幕上显示该对象所需的数据（颜色，线条样式，可见性和透明度）分开。

FreeCAD 中的大多数操作都是非破坏性的，并导致将新对象添加到项目树中。每个新对象都有其自己的“视图”和“数据”选项卡，并保留与其简单前体的关系。这样就可以删除后面的对象，并在需要时还原为较简单的对象。在添加和修改对象时，它们会在组合视图树中以分层方式进行组织。通常，将显示最后一个子对象，并自动隐藏父对象。

还有更多...

FreeCAD 使用术语“视图”表示。参考几个相关的概念。

默认视图

工具栏中提供了八个默认视图和图标，使您可以轻松地在屏幕上放置项目。这些视图实际上是您工作的方向，不应与前面描述的视图窗口混淆。

术语“视图”也指摄影机投影类型：正交或透视。

冻结和保存视图

通过冻结（Shift+F），可以保留您喜欢的工作方向。这将为冻结视图添加一个菜单项，以使您稍后可以轻松返回该方向。通过冻结（Shift+F），可以保留您喜欢的工作方向。这将为冻结视图添加一个菜单项，以使您稍后可以轻松返回该方向。

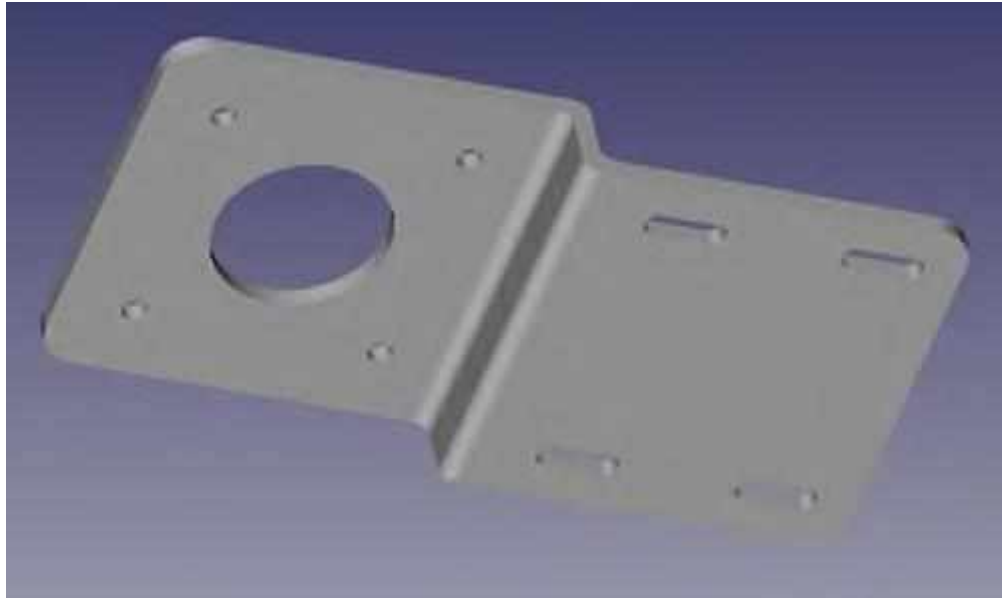
Python 和报表视图

FreeCAD 还有另外两个重要的窗口，它们在默认情况下是隐藏的，并且也被混淆地称为“视图”。本书的后续部分将详细讨论 Python 控制台。报告视图提供有关 FreeCAD 在后台执行操作的有用信息。

零件工作台中的 CSG 建模（必须知道）

构造实体几何（CSG）[ConstructiveSolidGeometry (CSG)]是将原始实体与布尔运算结合起来以创建更复杂形状的一种建模技术。FreeCAD 的 CSG 工具可在**零件工作台** [Partworkbench] 上找到。

该配方将使用零件工作台为 NEMA17 步进电机的偏置安装支架建模简单零件。



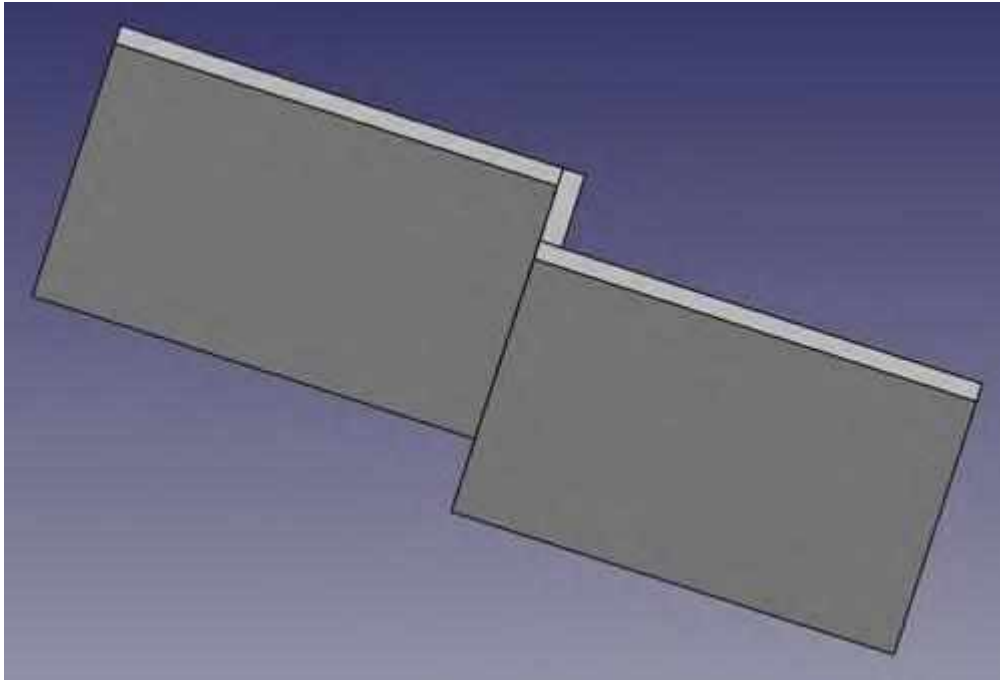
做好准备

创建一个新的空文档，并使用**设置为顶视图** [Set to top view] 顶视图图标或按数字

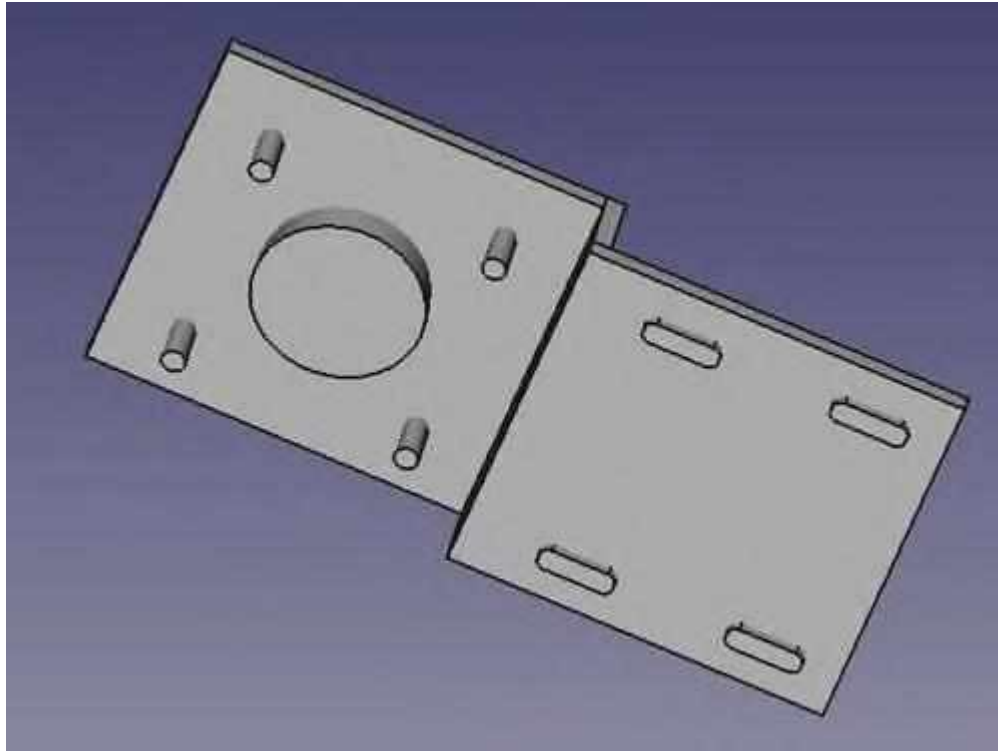
2 将视图定向到顶部。使用工作台更换器切换到**零件** [Part] 工作台。

怎么做...

1. 通过添加和定位三个框体来对弯曲的金属板建模。将第一个框添加到工程图，然后在项目树中选择其节点。切换到**数据** [Data] 选项卡并编辑属性以使其为 2.5x50x50mm（高度，长度，宽度）。
2. 创建一个具有相同尺寸的第二个框，但是编辑位置以将框在 X 方向上移动 50mm，在 Z 方向上移动 10mm。这会将第二个立方体放置在第一个立方体旁边并向上移动。
3. 添加第三个多维数据集以连接它们。使其尺寸为 10x2.5x50mm，并将其放置位置在 X 方向上移动 50mm。
4. 如果稍微旋转视图，则三个实体应如下图所示：



5. 接下来，我们将创建插槽。FreeCAD 没有插槽功能，因此我们将以所需孔的形状制作实体。稍后，我们将从平板中减去它以创建孔。
6. 添加另一个框并将其尺寸设置为 10x10x3mm。
7. 我们想要的槽有圆角的结束，所以我们将圆角的盒子。选择方框并按下菜单 **零件**[Part] | **圆角**[Fillet]。
8. 组合视图将切换到**任务**[Tasks]面板并显示圆角选项。您可以使用它来细化边缘选择并设置圆角的半径。对于我们的示例，选择 **Edge1**, **Edge3**, **Edge5** 和 **Edge7**。我们将半径值设为 1。单击**确定**[OK]按钮将应用更改。如果输入有误，请双击树中的圆角对象以更改属性，然后重试。.
9. 多维数据集看起来正确后，选择它并与菜单项**编辑**[Edit] | **复制**[Duplicate selection]复制它。选择重复。再重复两次，这样我们总共有四个圆角对象。选择每个并更改其放置属性，以将其移动到板的凸起部分上的位置。该职位取决于您。只要确保圆角物体完全穿透盘子即可。
10. 现在，为孔添加五个圆柱体。更改其放置和半径属性。中心孔为 22.5 毫米，小螺丝孔为 2.5 毫米。
11. 将所有实体添加到项目后，它应如下图所示：



12. 我们几乎准备好从底板上减去圆柱和槽，但是我们首先需要清理一下。在项目树中，选择所有圆柱体和圆角对象。选中所有选项后，单击**创建多个形状的并集**[Makeaunionofseveralshapes]图标：



13. 现在，为我们的底板选择三个框，然后重复操作，将它们融合为一个对象。

14. 现在，树中应该有两个融合对象，一个用于基板，一个用于从中切出的孔。首先选择底板融合，然后在按住 Ctrl 的同时选择另一个。单击**切成两个形状**[Makeacutoftwoshapes]图标：



15. 选择对象的顺序很重要。第一个对象将被第二个对象剪切。如果一切顺利，您将拥有一个对象，该对象几乎像使用说明开始时的图像一样。

16. 为了完成它，我们在板的角和折弯的角上应用了另一轮圆角。完整的对象应类似于本使用说明开始时的图像。

怎么运行的....

联合操作将对象添加在一起。实体不必接触或重叠。合并后，他们将有一个单一的位置和方向。

不同的布尔值将第一个对象与第二个对象切开。如果两个对象不重叠，则原始对象将不受影响。

相交消除了除原始两个对象重叠的区域以外的所有内容。

执行任何这些操作之后，将创建一个新对象，原始项目将成为其隐藏子项。您

可以删除父节点并切换可见的原始对象，以返回到原来的位置。可以双击圆角和倒角之类的对象，以重新打开创建它们的对话框并调整选项。

还有更多...

通常，有多种方法可以完成相同的工作，而**零件**[Part]工作台具有一些隐藏的功能。

创建基元

FreeCAD 可以创建比工具栏中更多的原始类型。使用**创建基元**[Createprimitives]工具创建许多其他工具。例如，类似弹簧的参数螺旋。

为什么我的部分看起来不正确？

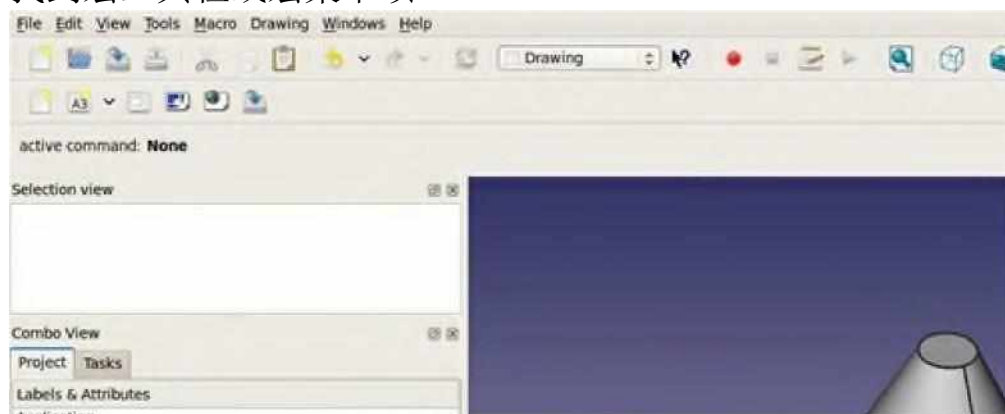
如果您的零件与本使用说明开始时的示例不同，请尝试使用最终对象的视图[View]属性。将**显示模式**[DisplayMode]更改为**阴影**[shaded]。

记录和编辑宏（应该知道）

宏是代表一系列命令或按键的命令，可以使繁琐的工作从重复的任务中解脱出来。宏录制可用于自动执行不需要完整脚本编写的任务。本使用说明将介绍记录，编辑和运行宏所需的步骤。

做好准备

找到宏工具栏或宏菜单项。



怎么做...

1. 按下以下屏幕快照中所示的红色记录按钮：



2. 宏录制对话框将弹出。为您的宏命名并保存：



3. 执行您要记录的任务。您可以创建在 FreeCAD 中通常执行的操作的宏。处理简单零件创建的宏最容易入手。尝试在**零件**[Part] 一个框并将其宽度更改为 20mm。

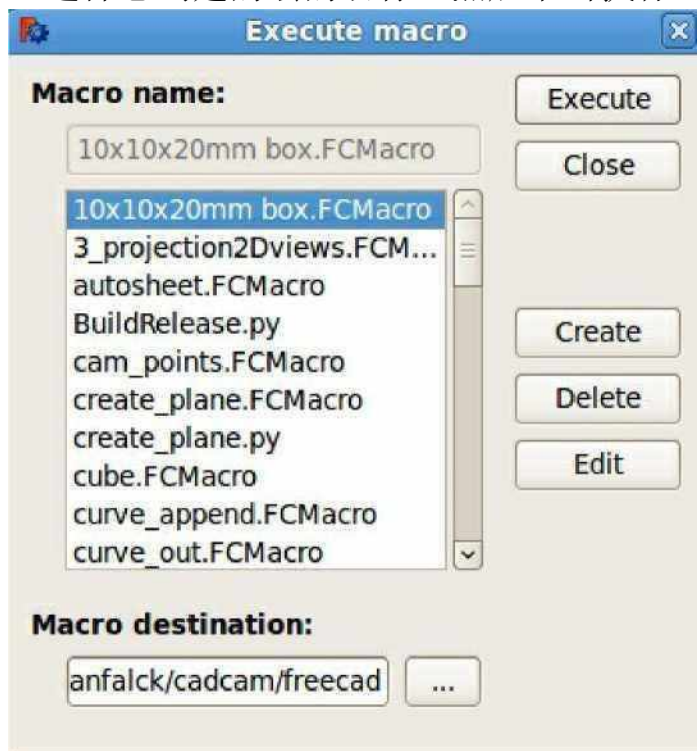
4. 完成记录任务后，请按绿色按钮停止记录：



5. 通过选择看起来像素描板的按钮来运行宏：



6. 选择您创建的宏的名称，然后单击执行：



怎么运行的...

在 FreeCAD 中，宏只是保存为 Python 脚本的命令的逐步表示，带有 FreeCAD 宏文件扩展名 `*.FCMacro`。在 FreeCAD 中执行任务时，它们具有相应的 Python 命令，该命令在后台运行。FreeCAD 宏仅捕获这些命令并将其保存到可重复使用的文件中。

如果您在保存宏时遇到问题，可能是由于 FreeCAD 的宏首选项设置造成的。打开一般偏好通过点击**编辑**[Edit] | **首选项**[Preferences] | **常规**[General]，然后

点击**宏标**[Macro]项卡。在**宏记录设置**[Macro recording settings]下，**宏路径**[Macro path]应该有一个到文件目录的有效路径。如果没有，则单击右侧的按钮，并使用有效的目录名进行更正。

还有更多...

大多数 FreeCAD 命令都可以通过宏和 Python 脚本访问。

利用宏的强大功能

宏实际上在执行时记录 Python 指令。我们也可以稍后阅读和编辑它们。使用宏对话框中的**编辑**[Edit]按钮打开我们的新宏。这将在 FreeCAD 的内置宏编辑器中打开我们的宏。以下是我们的代码：

```
# Macro Begin: /home/freecad/10x10x20mm box.FCMacro ++
import FreeCAD
import Part

#Gui.activateWorkbench("PartWorkbench")
App.ActiveDocument.addObject("Part::Box","Box")
#App.ActiveDocument.recompute()
#Gui.SendMsgToActiveView("ViewFit")
FreeCAD.getDocument("Unnamed").getObject("Box").Wi
# Macro End: /home/freecad/10x10x20mm box.FCMacro
```

下载示例代码[**Downloading the example code**]

您可以在 <http://www.PacktPub.com> 上从帐户下载所有 Packt 书籍的示例代码文件。如果您在其他地方购买了此书，则可以访问 <http://www.PacktPub.com/support> 并注册以将文件直接通过电子邮件发送给您。

FreeCAD 不会执行以**#**开头的行；它们是 Python 注释。注释可帮助我们了解代码中正在发生的事情。实际上仅有的几行代码

执行如下：

```
import FreeCAD
import Part

App.ActiveDocument.addObject("Part::Box","Box")
FreeCAD.getDocument("Unnamed").getObject("Box").Wi
```

以 **import** 开头的语句打开了一些宏模块，这些宏模块才能使我们的宏正常运行。**importPart** 使我们制作几何对象，例如盒子，圆柱体，球体等等。

App.ActiveDocument.addObject("Part::Box","Box") 实际上创建了该框。

您可以编辑宏，以执行宏在录制时最初没有执行的操作。通过按下看起来像文本编辑板的按钮，在内置的 FreeCAD 编辑器中打开宏。选择您的宏，然后选择 **编辑**[Edit] 按钮。您可以更改许多不同对象的参数值。在我们的示例宏中，您可以将以下两行更改为不同的名称和长度：

```
App.ActiveDocument.addObject("Part::Box","newbox")FreeCAD.getDocument("Unnamed").getObject("newbox")
```

通过选择 **文件**[File] | **保存**[Save] 更改。保存在顶部菜单中。然后，您可以按绿色的播放图标来尝试更改的宏：



通过阅读宏脚本来学习一些 Python 编程

创建和编辑宏是学习如何在 FreeCAD 中使用 Python 的好方法。

在编辑器中打开您的宏，然后通过更改不同的参数来使用它们并重新运行它们。

要了解有关 Python 编程语言的更多信息，请访问

<http://docs.python.org/tutorial>。

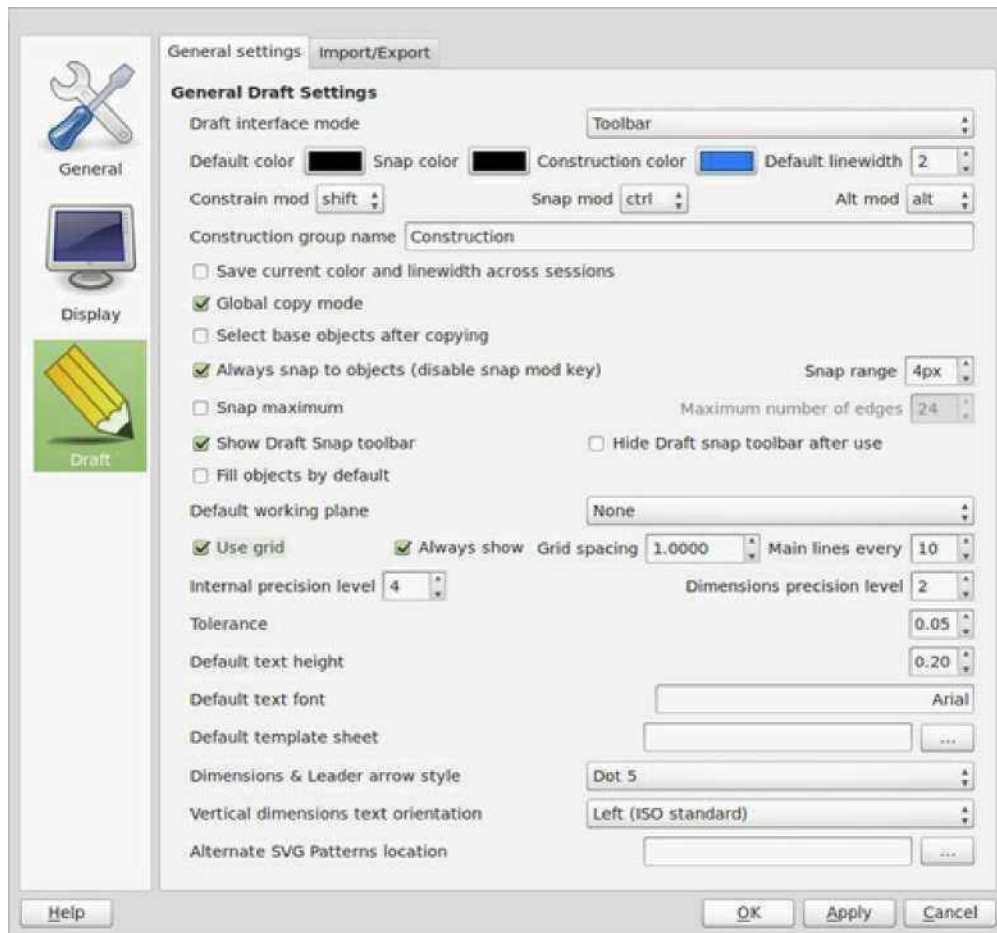
使用草稿工作台为简单零件建模（必须知道）

许多 3D 设计都是从二维图形开始的，这些图形被拉伸或旋转为三维。例如，可以将二维正方形挤出为三维立方体。草稿工作台提供了用于处理 2D 几何的工具。

在此配方中，我们将 **绘制**[Draft] 并编辑零件的 2D 轮廓。在以后的使用说明中，我们将旋转此轮廓以创建定时皮带轮的 3D 模型。

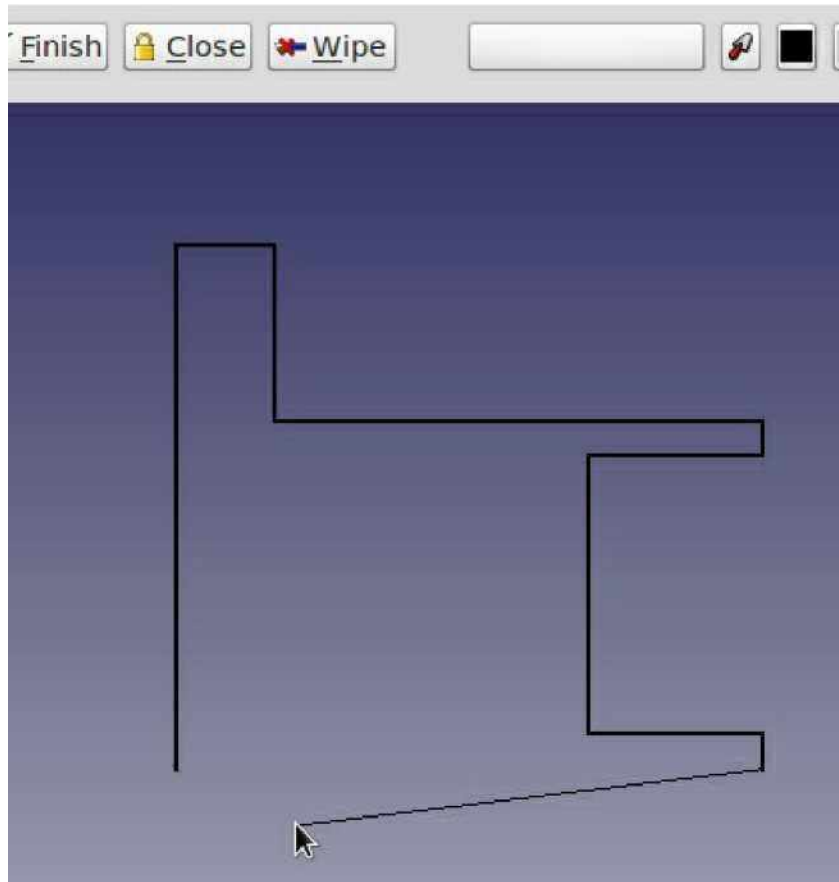
做好准备

启动一个新项目，然后切换到 **草稿**[Draft] 工作台。确保您的视图设置为顶部。更改草稿首选项，使其更容易对齐网格。将 **网格间距**[Gridspacing] 值设置为 1.000，并将主行中的每个值设置为 10。将 **对齐范围**[Snaprange] 设置为约 4 像素。

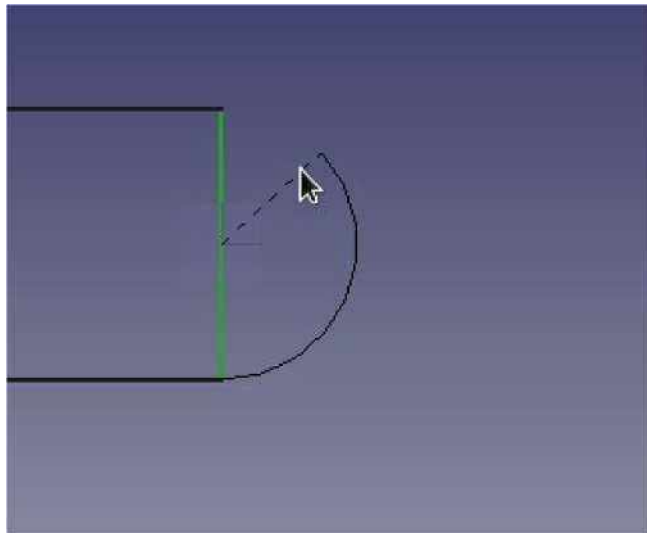


怎么做...

1. 选择导线工具。取消选择**相对**[Relative]复选框。可以通过单击画布或通过键入坐标来手动添加导线的点。手动输入 $X=3.175$, $Y=-5$ 和 $Z=0$ 的第一点。您可以在各字段之间制表，然后在 Z 上按 Enter 键以创建点。
2. 继续在 $X=3.175$, $Y=10$, $Z=-0$ 处添加第二个点。
3. 输入顺时针旋转的其余点，直到导线如下图所示。剩余的点可以通过单击画布并根据需要捕捉到网格来添加。按住 Shift 键将强制新段正交。输入最后一点但返回起点之前，请单击**关闭**[Close]以自动关闭导线并退出工具：



1. 在编辑导线的任何部分之前，我们必须将其分解为一组边线。在项目树中选择导线。选择**草稿**[Draft]菜单，然后单击三次**降级**[Downgrade]按钮。
2. 我们将圆弧的正方形边缘替换为圆弧。确保在捕捉工具栏上启用了中点和端点。使用圆弧工具在边缘上绘制曲线。



3. 对其他边重复上述步骤，然后删除旧的笔直边缘。
4. 选择树中的所有边缘。选择**草稿**[Draft]菜单，然后单击**升级**[Upgrade]按钮以将它们重新连接到线路中。

怎么运行的...

复杂的拓扑形状是由更简单的几何对象(如线、圆和弧)构成的。例如，一条**边**[edge]可以由连接两个**顶点**[vertices,]的直线构成，每个顶点只是空间中的一个点。边也可以是弧或圆。当多条边端到端连接时，它们就成了一条**线**[wire]。一个封闭的导线可以升级为一个**面**[face]，如果一个面集合完全包围了一个体块，它就是一个**实体**[solid]。如果一组面没有完全包围空间(即缺少一个或多个面)，则称为**外壳**[shell]。

草案工作台提供了直接创建、修改和转换这些数据类型的工具。

还有更多...

除了用于创建和修改 2D 几何的工具外，最新版本的**草稿**[Draft]工作台还具有一些强大而有用的功能。

创建数组

数组工具创建单个对象的数组。极坐标阵列对于创建螺栓圆和齿轮非常有用。正交阵列创建网格状排列。



选择要复制的对象，然后单击阵列工具。然后在**数据**[Data]选项卡中调整数组的属性。其中包括份数，旋转中心和阵列类型。

转换草稿和草图

草图[sketch]是在 FreeCAD 中绘制 2D 数据的另一种方式，我们将在以后的使用说明中对其进行详细介绍。草图和工程图具有不同的用途，但有时需要在它们之间进行转换。草稿工作台具有用于执行此转换的工具。请注意，特定于草图的数据在转换为草稿后将会丢失。

旋转和拉伸以创建零件（应该知道）

通过沿轴拉伸 2D 工程图或围绕轴旋转 2D 工程图，可以将其转换为 3D 模型。

做好准备

使用**草稿**[Draft]工作台绘制零件的闭合轮廓。对于我们的示例，我们使用“草稿”工作台（必须知道）配方中的“简单零件建模”中的定时皮带轮轮廓。

怎么做...

1. 切换到**草稿**[Draft]工作台。
2. 选择导线，然后单击**升级**[Upgrade]按钮将其转换为面。
3. 切换到**零件**[Part]工作台。

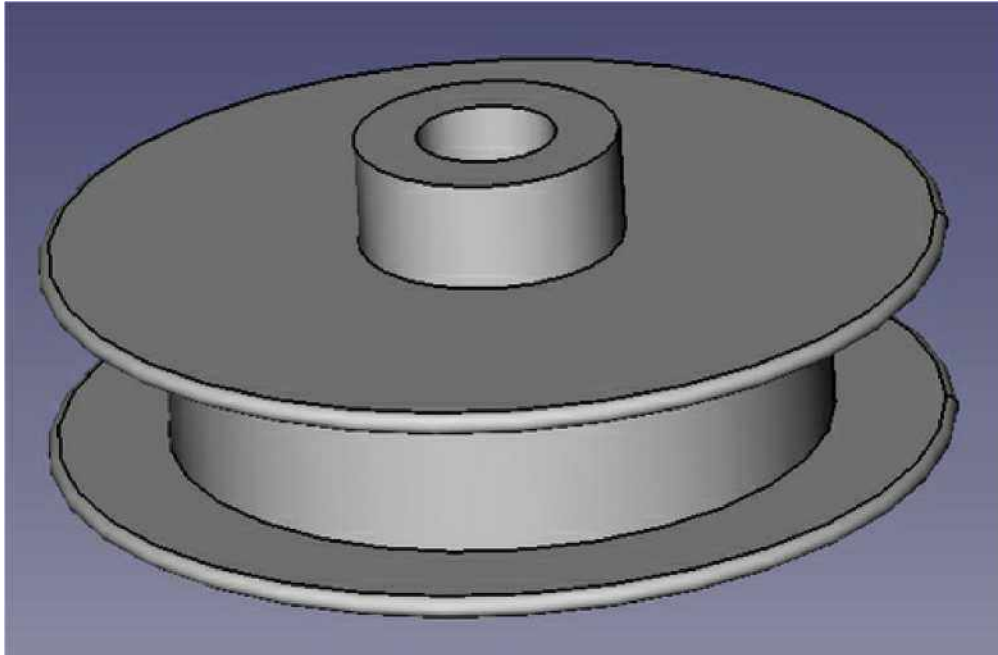
4. 按**撤消选定的形状**[Revokeaselectedshape]按钮：



5. 在**旋转**[Revolve]对话框中，选择要旋转的形状。

6. 选择要绕其旋转的轴。对于我们的示例，使用 Y 轴。保持其他设置不变，然后单击**确定**[ok]。

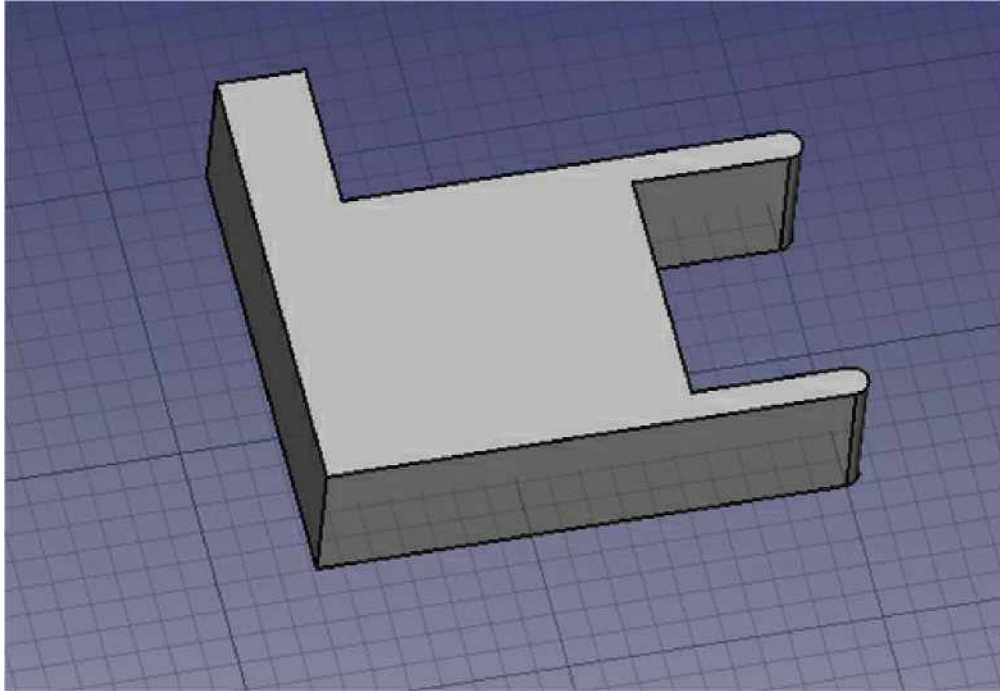
7. 旋转的对象将添加到项目中。



怎么运行的...

FreeCAD 有两种线对象。拉伸线在更高版本中称为**草线**[dwire]，在**数据**[Data]选项卡上具有其他功能和属性。在我们的示例中使用的电线包含弧线，并且是常规电线。如果要旋转导线而不先将其升级为面，则在屏幕上看到的结果将看起来完全相同。但是，该物体将是空心壳体。如果您执行其他布尔运算，则可能导致问题。通过旋转面而不是导线，旋转所扫出的区域将成为实体。

拉伸对象的工作原理类似。选择一个形状，并给出要拉伸的方向和距离的参数。在我们的原始电线上应用挤出会产生非常不同的结果。



零件工作台中的拉伸和旋转工具可用于包括草图在内的多种对象。产生的对象的类型取决于原始对象

输入形状	输出形状
顶点[Vertex]	边缘[Edge]
边缘[Edge]	面[Face]
电线（闭合）[Wire(closed)]	外壳[Shell]
草线[Dwire]	实体[Solid]
面[Face]	实体[Solid]
外壳[Shell]	复合固体[CompoundSolid]
草图[Sketch]	外壳[Shell]

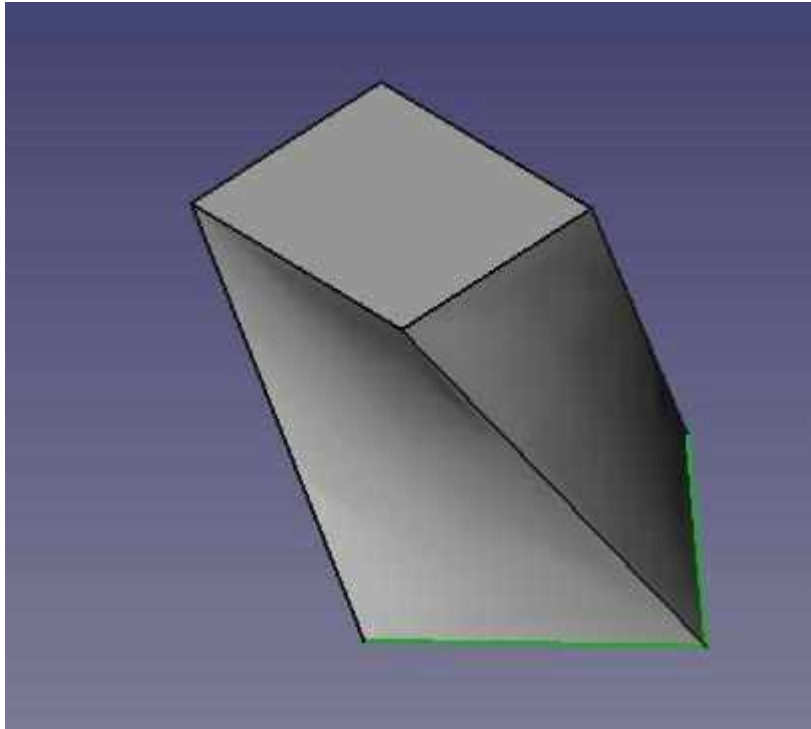
零件设计工作台具有自己的工具，用于旋转和拉伸（称为**填充**[padding]）草图以创建零件。零件设计垫和旋转工具只能在草图上使用。如果草图已关闭，它们始终会生成实体，否则将失败。

还有更多...

FreeCAD 还有另一个用于将 2D 几何图形转换为 3D 对象的工具。

放样

可以使用两条或多条导线定义实体的边界，放样工具将通过扫掠它们之间的区域来创建实体。在下面的屏幕截图中，两个矩形形状彼此重叠放置并稍微旋转。放样工具通过连接两者来创建扭曲的长方体：



使用 Python 创建 3D 实体（成为专家）

Python 是一种易于学习，易于阅读且功能强大的高级编程语言。要了解有关 Python 编程语言的更多信息，请访问 www.python.org。

FreeCAD 使用 Python 作为其脚本语言，使其非常灵活地建模零件。在 Python 的控制下，用户可以执行难以手动执行的操作。通过此配方，可以绘制伺服电动机的模型。

做好准备

对于此使用说明，需要 Python 控制台。确保它是打开的。在菜单栏中，单击查看|查看，然后确保已选中 Python 控制台。Python 控制台将位于 FreeCAD 的最底部面板中。它看起来类似于以下屏幕截图：



您还需要打开一个文档，以便 Python 脚本化的实体模型可以显示。

Python 区分大小写。确保完全按照显示输入以下配方。
Python 对缩进也很敏感，因此请注意不要在任何行的开头 添加额外的空格或制表符。

怎么做...

1. 在 Python 控制台中输入以下文本：

```
import Part
from FreeCAD import Vector
plate = Part.makeBox(40, 40, 5, Vector(-20, -20,
hole1 = Part.makeCylinder(1.5, 5, Vector(-15, -1
hole2 = Part.makeCylinder(1.5, 5, Vector(-15, 15
hole3 = Part.makeCylinder(1.5, 5, Vector(15, 15,
hole4 = Part.makeCylinder(1.5, 5, Vector(15, -15
faceplate = plate.cut(hole1)
faceplate = faceplate.cut(hole2)
```

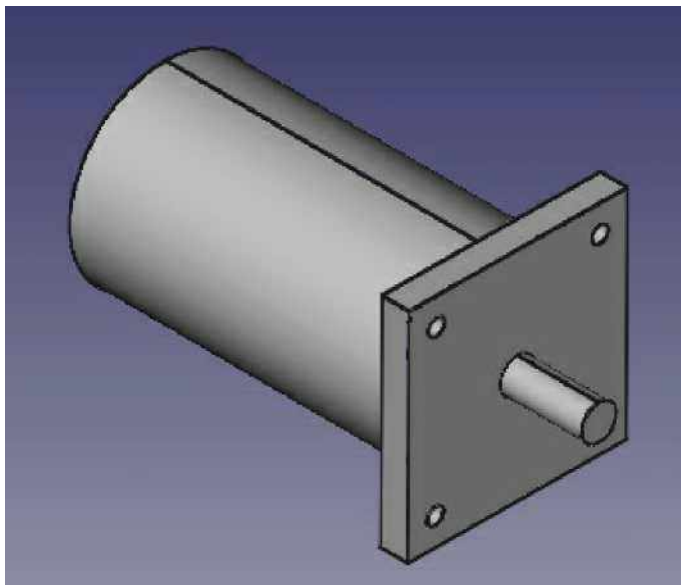
```

faceplate=faceplate.cut(hole3)
faceplate=faceplate.cut(hole4)
motorbody=Part.makeCylinder(17.5,60,Vector(0,0,0))
shaft=Part.makeCylinder(3.175,15,Vector(0,0,0))
servo=motorbody.fuse(faceplate)
servo=servo.fuse(shaft)
servo.translate(Vector(-20,-20,0))
servo.rotate(Vector(0,0,0),Vector(0,1,0),-90)
Part.show(servo)

```

2. 确保在最后一行之后按 Enter 键。

3. 您应该在 FreeCAD 的图形屏幕中看到一个 3D 实体，其外观类似于以下图形：



怎么运行的...

在脚本开始时，FreeCAD 需要导入其一些模块：

```

importPart
fromFreeCADimportVector

```

如果不导入这些内容，该脚本将无法执行标准 Python 库中已有的任何操作。零件模块允许访问 FreeCAD 中的几何对象。向量模块与平移和旋转对象有关。

在诸如 `plate=Part.makeBox(40,40,5,Vector(-20,-20,0))` 的行中，Part 模块为我们提供了一种创建尺寸为 40mmx40mmx5mm 大的实心盒子的方法，这将是我们的伺服电机型号的面板。它沿着向量移动；x 方向 20 毫米；y 方向 20 毫米；在 z 方向上为 0 毫米。FreeCAD 允许我们在 Python 中创建多种类型的实体，例如使用 `Part.makeCylinder` 的圆柱体，使用 `Part.makeSphere` 的球体，使用 `Part.makeCone` 的圆锥体以及使用 `Part.makeTorus` 的圆环。

FreeCAD 可以剪切 3D 对象并将其彼此融合。在 Python 代码中，如下所示：

`faceplate=plate.cut(hole1)`，我们正在切割带有 `hole1` 的板。可以在 `servo=motorbody.fuse(faceplate)` 这样的行中看到将对象融合在一起，其中将电机主体融合到面板上。

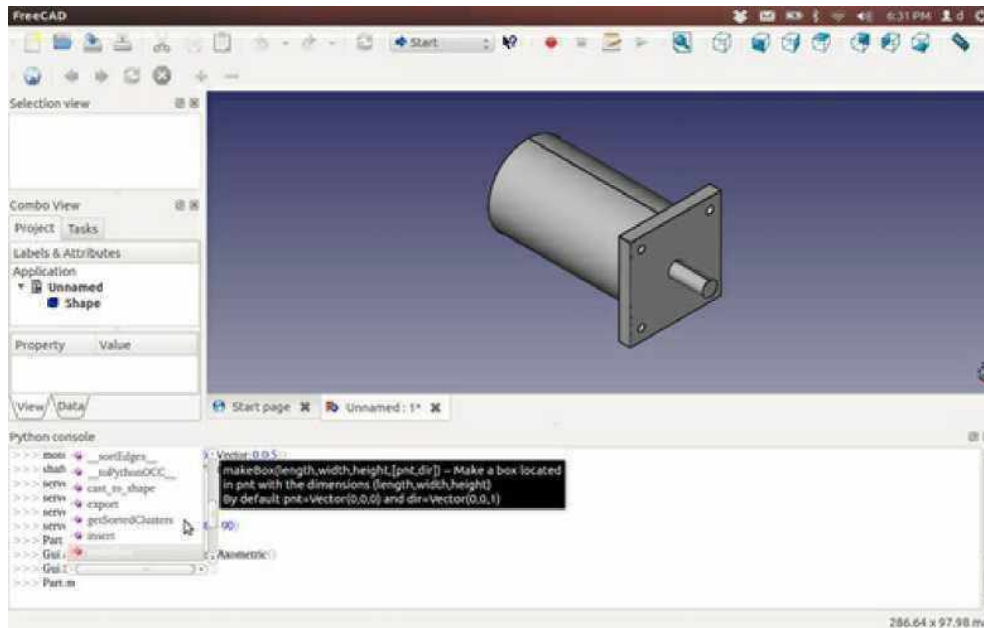
在脚本的最后，我们使用以下代码显示伺服电机，使其可见：

```
Part.show(servo)
```

还有更多...

您可以在 FreeCAD 中使用 Python 控制台来帮助您探索零件模块的深度。只需开始输入 Part.。进入控制台，FreeCAD 的自动完成功能将向您显示可用的类并提供提示。是什么导致控制台自动完成的。

这是实际的自动完成功能：



注意，它如何为您提供关于 makeBox 类的有用提示。

了解有关在线零件脚本的更多信息

转到 https://sourceforge.net/apps/mediawiki/free-cad/index.php?title=二_Topological_data_scripting 有更多关于如何在 FreeCAD 中使用 Python 的示例。通过 <http://www.thingiverse.com/> 使用关键字 **freecad** 搜索，也可以找到更多示例。

创建自定义对话框以自动执行任务 (成为专家)

FreeCAD 中的 Python 编程语言允许我们使用 PyQt4 或 PySide 添加我们自己的小部件以创建自定义图形用户界面。在此配方中，我们将创建一个对话框，让我们创建一个简单的框。

做好准备

如果草稿工作台在 FreeCAD 中正常运行，则您的计算机上已经安装了 PyQt4。打开一个新文档，以便有空间显示 3D 框。

怎么做...

以下是将弹出对话框的一些代码，使我们可以创建带有一些参数的 3D 实体框：

1. 完全按照所示在 Python 控制台中输入以下代码。Python 区分大小写和缩进。缩进要一致。我们将在此处每个缩进使用四个空格。目的在本文中看起来像一个标签。我已经在一些不完整的行中添加了 `\` (反斜杠) 形成。

```
from PyQt4 import QtGui, QtCore
import Part, FreeCAD
from FreeCAD import Base, Vector

class BoxExample(QtGui.QWidget):
    def __init__(self):
        super(BoxExample, self).__init__()
        self.initUI()

    def initUI(self):
        self.setGeometry(100, 100, 300, 200)
        self.setWindowTitle('Make a Box!')
        self.lengthLabel = QtGui.QLabel("Length")
        self.lengthLabel.move(50, 15)
        self.length = QtGui.QLineEdit(self)
        self.length.move(100, 15)
        self.widthLabel = QtGui.QLabel("Width")
        self.widthLabel.move(50, 50)
        self.width = QtGui.QLineEdit(self)
        self.width.move(100, 50)
        self.heightLabel = QtGui.QLabel("Height")
        self.heightLabel.move(50, 85)
        self.height = QtGui.QLineEdit(self)
        self.height.move(100, 85)
        self.centered = QtGui.QCheckBox("Centered")
        self.centered.move(80, 115)
        self.centered.setChecked(False)
        self.okButton = QtGui.QPushButton("OK")
        self.okButton.move(160, 150)
        self.show()
        QtCore.QObject.connect(
            self.okButton, QtCore.SIGNAL("pressed()"), self
        )
        def changeState(self, state):
```

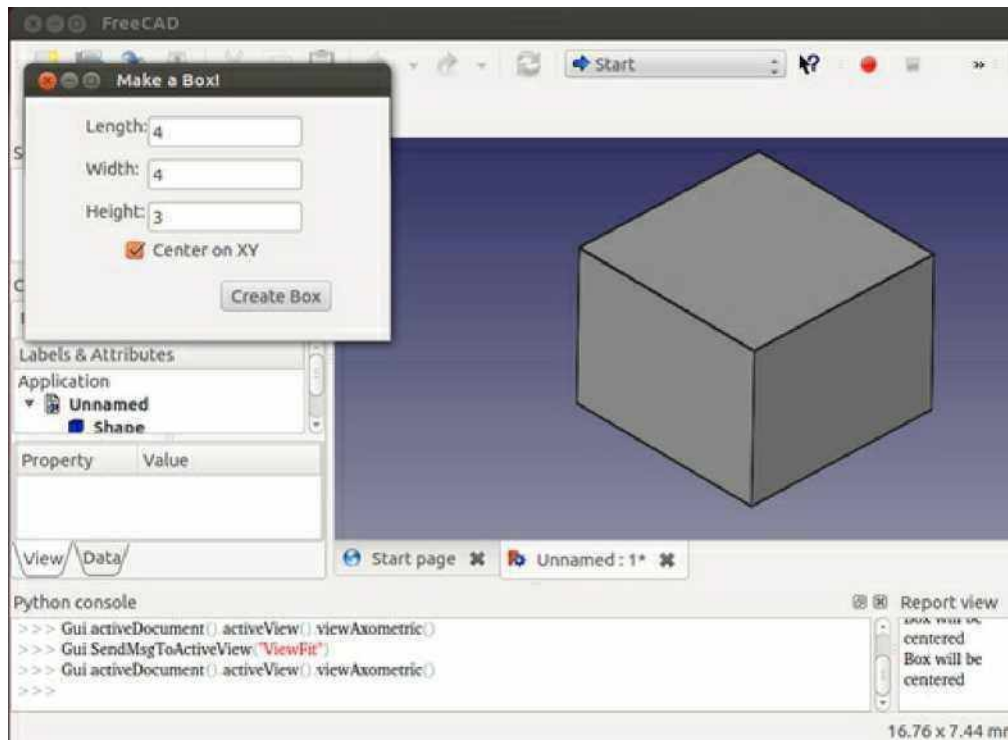
```

console=FreeCAD.Console
ifstate==QtCore.Qt.Checked:
console.PrintMessage("Boxwillbself.centerbox=True")
else:
self.centerbox=False
defbox(self):
l=float(self.length.text())
w=float(self.width.text())
h=float(self.height.text())
ifself.centerbox==True:
box=Part.makeBox(l,w,h)
box.translate(Base.Vector(-l/2,-
else:
box=Part.makeBox(l,w,h)
Part.show(box)
d=BoxExample()

```

2. 输入最后一行后，按两次 Enter 键。

3. 弹出对话框时，填写值，然后单击**创建框**[CreateBox]按钮。一个简单的 3D 框应出现在 FreeCAD 文档中。它看起来类似于以下屏幕截图：



怎么运行的...

我们首先创建一个包含所有对话框功能的类：

```
classBoxExample(QtGui.QWidget):
```

`BoxExample` 类中的函数 `definitUI(self)`：的一部分中的代码用于设置对话框的小部件。其中包含 `QtGui.QLabel` 的行允许我们标记使用 `QtGui.QLineEdit(self)` 制作的文本框。函数结尾处有一个复选框和一个按钮，分别为 `QtGui.QCheckBox` 和

QtGui.QPushButton 形式。defchangeState (self, state)：用于查看是否选中了 XY 中心[CenteronXY]复选框。

以下代码行用于将标有“创建框”的按钮连接到下一个功能：

```
QtCore.QObject.connect\  
(self.okButton,QtCore.SIGNAL("pressed()"),self.bo)
```

我使用\延续字符将一长行代码放入本书的格式化文本中，因此这两行似乎是 Python 的一行。

defbox(seif)函数可以在文档中创建 3D 实体框。box=Part.makeBox(i,w,h)创建盒子，Part.show(box)使它出现在我们的文档中。

在 FreeCADPython 脚本中，我们不使用主调用，就像在制作独立应用程序时那样。相反，我们使用 d=BoxExample() 来调用并显示我们的 BoxExample() 类。这就是打开我们的对话框的原因。

还有更多...

要了解有关 FreeCAD 对话框创建的更多信息，请访问

https://sourceforge.net/apps/mediawiki/free-cad/index.php?title=Dialog_creation.

了解有关 Python 和 PyQt 编程的更多信息

新 Python 程序员应该检查的第一个地方是官方 Python 文档网站，网址为

<http://python.org/doc/>。

在 <http://zetcode.com/tutorials/pyqt4/firstprograms/> 上也有关于 PyQt 编程的很好的介绍。

使用 QtDesigner 使事情变得更轻松

为了使编程对话框更加容易，您将需要使用 QtDesigner（图形对话框创建工具）。它将使您可以使用可以转换为 Python 代码的图形编辑器来创建对话框。如果您使用 UbuntuLinux 作为操作系统，请在 Synaptic 中查找 **qtdesigner**。

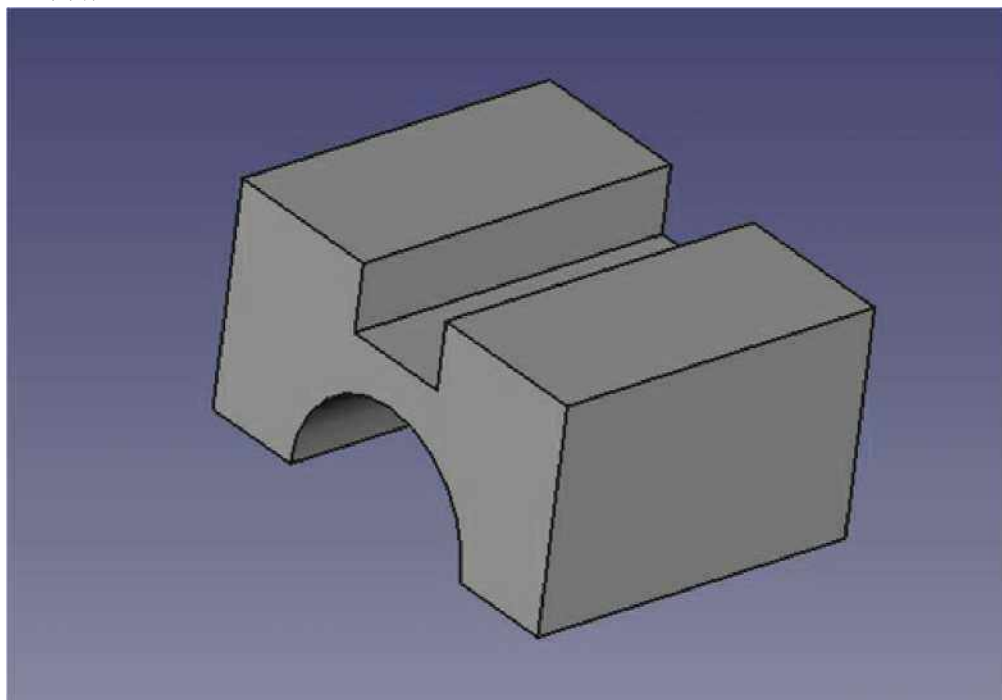
以下网页很好地介绍了为 FreeCAD 创建对话框：

https://sourceforge.net/apps/mediawiki/free-cad/index.php?title=Dialog_creation

有约束的建模（必须知道）

零件设计工作台中的许多工具在草稿工作台中似乎都很熟悉。但是，工作流程完全不同。建立在基于特征的设计思想的基础上，零件设计草图编辑器中的几何图形首先非常粗略地绘制，然后根据约束进行精炼。添加约束后，内置求解器将调整几何形状以满足约束要求。

在本使用说明中，我们将设计一个简单的零件，该零件在两个轴上具有特征，并显示如何在约束条件下进行修改。在以后的使用说明中，我们将向该部分添加功能。

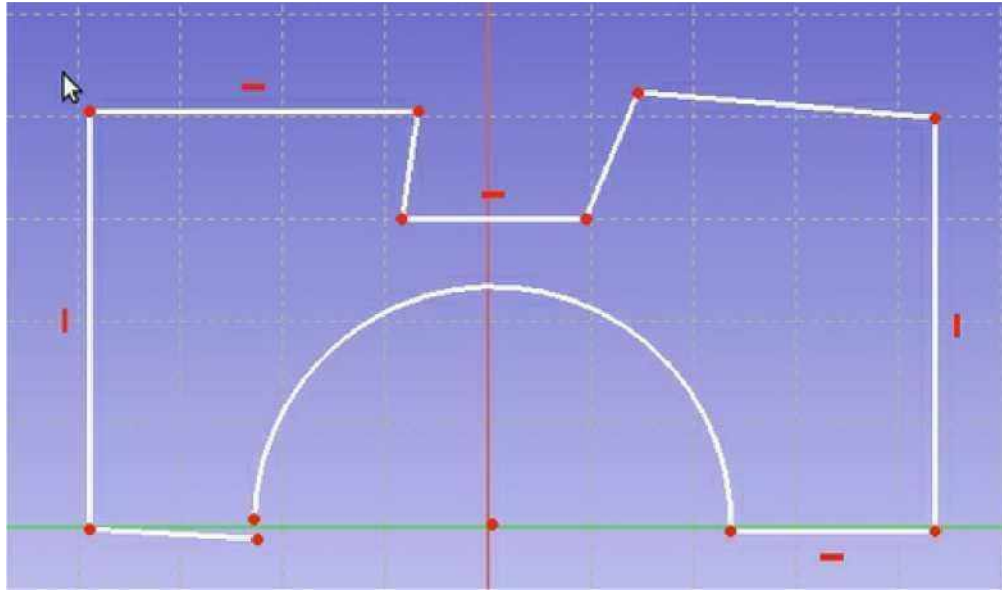


做好准备

从一个新的空文档开始。切换到零件设计工作台[PartDesign]后单击按钮以创建新的或编辑选定的草图。将会弹出一个对话框，询问您要在哪个平面上定向草图。选择 XZ-Plane 选项，然后单击确认[OK]。

怎么做...

1. 大致绘制零件的端部轮廓。现在不必担心准确性。以下示例是用圆弧和线绘制的：



2. 添加几何约束。通过选择草图中要约束的部分并按下适当的约束按钮来添加约束。新的约束图标将添加到图形中，并将在左侧面板中列出。

3. 首先向任何弯曲的线添加水平和垂直约束。如果应该连接线，请选择顶点并添加重合约束。

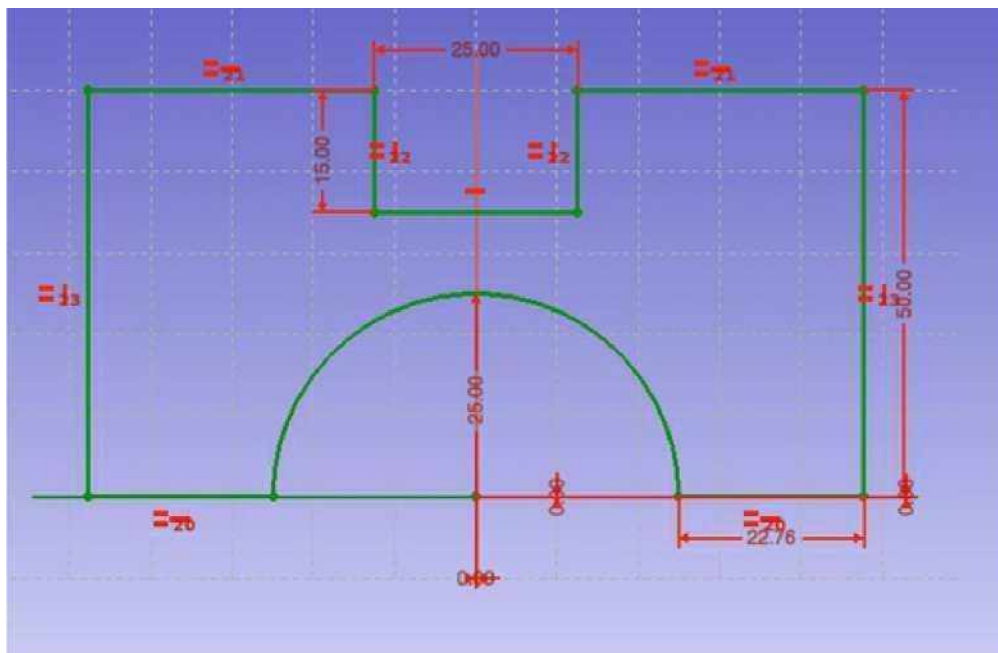
选择成对的匹配线并添加等式约束。这将迫使线等长。

4. 当草图看起来对称时，添加尺寸约束。首先，限制最重要的尺寸。例如，我们已将圆弧的半径限制为 25mm。

5. 添加约束时，请注意任务面板顶部的求解器消息。随着自由度

（DOF）的数量接近零，您可能会发现很难添加约束而又不会遇到冲突的约束错误。抓取顶点或线段并将其拖动可以向您显示草图仍受约束的位置。

6. 如果草图达到 2DOF 并拖动线段仅会移动整个草图，那么该锁定它了。选择一个顶点（我们使用了圆弧的中心），然后单击锁定按钮。这将添加垂直和水平距离约束，这些约束将顶点锁定到坐标系的原点。完全受限的草图将变为绿色。庆祝。



7. 单击任务面板中的关闭[**close**]。使用填充工具将草图填充到 75 毫米，然后旋转草图以查看零件。

8. 即使已填充草图，您仍然可以通过在树中双击它来对其进行编辑。保存后，您的更改将立即合并到零件中。

怎么运行的...

约束限制了对对象转换的方式。平面上的圆可以在两个方向上移动，并且半径可以更改。因此，它具有三个自由度。当其半径受约束时，它具有 2 自由度；当其中心锁定在 X 和 Y 中时，则视为完全受约束。

几何约束是那些影响实体形状或关系的约束。水平，垂直，切线，对称等。

尺寸约束有一个数字。双击约束以编辑值。长度，半径和角度都是尺寸限制。

对于具有零件之间隐含关系的离散零件，使用约束进行建模非常实用。

还有更多...

避免和消除冲突约束是大多数用户使用草绘器面临的最大挑战。练习对提高技能有很大帮助，但遵循一些基本准则也将有所帮助。

仔细应用约束

首先应用对称约束。接下来在尺寸标注之前应用几何约束（水平，垂直，切线等）

约束（带有数字的任何东西）。尽可能固定水平或垂直距离，而不要限制长度。

尝试最后应用坐标锁定。换句话说，在尝试将草图约束到坐标系之前，请先使其受到约束。

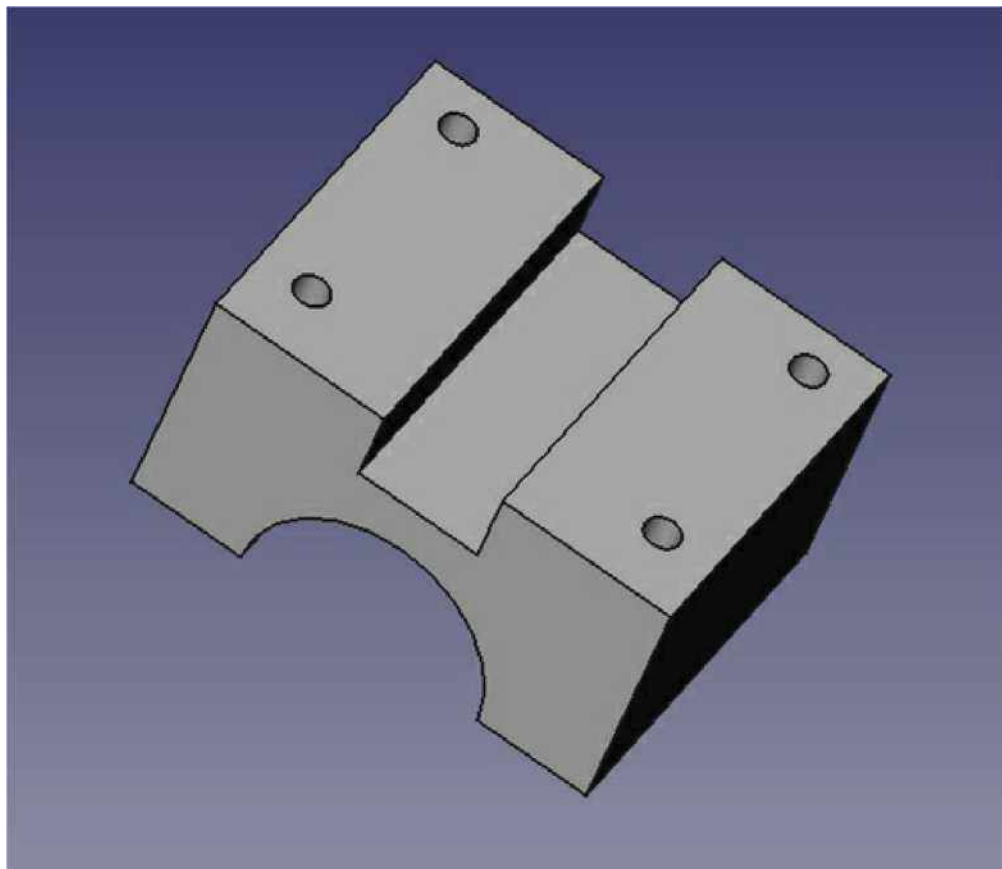
使用多个简单草图

通常，使用几个简单的草图来构建零件的复杂性要比使用一个具有很多约束的大型草图要容易得多。保持草图尽可能简单。尽可能在实体上使用倒角和圆角操作，而不是对草图中的斜角或圆角建模。

使用外部约束（应该知道）

我们要设计的零件的所有特征很少都在同一二维平面上，这是很少见的。想象一下，我们想在具有约束条件（必须知道）的建模中设计的零件的顶部表面添加孔，如以下屏幕快照所示。为此，我们需要一种将新草图与零件顶面关联的方法。

外部约束为我们提供了一种将草图元素与草图本身之外的实体（例如零件的面）相关联的方法。



在本使用说明中，我们还将展示草图中的要素如何与现有对象相关。在此过程中，我们还将介绍构造线和外部约束。

做好准备

该配方使用 FreeCAD 版本 0.13 及更高版本中的功能。确保您使用的是最新版本或开发快照。

使用约束条件（必须知道）打开在建模中创建的模型。将视图设置为顶部

[top]便您从上方俯视零件。

怎么做...

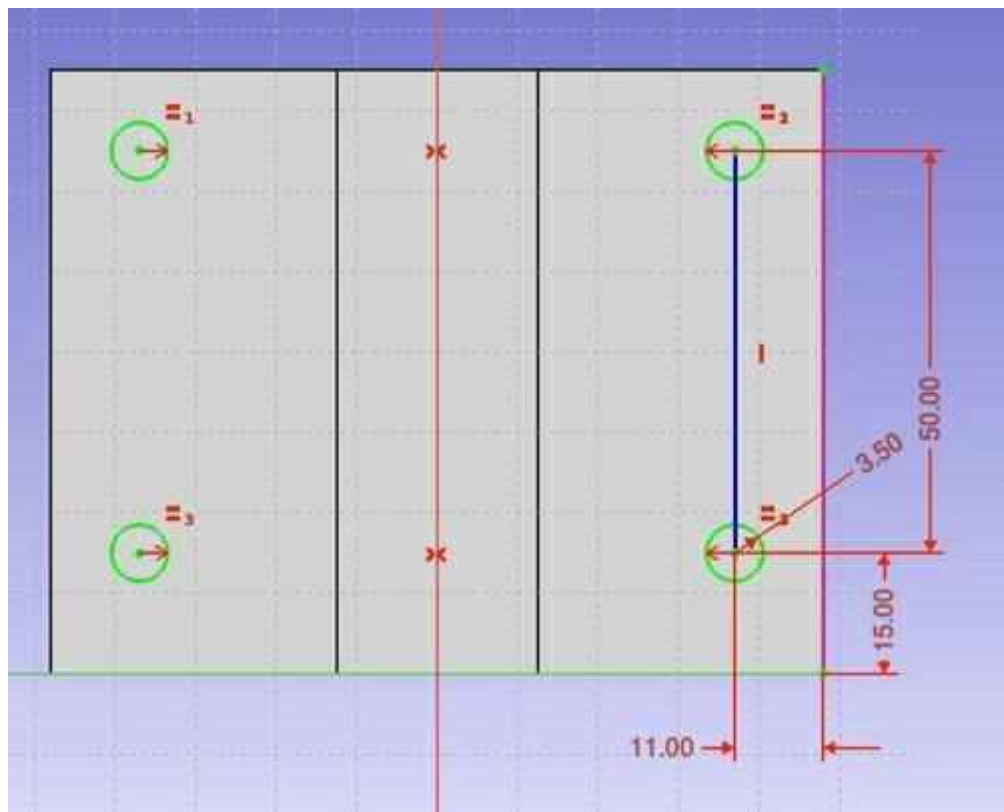
1. 选择零件的顶面使其突出显示，然后单击草图按钮。新草图将被映射到面部。
2. 画四个圆圈代表我们的螺栓孔。通过单击边缘（而不是中心）来选择所有四个，并设置相等约束以使它们的大小相同。
3. 选择这四个之一，然后添加一个半径约束。将半径设置为 3.5 毫米。现在，可以通过更改一个数字来调整全部四个的大小。
4. 我们希望孔对称放置。选择顶部两个孔的中心和垂直轴。添加对称约束。重复下面两个孔。现在，两组圆是独立对称的。
5. 从右上孔的中心到右下孔的中心画一条线。确保端点与每个圆的中心重合。选择该行，然后通过单击构造模式图标将其变成构造线。



6. 在工作线上将长度限制设置为 50mm。
7. 现在，我们需要一种方法来设置孔相对于零件边缘的位置。
8. 单击**外部约束**[ExternalConstraint]图标，然后选择原始实体的右边缘。现在可以在约束中引用边及其顶点，就像它们是草图的一部分一样。



9. 选择右下角圆的中心和外部约束线的下端点。添加水平距离约束（11 毫米）。重复并添加垂直距离约束（15mm）。



10. 草图应完全约束。要在整个图块上创建孔，请关闭草图并使用口袋工具。将口袋深度设置为至少 50 毫米。

怎么运行的...

可以在现有实体的平面上绘制草图以添加特征，例如凸起或凹陷。

可以将构造几何体添加到任何草图中。这些线，圆和弧线不会成为最终对象的一部分，但可以采用所有相同的约束。它们有助于绘制其他实体并在实体之间建立关系。

外部约束线类似于构造线，但是从其上绘制草图的面中选择的一条边。

还有更多...

要使用外部约束，必须将草图映射到面。通过选择对象的面开始草图时，会自动发生这种情况。但是，也可以将现有草图应用于最初未在其上绘制的面。

将草图映射到面

选择一个面，然后单击菜单。将**草图**[PartDesign] | **映射**[MapSketch]到面。

在对话框中，选择要映射到该面的草图，然后单击确定[OK]。将草图映射到面部会使草图进入编辑模式。如有必要，调整草图并将其关闭。草图可以重新映射到不同的面，但不能同时映射到多个面。

使用 Python 添加或修改约束（成为专家）

除了以交互方式和图形方式使用 Sketcher 之外，我们还可以使用 Python 自动对其进行编程。

做好准备

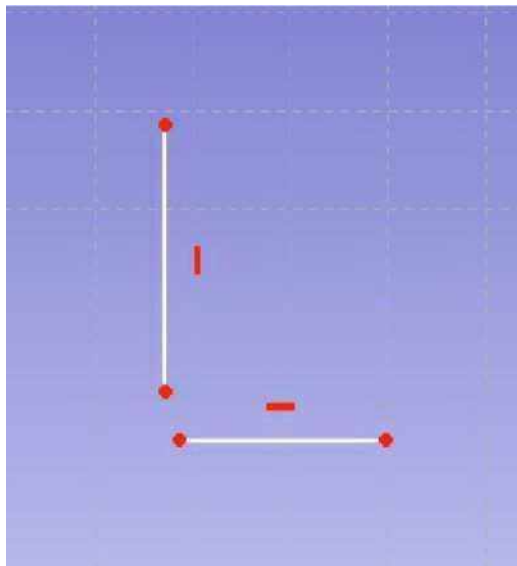
对于此使用说明，我们需要打开 Python 控制台。在菜单栏中，依次打开视图 [View] 和显图 [Views]，然后确保选中了 Python 控制台 [Pythonconsole]。

怎么做...

1. 在控制台中输入以下代码块:

```
fromSketcherimport*importPart
importFreeCADasAppfromFreeCADimportVectorif(App.activeDocument()==
None):App.newDocu
f=App.activeDocument().addObject("Sketcherf.addGeometry(Part.Line(v
ector(0,0,0),Vectorf.addGeometry(Part.Line(vector(0,0,0),Vectorf.Cons
traints=[Constraint('Vertical',0),\App.activeDocument().recompute(
)
```

2. 双击项目树中的草图[Sketch]图标。
3. 注意水平和垂直约束符号（红色小条）。另请注意，这些线具有端点，但不受限制。



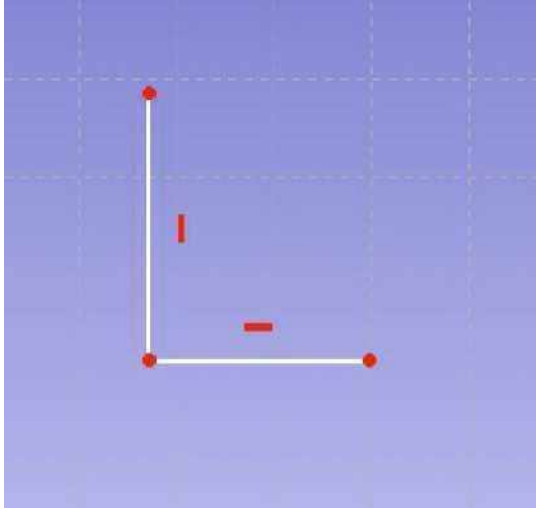
4. 抓住其中任何一条线并移动它，而另一条线则不会随之移动。
5. 向控制台添加更多 Python 代码以约束一些要点：

```
StartPoint=1;
l=f.Constraints1.append(Constraint('Coincident',0,StartPoirf.Constra
```

```
ints=lApp.activeDocument().recompute()
```

6. 请注意，每条线的一端如何连接到另一端。

7. 尝试在屏幕上移动其中一条线，您将看到这些线一起移动。



怎么运行的...

1. 导入 Sketcher 模块中的所有可用功能:

```
fromSketcherimport*
```

2. 我们将需要 Part 模块来制作几何对象:

```
importPart
```

3. FreeCAD 模块将使我们能够处理文档:

```
importFreeCADasApp
```

4. 我们需要给线段终点，所以我们需要:

```
fromFreeCADimportVector
```

5. 如果尚未打开文档，请创建一个新文档:

```
if(App.activeDocument()==None):App.newDoc
```

6. 创建一个新的 Sketch 对象:

```
f=App.activeDocument().addObject("Sketcher
```

7. 将几何体添加到草图:

```
f.addGeometry(Part.Line(Vector(0,0,0),Vector(2,20,0))
```

```
f.addGeometry(Part.Line(Vector(0,0,0),Vector
```

8. 在草图上添加一些约束以使线条水平和垂直:

```
f.Constraints=[Constraint('Vertical',0),\
```

9. 重新计算草图以查看更改后的内容:

```
App.activeDocument().recompute()
```

10. 让我们在行上为起点添加一个名称:

```
StartPoint=1
```

11. 创建一个与原始对象相等的代理对象，因为我们无法直接向其添加约束:

```
l=f.Constraints
```

12. 为我们的代理对象添加更多约束。0 是第一行，1 是第二行:

```
l.append(Constraint('Coincident',0,StartPoir
```

13. 使原始约束与代理相等并重新计算：

```
f.Constraints=lApp.activeDocument().recompute()
```

还有更多...

您可以使用 Python 使用尺寸约束几何。

向行添加长度约束

您可以将长度约束添加到第二行，如下所示：

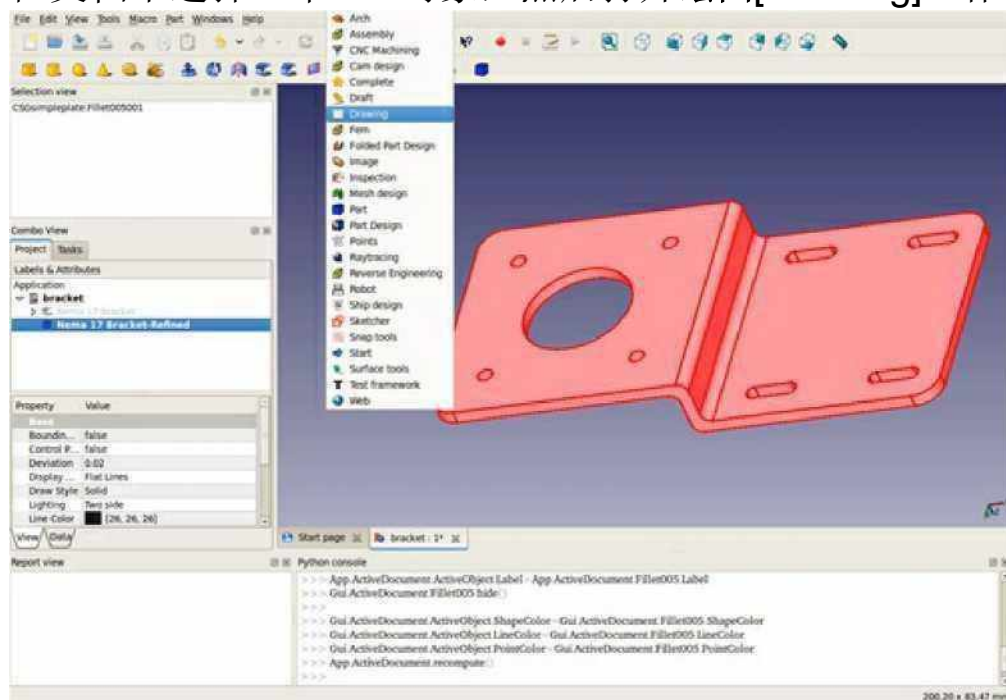
```
l.append((Constraint('DistanceX',1,20.0)))
```

创建零件图纸（应该知道）

通过“绘图”工作台，我们可以创建 3D 对象的 2D 视图，以适合打印的格式进行呈现。在此配方中，我们将创建一个具有三个视图的图形。

做好准备

在文档中选择一个 3D 对象，然后打开绘图[Drawing]工作台：

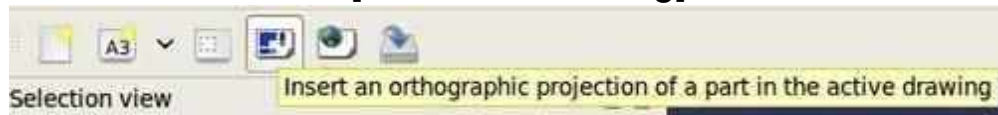


怎么做...

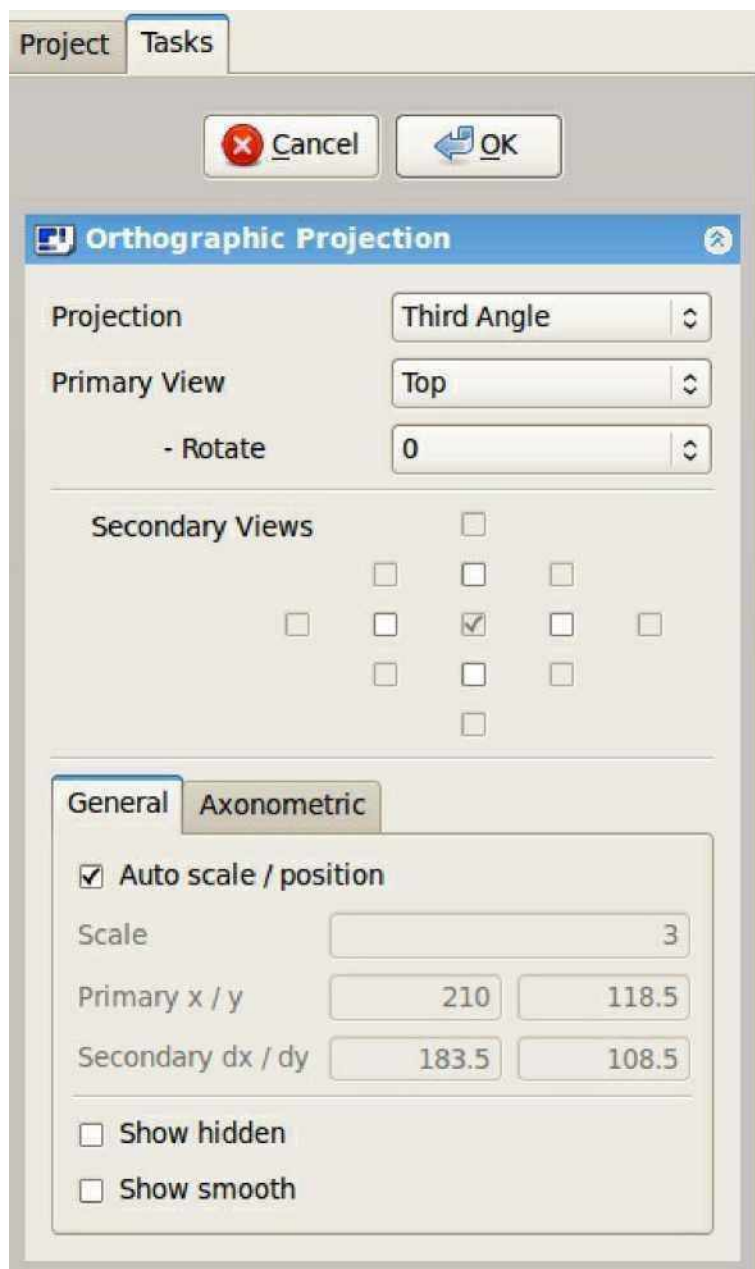
1.选择插入新图形图标：



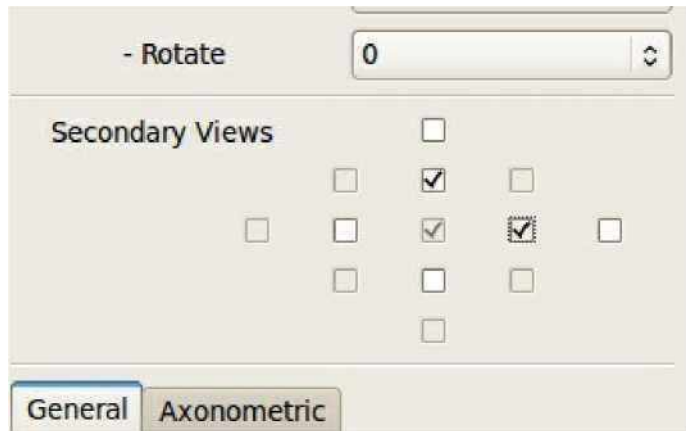
2.选择插入正交投影[Insertnewdrawing]图标:



3.新任务将在任务[Task]面板中弹出。选择一个主视图[TaskView]:



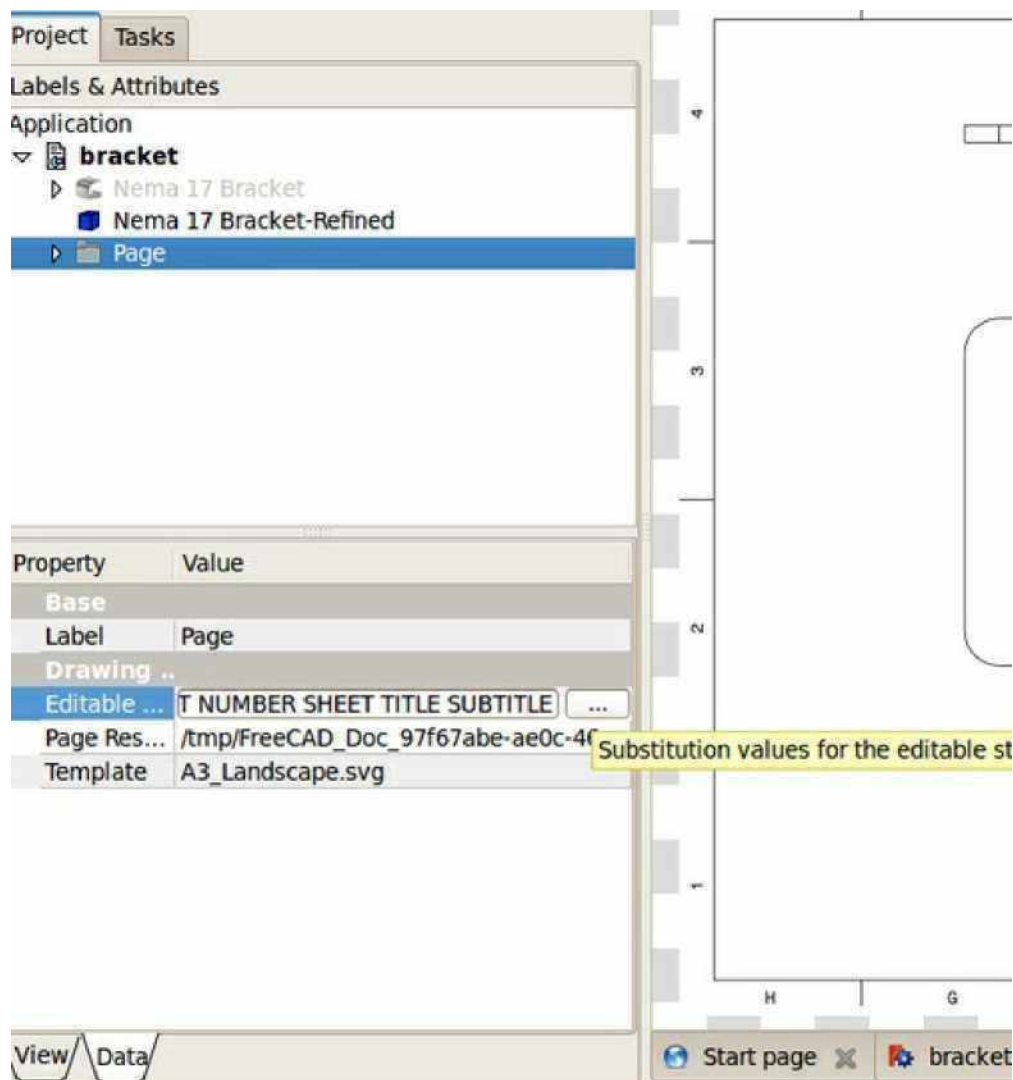
4.选择辅助视图[SecondaryViews]:



5.单击任务[Task]面板中的确定[OK]按钮。

6.项目树中选择页面[Page]图标后，单击其数据[Data]选项卡。

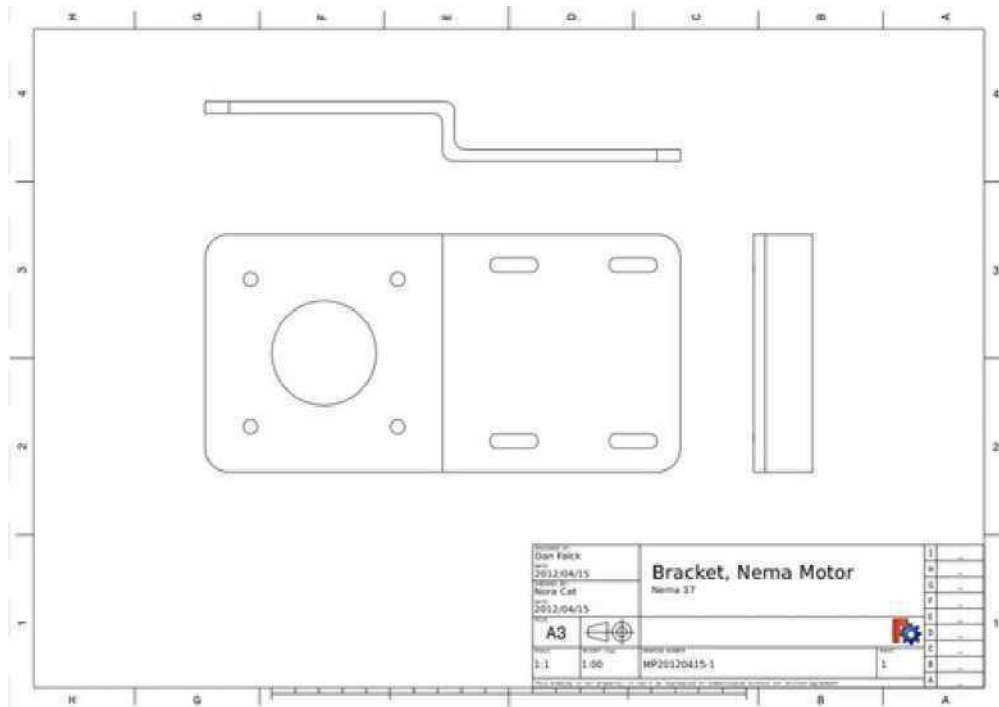
7.双击以...为标签的按钮。



8.将弹出一个文本对话框。编辑值，并输入您的姓名，日期等：



9.单击重新计算按钮以查看结果:



怎么运行的...

FreeCAD 使用 OpenCascadeCAD 内核中的算法来从 3D 对象计算 2D 投影。FreeCAD 充分利用了这一点，并将这些投影插入到 SVG 画布上。该画布实际上是在步骤 1 中导入的模板。

“任务”面板中的“正交投影”对话框以设定的角度创建不同的投影，对其进行缩放并以一致的方式隔开它们。

文本对话框可帮助我们编辑 SVG 可编辑文本，并在标题栏的右下角填充文本信息。

还有更多...

如果要将图形提供给没有 FreeCAD 的人，则可能需要将其导出为 PDF 文件。

从工程图导出 PDF 文件

从工程图导出 PDF 文件

在项目树中选择页面[Page]对象。从顶部菜单中，选择文件[File]，然后选择导出 PDF[ExportPDF]。给您的文件起一个*.pdf 扩展名，然后选择一个保存目录。

为其他应用程序导出 DXF 文件（应该知道）

大多数 CAD 应用程序都不是处于真空状态，需要能够保存为其他文件格式，以便其他 CAD/CAM 程序使用。FreeCAD 也不例外。FreeCAD 不能很好地完成某些任务，例如标注图纸尺寸。其他应用程序，例如 LibreCAD 或 DraftSight 在此方面表现更好。将 2D 几何图形导出为 DXF 格式是简化此操作的绝佳方法。

做好准备

导出几何的第一条规则是确保几何是干净且一致的。这意味着要连接的直线和圆弧的端点应无间隙地连接。隐藏在其他线段下方的线段应删除。不可能太小的弧线也应删除。不要在导出文档中留下不必要的任何额外实体。3D 几何图形在 FreeCAD 中不会导出为 DXF 格式。

怎么做...

1. 在项目树中，从草图[Sketch]或草稿[Draft]工作台中选择 2D 几何[2Dgeometry]。确保并选择要导出的所有几何。
2. 顶部菜单中，选择文件[File]，然后选择导出[Export]。
3. 在导出文件[Export]对话框的右下角，选择“AutodeskDXF”。
4. 在名称[Name]输入框中，为文件提供扩展名*.dxf。
5. 选择您要保存文件的目录。

怎么运行的...

FreeCAD 通常使用草稿[Draft]工作台中的脚本导出 2D 几何。

导入数据（应该知道）

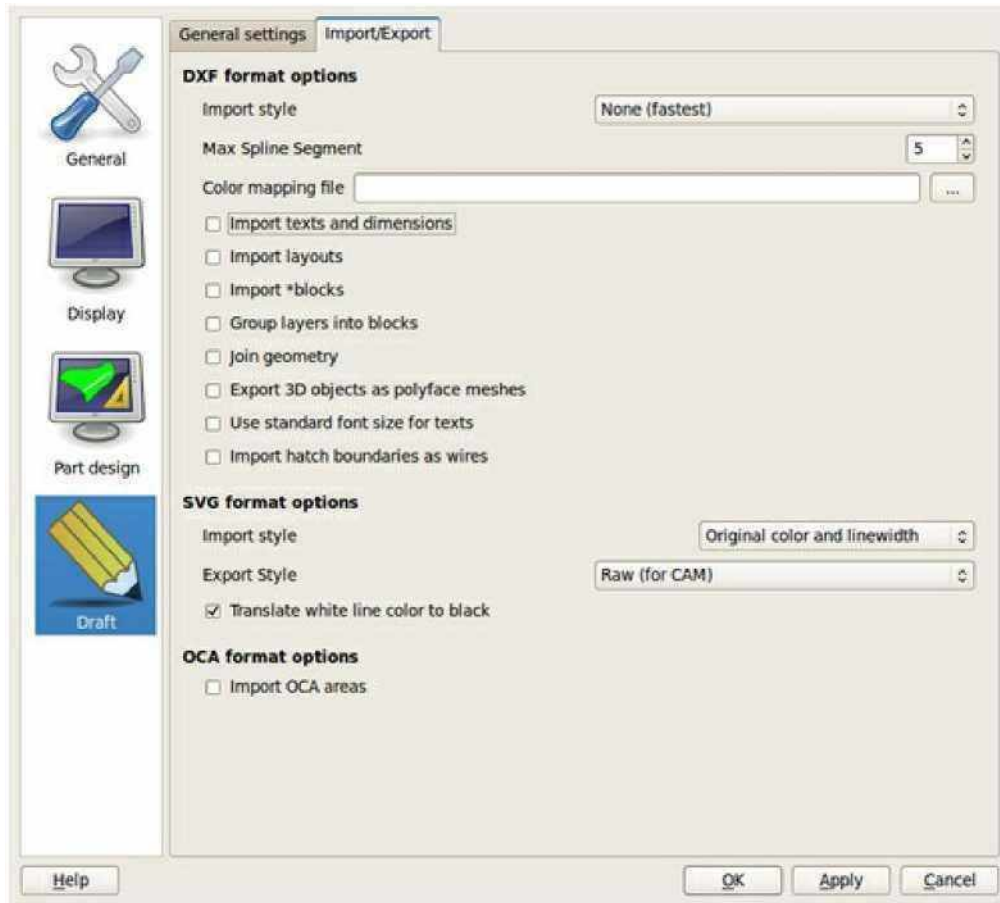
有时，希望使用在其他 CAD 或建模程序中创建的几何。FreeCAD 可以导入各种各样的文件类型。DXF，STEP，STL，SVG，VRML，Collada 和 IDF 只是它可以使用的一些文件格式。

在此配方中，我们将打开 DXF 文件，并使用其中的几何图形来形成 3D 实体。

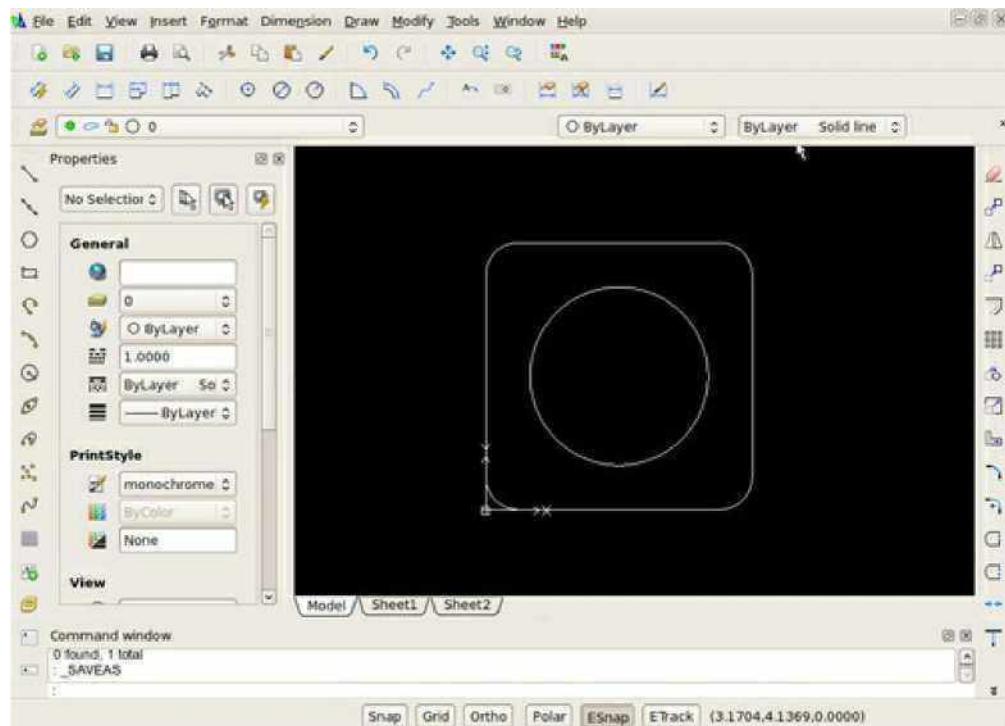
做好准备

您应该具有由 CAD 程序（例如 QCAD，LibreCAD，Draftsight 或 AutoCAD）生成的 DXF 文件，该文件可用于生成 DXF 文件。

- 1.在 FreeCAD 中，打开草稿[Draft]工作台，然后从编辑菜单[Edit]中选择首选项[Preferences]。
- 2.在“首选项”对话框中，单击“草稿”图标，然后单击“导入/导出”选项卡。
- 3.在 DXF 格式[DXFformatoptions]选项下，将导入样式[Importstyle]更改为无（最快）[None(fastest)]。
- 4.取消选中同一部分中的所有其他框。单击应用[Apply]，然后单击确定[OK]。



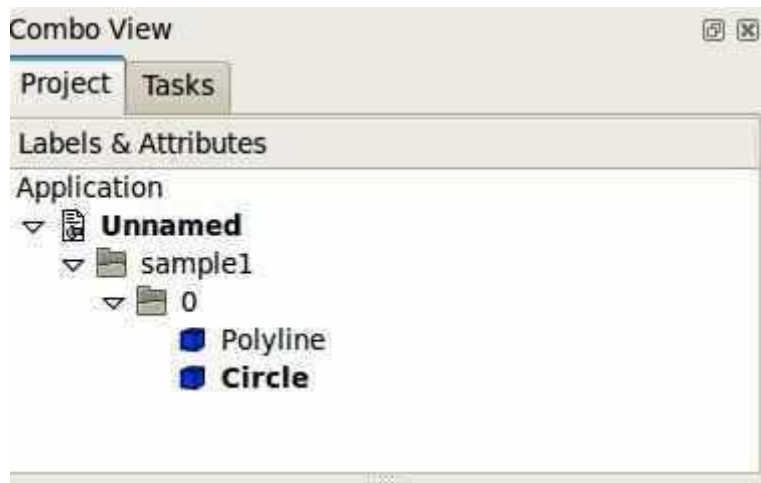
在另一个 CAD 程序中创建一些几何。对于此示例，我们将使用 DraftSight 创建一个简单的形状。以下是具有四个圆角和一个圆的折线矩形：



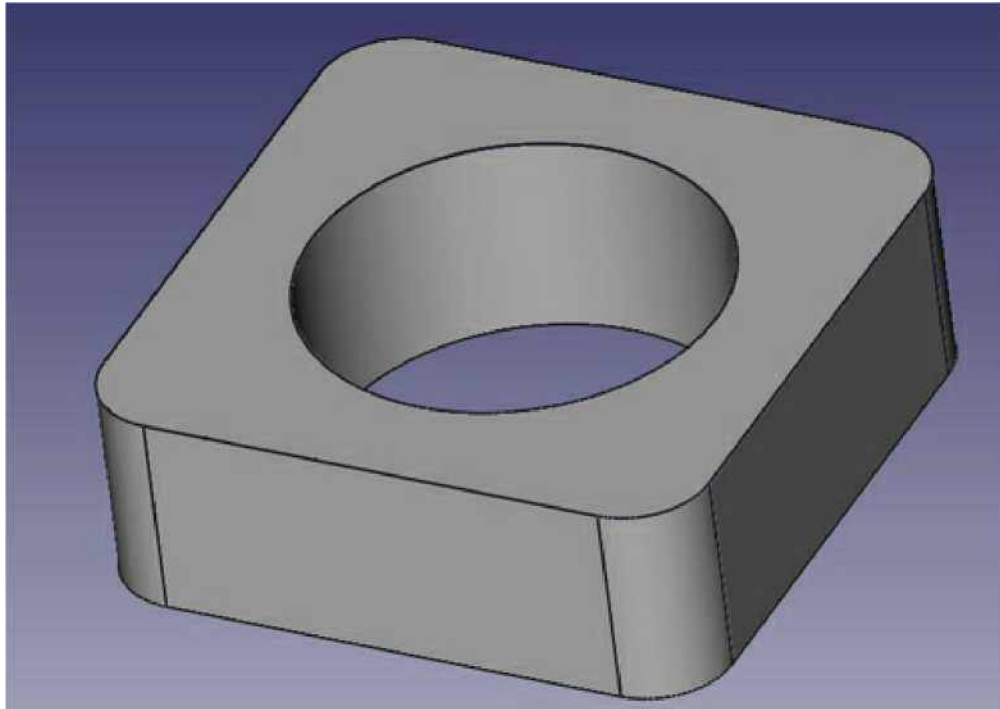
该图中的直线和圆弧是多段线。这些将很好地转换为 FreeCAD，并帮助我们创建实体以从中减去圆。将文件另存为 ASCII 类型 [dxf](#) (不要另存为二进制)。

怎么做...

- 1.在 FreeCAD 中，从文件[File]菜单中选择导入[Import]。
- 2.选择要导入的 [dxf 文件](#)。
- 3.在组合视图[ComboView]中，查看项目[Project]选项卡，然后展开图标。



- 4.切换到草稿[Draft]工作台，以便我们可以使用类似于向上箭头的图标来升级[Upgrade]圆形。
- 5.选择标记为折线的几何，然后使用带有拉伸[Polyline]图标的零件[Part]工作台将其拉伸为实体。
- 6.将标记为圆[Circle]的几何体挤出为实体。确保其高度与由 Polyline[Polyline]制成的实体相同。.
- 7.选择您从 Polyline 创建的实体，然后选择 Circle 创建的实体。
- 8.在零件[Part]工作台使用[BooleanCut]布尔切割操作从折线实体中减去[Circle]圆[Polyline]实体。



怎么运行的...

DXF 导入依赖于 Draft 工作台中的代码，因此可以通过 Draft 首选项控制其设置。导入的 dxf 折线作为可以在零件工作台中拉伸的面。草图工作台具有使用“升级”工具将其他 2D 几何图形更改为面的工具，该工具允许“零件”工作台从其生成实体。面孔将以实体形式挤出，而没有面孔的封闭形状只会挤出为看起来像墙壁的薄物体。