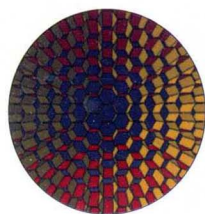
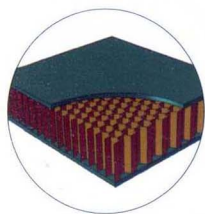
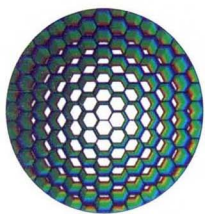


Abaqus GUI

程序开发指南

Python 语言

贾利勇 富琛阳子 贺高 周正光 编著



- + 内容涉及 Python 语言的基本语法、Abaqus 常用内核脚本、控件使用、Abaqus 插件开发和自定义应用平台开发等多个方面。
- + 专注于 Abaqus GUI 程序开发，详细介绍了 20 多种 GUI 控件的使用，可满足各类复杂 GUI 程序的开发需求。
- + 全彩印刷，书中所有代码均按免费软件 Notepad++ 中 Python 语言格式进行语法着色，方便读者阅读。
- + 作者通过 Simwe 仿真论坛与读者交流互动，读者可及时反馈学习过程中遇到的问题。
- + 异步社区提供本书配套程序源代码及 CAE 模型，供读者下载学习使用。

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

Abaqus GUI 程序开发指南

Python 语言

贾利勇 富琛阳子 贺高 周正光 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Abaqus GUI程序开发指南 : Python语言 / 贾利勇等
编著. — 北京 : 人民邮电出版社, 2016. 3
ISBN 978-7-115-41659-9

I. ①A… II. ①贾… III. ①软件工具—程序设计—
指南 IV. ①TP311.56-62

中国版本图书馆CIP数据核字(2016)第033736号

内 容 提 要

本书共6章,主要讲解采用Python语言对Abaqus进行图形界面程序的开发,包括GUI插件程序(Plug-ins)的开发以及用户自定义GUI应用程序(Customized Applications)的开发。各章的主要内容是:第1章以简单的入门插件程序为例,介绍了Abaqus图形界面程序开发的意义。第2章简要介绍了Python语言的基本语法。第3章介绍了一些Abaqus中常用的Python内核指令。第4章详细介绍了Python语言中多种常用GUI控件的使用方法。第5章介绍了Abaqus中基于Python语言的GUI插件程序开发过程以及插件程序的管理,同时给出了多个完整的插件程序实例。第6章介绍了Abaqus中用户自定义GUI应用程序的开发过程,包括GUI工具条、工具箱、菜单、模块的创建方法及管理,同时给出了自定义GUI应用程序的开发实例,并对实例做了详细的注释。

本书适合高校、研究所以及工程设计单位的Abaqus中高级用户学习使用,要求读者对Abaqus有限元软件有一定的认识和操作能力。

-
- ◆ 编 著 贾利勇 富琛阳子 贺 高 周正光
责任编辑 王峰松
责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京天宇星印刷厂印刷
 - ◆ 开本: 720×960 1/16
印张: 16.25
字数: 297千字
印数: 1-2 500册
- 2016年3月第1版
2016年3月北京第1次印刷
-

定价: 59.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

序 言

认识贾利勇先生大约是在 2010 年初，那时的他还是个年轻小伙子，刚参加工作没几年，对 CAE 分析有着浓厚的兴趣。之后的几年里在 SIMULIA 的研讨会以及年会上见过几次面，期间多以邮件联系。贾利勇先生多年以来一直从事飞机复合材料结构设计及强度分析方面的工作，鉴于复合材料结构前后处理过程过于繁杂，于是也开展一些复合材料结构 GUI 工具开发方面的工作。贾利勇先生在平时的学习和工作中，有随手记录的习惯，用他自己的话说“这是一种笨拙的学习方法”，将新的知识点逐一记录下来以免过后忘记，当积累的知识点多了就进行分类整理，久而久之，一本书的雏形就自然而然形成了。无论是何种原因，这种知识的积累、总结与传承都是值得学习与尊重的。

Abaqus 作为世界上最先进的大型通用非线性有限元软件之一，在世界范围内具有广泛的知名度和用户群，其分析结果具有极高的可信性和说服力。被广泛应用于机械制造、航空航天、汽车、船舶、土木工程、能源矿产、材料工程、医疗卫生以及家用电器等诸多领域。随着软件应用的不断深入，越来越多的用户开始基于 Abaqus 的平台开展二次开发，以满足各行业特定的功能需求。目前，在国内的图书市场上，尚没有系统介绍 Abaqus GUI 程序开发方面的书籍，相信本书的出现会为用户学习 Abaqus GUI 程序开发提供便捷的条件。

从内容上看，本书基本涵盖了 Abaqus GUI 程序开发的要点，同时给出了丰富的控件使用、插件程序开发以及用户自定义应用程序开发实例，完全可以用于指导 Abaqus 初学者及中高级用户开展 GUI 程序开发方面的工作。Abaqus 是一个开放性的软件平台，GUI 程序开发是对软件功能的拓展与升华。希望本书的出版，能够帮助越来越多的用户更准确、更便捷地解决好科研以及工程应用问题。

DS SIMULIA 高级技术经理

龚平 博士

2015 年 11 月

前 言

Abaqus 是目前国际上比较先进的大型通用有限元软件之一，它可以分析复杂的工程力学问题，其驾驭庞大求解规模的能力以及非线性力学分析功能均达到世界领先水平。Abaqus 在欧洲、北美和亚洲许多国家得到了广泛的应用，其用户遍及航空、航天、船舶、机械、化工、冶金、土木、水利、材料、汽车、电器等各个工程和科研领域。

Abaqus 强大的非线性求解器、清晰友好的用户界面以及开放的开发平台使得近年来其用户数量迅速增长，同时随着用户软件使用能力的提高，对 Abaqus 提出了更高的使用要求。借助 Abaqus 软件为用户提供的二次开发接口，越来越多的人开始学习和使用 Abaqus 的二次开发，研究适用于用户自己的算法、用户单元、材料本构模型以及用户自定义的 GUI 工具等。

Abaqus 二次开发分为子程序开发和用户图形界面程序（Graphical User Interface, GUI）开发两类：子程序开发基于 Fortran 语言，用户可以根据实际需求编写材料本构关系（UMAT/VUMAT）、自定义单元 UEL 以及其他多种子程序（UVARM、DLOAD、USDFLD 等）；用户图形界面程序开发基于 Python 语言，用户可以根据需求对原有 Abaqus/CAE 界面程序进行扩展，开发专用的前后处理模块以及 GUI 工具等。

目前，国内针对基于 Python 语言的 Abaqus 二次开发方面的书籍只有曹金凤老师的《Python 语言在 Abaqus 中的应用》一书，该书详细介绍了 Python 语言的基本语法、Abaqus 脚本接口、参数化建模指令、脚本的高级处理功能等，并给出了很多前后处理脚本实例，为读者学习 Abaqus 脚本编写提供了很大的帮助。

本书侧重于基于 Python 语言的 Abaqus 图形界面程序开发，主要包含 GUI 插件工具（Plug-ins）开发以及自定义 GUI 应用程序（Customized Applications）开发，用户可以根据实际需求开发 GUI 插件工具或者自定义 GUI 应用程序，从而实现高效参数化建模、标准工具箱或者模块的建立，一方面可大大减少前后处理时间，另一方面可有效避免反复手动操作引入的建模错误，提高前后处理效率。本书共 6 章内容：第一章以简单的入门插件程序介绍了 Python 语言在 Abaqus GUI 插件程序开发中的应用；第二章与第三章简要介绍了 Python 语言的基本语法和 Abaqus 常用的内核指令，因为本书的侧重点不在这方面，所以有关 Python 基本语法和 Abaqus 常用内核指令方面的

内容请参照 Abaqus 帮助文档或者曹金凤老师的《Python 语言在 Abaqus 中的应用》一书；第四章主要介绍了 Python 中常用控件的使用方法以及实例；第五章主要介绍了基于 Python 语言的 GUI 插件程序开发过程以及多插件程序的管理，同时给出了完整的插件程序实例；第六章介绍了用户自定义 GUI 应用程序的组成以及 GUI 工具条、工具箱、菜单、模块的创建方法等，同样，在本章给出了自定义 GUI 应用程序的开发实例，并对实例做了详细的注释。

读者对象

本书主要面向 Abaqus 软件的中级和高级用户。在开始学习 Abaqus 图形界面程序开发之前，读者应熟练掌握 Abaqus 有限元分析软件的基础知识，熟悉 Abaqus/CAE 的操作界面，了解 Python 语言的基本编码格式和语法规则。Python 语言的基本语法以及 Abaqus 中常用的函数库可参照 Abaqus 帮助文档，本书中将不再赘述。

本书特色

- 本书从 Abaqus 的实际开发应用出发，结合作者的开发和应用实践，深入浅出地介绍了 Abaqus 用户图形界面程序开发的方法，文字通俗易懂，读者可轻松地掌握 Abaqus GUI 程序开发的编程技巧。
- 本书详细介绍了多种 Python 常用控件的使用方法和注意事项，帮助读者少走弯路。
- 本书中详细介绍了 GUI 插件程序以及自定义 GUI 应用程序的组成、开发过程和方法，给出了完整的 GUI 插件实例和 GUI 应用程序实例，并对实例代码做了详尽的注释，以方便读者理解。
- 为了方便读者学习，本书的配套程序源代码及 CAE 模型均可以从人民邮电出版社的异步社区下载，网址为：www.epubit.com.cn/book/details/4252。
- 本书中 Python 语言选用的是 2.6.2 版本，Abaqus 软件采用的是 6.10 版本。

致谢

- 特别感谢中航工业第一飞机设计研究院结构设计研究所各位领导、同事多年来在工作以及生活中给予的关心和帮助。
- 特别感谢导师贺高研究员多年来在学习和工作中的悉心指导以及对本书的审校和修改。感谢师妹富琛阳子、师兄周正光对书稿的校对、修改以及在本书出版过程中所付出的努力。

- 特别感谢达索 SIMULIA 高级技术经理龚平博士，达索 SIMULIA 北京办事处张涛、焦中华、原中晋老师，中航信息技术中心许鸿杰老师，安托信息技术有限公司范艳杰老师，重庆大学赵友选老师多年来在 Abaqus 软件应用方面的支持与帮助。
- 在本书的编写过程中，作者还参考了一些其他关于 Python 以及 Abaqus 应用的书籍，同时大量参考了 Abaqus 6.10 帮助文档，感谢相关作者的辛勤劳动。

目前，国内尚没有一本系统介绍 Abaqus GUI 图形界面程序开发的书籍，可参考资料甚少，作者尽最大能力将内容介绍清楚，便于读者学习，但是鉴于水平有限，书中难免有错误及纰漏之处，敬请诸位专家和广大读者朋友批评指正，并欢迎通过以下联系方式与作者交流。

电子邮件: taishanbuzuo@163.com

微信: taishanbuzuo

作者

2014 年端午于阎良

目 录

第一章 概述	1
1.1 简单插件实例——创建带孔板有限元模型	2
1.2 Abaqus 图形界面程序开发的意义	10
第二章 Python 语言基础	11
2.1 Python 语言的特点	12
2.2 Python 与 Abaqus	13
2.3 Python 的编码规则	15
2.4 内置对象类型	17
2.5 动态类型简介	29
2.6 运算符	35
2.7 if 条件语句	38
2.8 while 和 for 循环	39
2.9 函数	42
2.10 模块	44
2.11 包导入	46
2.12 文件的操作	48
第三章 常用 Abaqus 内核指令	52
3.1 Abaqus/CAE 中 Python 指令的执行方式	53
3.2 内核脚本的快捷编辑方式	53
3.3 高亮显示及取消高亮显示	54
3.4 几何导入	56
3.5 创建集合	57
3.6 单元和节点重新编号	58

3.7	更改草图平面	59
3.8	创建基于单元的面	62
3.9	统计零件或实例的单元和节点数量	62
3.10	sendCommand() 命令注意问题	64
3.11	合并节点 mergeNodes()	65
3.12	elementFaces 与 elemFaces 的区别	66
3.13	by angle 的选取方式	67
3.14	使用 findAt() 选取对象	68
3.15	有关系统日期和时间的指令	69
3.16	.rpy 及 .jnl 文件输出格式控制	69
第四章	常用控件使用方法	72
4.1	GUI 图形界面的创建方式	73
4.2	目标和消息	74
4.3	按钮控件 push button 使用方法	76
4.4	单选按钮 FXRadioButton 使用方法	83
4.5	复选框 FXCheckButton 使用方法	86
4.6	文本框控件 AFXTextField 使用方法	88
4.7	表格控件 AFXTable 使用方法	90
4.8	文件选择控件中文件类型的设置	103
4.9	控件提示语设置	104
4.10	快捷键的设置	104
4.11	如何创建并调用菜单	106
4.12	颜色按钮 AFXColorButton 使用方法	107
4.13	飞出按钮的使用方法	107
4.14	列表控件 AFXList 使用方法	108
4.15	树列表控件 FXTreeList 使用方法	109
4.16	带复选框的树控件 AFXOptionTreeList 使用方法	112
4.17	自定义下拉框控件 AFXComboBox 使用方法	113
4.18	进度条的使用方法	116
4.19	图标的创建及使用方法	120

4.20	Pick Button 的创建及使用方法	123
4.21	提示框的使用方法	125
4.22	单输入函数 getInput() 使用方法	130
4.23	多输入函数 getInputs() 使用方法	132
4.24	警告反馈函数 getWarningReply() 使用方法	133
4.25	在自定义窗体中调用 Abaqus/CAE 内嵌功能组件	133
4.26	工具条 Toolbar 和工具箱 toolbox 的区别	135
4.27	控件的格式及位置	136
4.28	控制控件的使用状态	139
4.29	调整对话框尺寸	145
4.30	行列布局管理器	145
第五章	插件程序开发指南	148
5.1	插件的创建方法	149
5.2	插件的保存方式	149
5.3	插件的使用方法	150
5.4	插件程序的组成	150
5.5	插件程序的管理	150
5.6	插件程序实例一：复合材料冲击损伤有限元模型自动建模程序	154
5.7	插件程序实例二：六边形蜂窝自动建模程序	184
第六章	用户自定义 GUI 应用程序开发指南	196
6.1	用户自定义 GUI 应用程序与插件程序的区别	197
6.2	用户自定义 GUI 应用程序组成	197
6.3	如何修改 Abaqus/CAE 标准应用程序	203
6.4	自定义 GUI 工具包 AFXToolsetGui	205
6.5	自定义 GUI 模块 AFXModuleGui	208
6.6	自定义 GUI 树形工具条 TreeToolsetGui	211
6.7	用户自定义 GUI 应用程序实例详解	213
参考文献		231
附录	本书中用到的对象及方法	232

第一章 概述



工欲善其事，必先利其器。对于CAE分析工程师而言，一个好的软件平台就是一把利器，基于软件平台的二次开发就是让工具更加丰富、便捷和高效。Abaqus的二次开发一般分为子程序开发和用户图形界面程序（Graphical User Interface, GUI）开发两类，有很多初学者经常把两者混淆，分不清两者的区别。Abaqus子程序开发基于Fortran语言，用户可以根据实际需求编写材料本构关系（UMAT/VUMAT）、自定义单元UEL以及其他多种子程序（UVARM、DLOAD、USDFLD等等）。用户图形界面程序开发则是基于Python语言，主要是根据需求对原有Abaqus/CAE功能组件进行扩展，开发专用的前后处理模块以及GUI工具等。本书的内容仅针对基于Python语言的GUI程序开发。在本章中，将通过一个简单的插件实例介绍Abaqus GUI程序开发的概念、作用及意义。

1.1 简单插件实例——创建带孔板有限元模型

熟悉 Abaqus/CAE 的人应该知道，当点击 Plug-ins → Toolboxes → Examples 菜单时，会弹出如图 1.1 所示工具条。

点击第一个按钮，弹出如图 1.2 所示窗口。



图 1.1 插件程序工具条

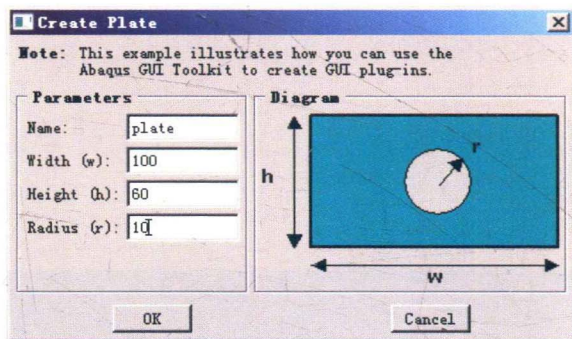


图 1.2 带孔板插件界面

这是 Abaqus/CAE 内嵌的一个简单插件程序 (plug-in)，在上述窗口中输入零件名、零件宽度、高度以及圆孔半径之后，点击“OK”，则会自动在 Abaqus 中生成带孔板的几何模型。

该程序仅能实现带孔板的几何创建，本节将以此为基础，在源程序的基础上稍加改进，使其可实现带孔板有限元分析模型的参数化建模。图 1.3 为改写之后的插件程序界面示意图。下面将对该插件程序的构成进行详细介绍。

该插件程序共由 4 个文件组成。

- (1) 注册文件 createPlateWithhole_plugin.py
- (2) 图形界面文件 createPlateWithholeDB.py
- (3) 内核执行文件 createPlateModul.py
- (4) 图标文件 planewithhole.png

其中图标文件如图 1.4 所示，图标文件即为插件窗口中的示意图，Abaqus 支持 bmp、gif、png、xpm 等多种格式的图片，本实例中采用的是 png 格式图片，其他 3 个组成文件将在后面分别介绍。

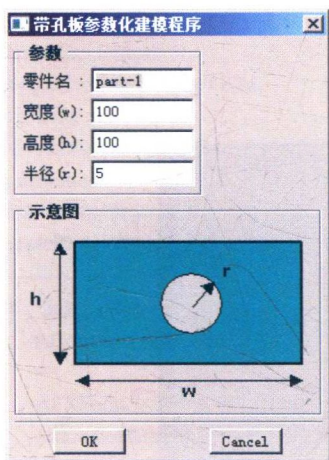


图 1.3 修改后的插件示意图

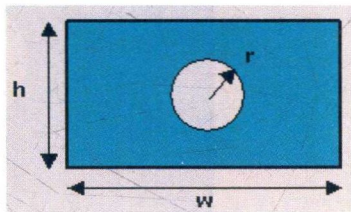


图 1.4 修改后的插件示意图

上述文件均存放在随书配套资源 \chapter 1\createPlateWithhole 文件夹内。将文件夹 createPlateWithhole 及内部文件复制到 Abaqus 工作目录或者安装目录下的 Abaqus_plugins 文件夹下（没有的话请自行创建该文件夹），例如：

安装目录 D:\SIMULIA\6.10\Abaqus_plugins\createPlateWithhole

当前工作目录 D:\TEMP\Abaqus_plugins

重新启动 Abaqus/CAE 主界面，在 Plug-ins 菜单下会出现名为“创建带孔板”的功能子菜单，如图 1.5 所示。

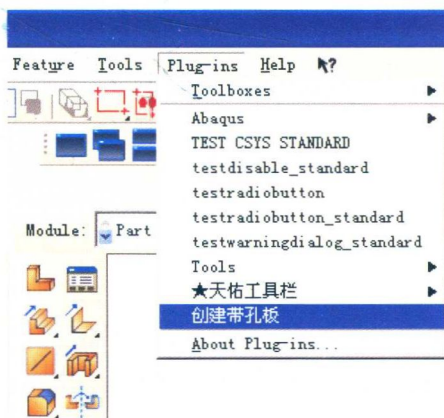


图 1.5 插件菜单

将图 1.3 中所示的参数填写完整并执行程序，可自动生成带孔板拉伸的有限元模型，并可直接将其提交计算，模型如图 1.6 所示。

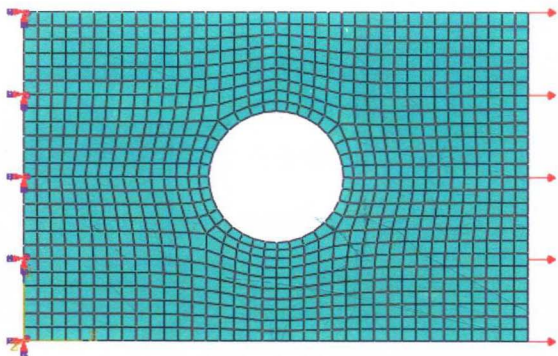


图 1.6 由插件生成的带孔板模型示意图

提交计算后，可以得到图 1.7 所示的应力云图。

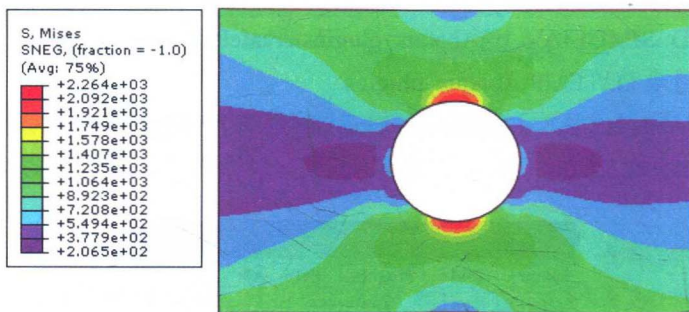


图 1.7 带孔板应力云图

1.1.1 注册文件

本插件程序中的注册文件命名为 `createPlateWithhole_plugin.py`，其主要作用是注册插件程序以及其中的关键字等。其程序源代码如下。

```
# 文件名 createPlateWithhole_plugin.py
# 中文编码
```

```
# -*- coding:UTF-8 -*-
from abaqusGui import *
from abaqusConstants import ALL
import osutils, os

#####
# Class definition
#####
class createPlateWithhole_plugin(AFXForm):
    #-----
    def __init__(self, owner):
        # Construct the base class.
        #
        AFXForm.__init__(self, owner)
        self.radioButtonGroups = {}
        self.cmd = AFXGuiCommand(mode=self, method='createPlateFunction',
            objectName='createPlateModul', registerQuery=False)
        pickedDefault = ""
        self.partnameKw = AFXStringKeyword(self.cmd, 'partname',
            True, 'part-1')
        self.widthKw = AFXFloatKeyword(self.cmd, 'width', True, 100)
        self.heightKw = AFXFloatKeyword(self.cmd, 'height', True, 100)
        self.radiusKw = AFXFloatKeyword(self.cmd, 'radius', True, 5)

    #-----
    def getFirstDialog(self):
        import createPlateWithholeDB
        return createPlateWithholeDB.createPlateWithholeDB(self)

    #-----
    def doCustomChecks(self):
        # Try to set the appropriate radio button on. If the user did
        # not specify any buttons to be on, do nothing.
        #
        for kw1, kw2, d in self.radioButtonGroups.values():
            try:
                value = d[ kw1.getValue() ]
                kw2.setValue(value)
            except:
                pass
        # 对输入数据格式进行检查，不满足要求时提示警告
        if self.widthKw.getValue()<=0:
            showAFXErrorDialog(getAFXApp().getAFXMainWindow(),
                '宽度必须为正数')
        return False
        elif self.heightKw.getValue()<=0:
```

```

        showAFXErrorDialog(getAFXApp().getAFXMainWindow(),
            '高度必须为正数 ')
    return False
elif self.radiusKw.getValue()<=0:
    showAFXErrorDialog(getAFXApp().getAFXMainWindow(),
        '孔半径必须是正数 .!')
    return False
else:
    return True

#-----
def okToCancel(self):
    # No need to close the dialog when a file operation (such
    # as New or Open) or model change is executed.
    #
    return False

#-----

# Register the plug-in
#
thisPath = os.path.abspath(__file__)          # 指定路径
thisDir = os.path.dirname(thisPath)
toolset = getAFXApp().getAFXMainWindow().getPluginToolset()
toolset.registerGuiMenuButton(
    buttonText=' 创建带孔板 ',
    object=createPlateWithhole_plugin(toolset),
    messageId=AFXMode.ID_ACTIVATE,
    icon=None,                                # 指定图标为空
    kernelInitString='import createPlateModul', # 内核初始化指令
    applicableModules=ALL,                    # 指定使用模块, 默认为全部
    version='1.0',                            # 软件版本
    author='jily',                            # 软件作者
    description='N/A',                        # 软件描述
    helpUrl='N/A'                             # 帮助文档路径指定
)

```



提示

以上代码是本书中出现的第一段代码, 需要提醒读者注意的是, 在 Python 语言中, 代码的缩进是一种语法, Python 依靠每行代码的缩进位置来判断代码的分块、区分代码块之间的层次关系, 错误的缩进位置将会导致程序执行错误, 读者应对此予以特别注意, 有关 Python 编码规则的内容请参照本书 2.3 节内容。

编写代码时建议使用 UltraEdit 或者 Notepad++ 等文本编辑软件，本书在编写过程中采用的是 Notepad++ 软件（免费），该软件支持 Python 等多种语言的语法着色。

1.1.2 图形界面文件

图形界面文件的主要作用是定义各类窗体、控件，关联控件的执行命令及对象。通过定义文本框、单复选按钮、表格等各类控件，用户可以方便地将所需参数集成到统一的图形界面下，形成友好的输入输出界面。本插件程序中的图形界面文件命名为 createPlateWithholeDB.py，其程序源代码如下。

```
# -*- coding:UTF-8 -*-
from abaqusConstants import *
from abaqusGui import *
from kernelAccess import mdb, session
import os
thisPath = os.path.abspath(__file__) # 设置当前路径
thisDir = os.path.dirname(thisPath)
#####
# Class definition
#####
class createPlateWithholeDB(AFXDataDialog):
# ~~~~~~
    def __init__(self, form):
        # Construct the base class.
        #
        AFXDataDialog.__init__(self, form, '带孔板参数化建模程序',
            self.OK|self.CANCEL, DIALOG_ACTIONS_SEPARATOR)
        # 创建对话框，该对话框有 OK、CANCEL 按钮
        okBtn = self.getActionButton(self.ID_CLICKED_OK)
        okBtn.setText('OK')
        # 定义 OK 按钮
        groupBox_1 = FXGroupBox(p=self, text='参数', opts=FRAME_GROOVE)
        AFXTextField(p=groupBox_1, ncols=12, labelText='零件名:',
            tgt=form.partnameKw, sel=0)
        # 定义文本框
        AFXTextField(p=groupBox_1, ncols=12, labelText='宽度 (w):',
            tgt=form.widthKw, sel=0)
        AFXTextField(p=groupBox_1, ncols=12, labelText='高度 (h):',
            tgt=form.heightKw, sel=0)
        AFXTextField(p=groupBox_1, ncols=12, labelText='半径 (r):',
            tgt=form.radiusKw, sel=0)
        groupBox_3 = FXGroupBox(p=self, text='示意图', opts=FRAME_GROOVE)
```