

大家好，这周我们来做一个计算器。

这次作业目的是让大家熟悉下 python 的 GUI 编程，目标：

1. 熟悉 qt Designer 的使用
2. 熟悉 pyqt5 的使用
3. 熟练 python 基本编程

作业提交：

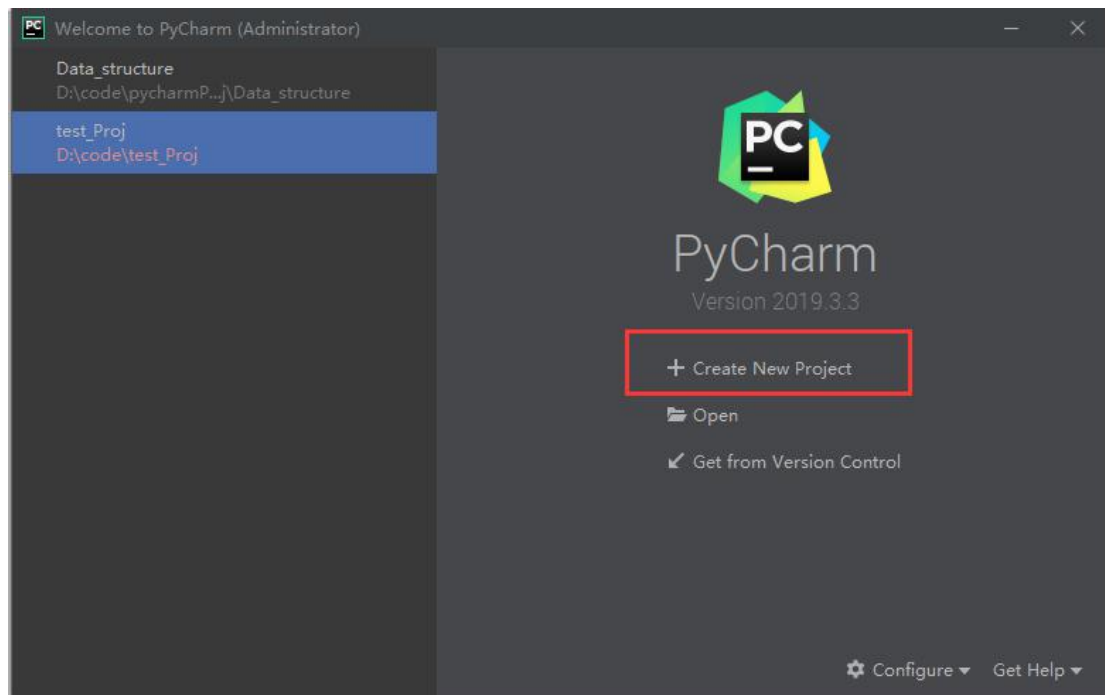
1. 作业命名方式：作业 1-学号-姓名。
2. 作业提交邮箱：910660298@qq.com
3. 截止日期：请于 3 月 13 日 24: 00 前上交压缩包
4. 上交内容：一个 word 的实验报告【详细功能（思路+实例）+核心代码】以及可以运行的项目。

先来看下我们要做的计算器的最终效果图：

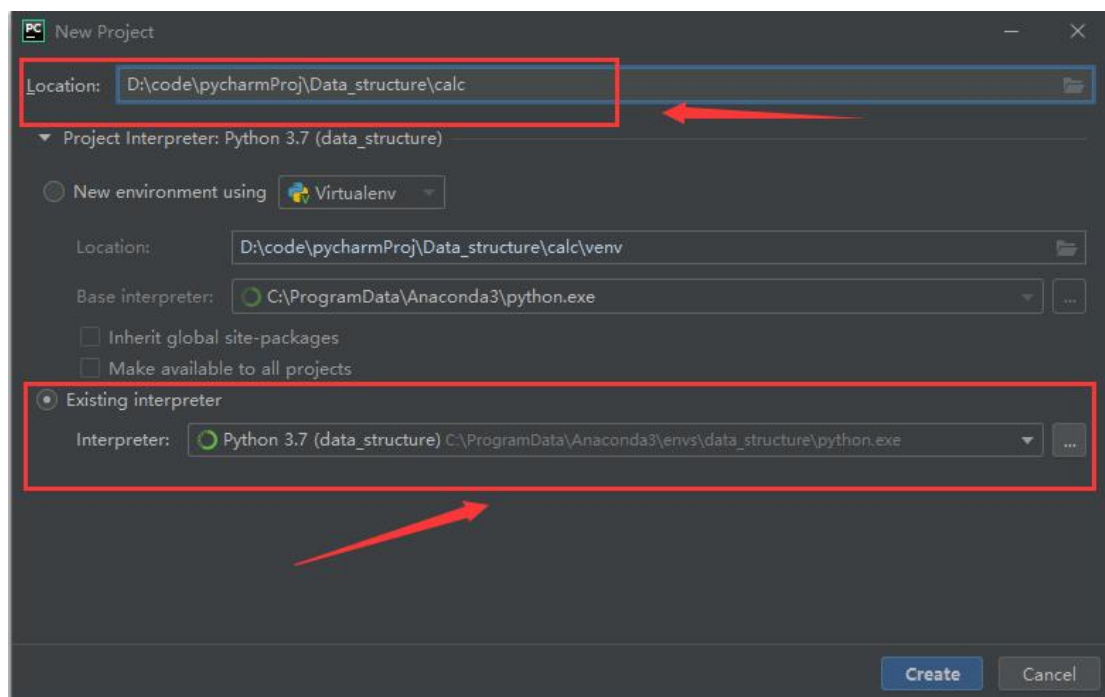


一. 实验准备

1. 打开 pycharm，新建项目



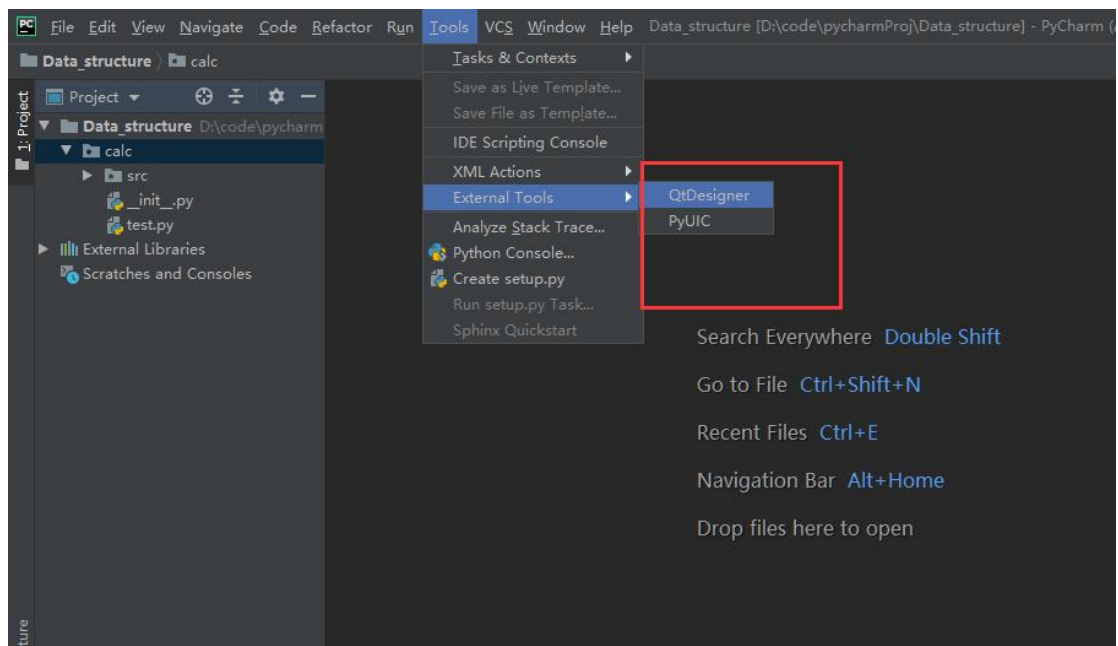
2. 填好项目所存放的路径 和 上节课新建的 anaconda 环境，点 “create”



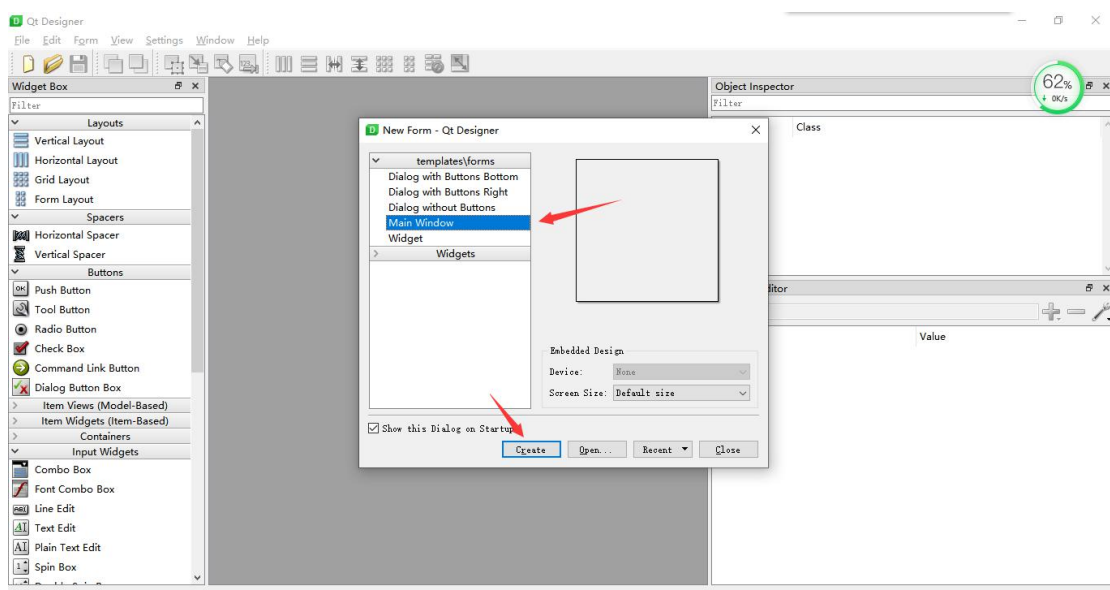
二. 实验步骤

UI 设计

1. 上节课的公告里有两篇博客，配好这两个插件：其中 QtDesigner 是打开 QtDesigner 这个 UI 设计软件，PyUIC 是将设计好的 UI 界面转化成 python 代码的工具。

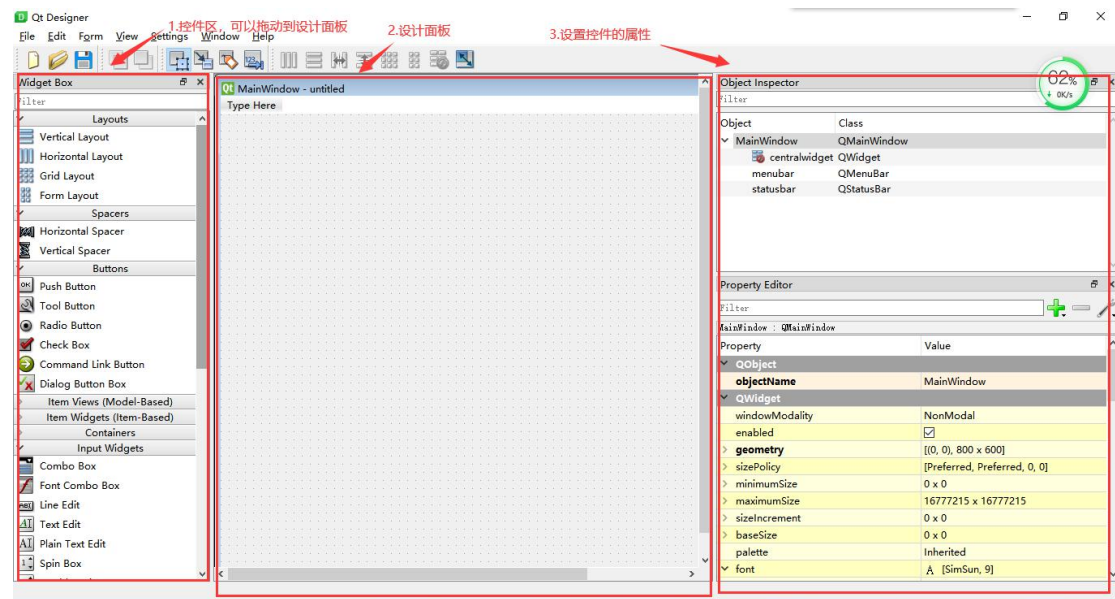


2. 点击 QtDesigner，弹出如下界面：



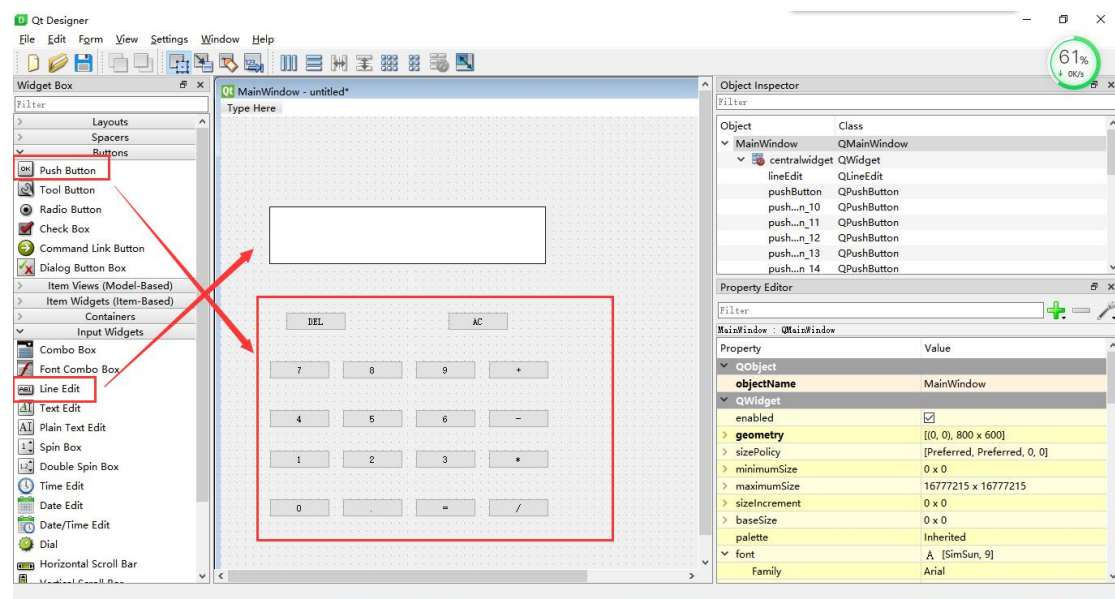
选择“Main Window”，“create”

3. 介绍下这个 Qt designer 的界面:

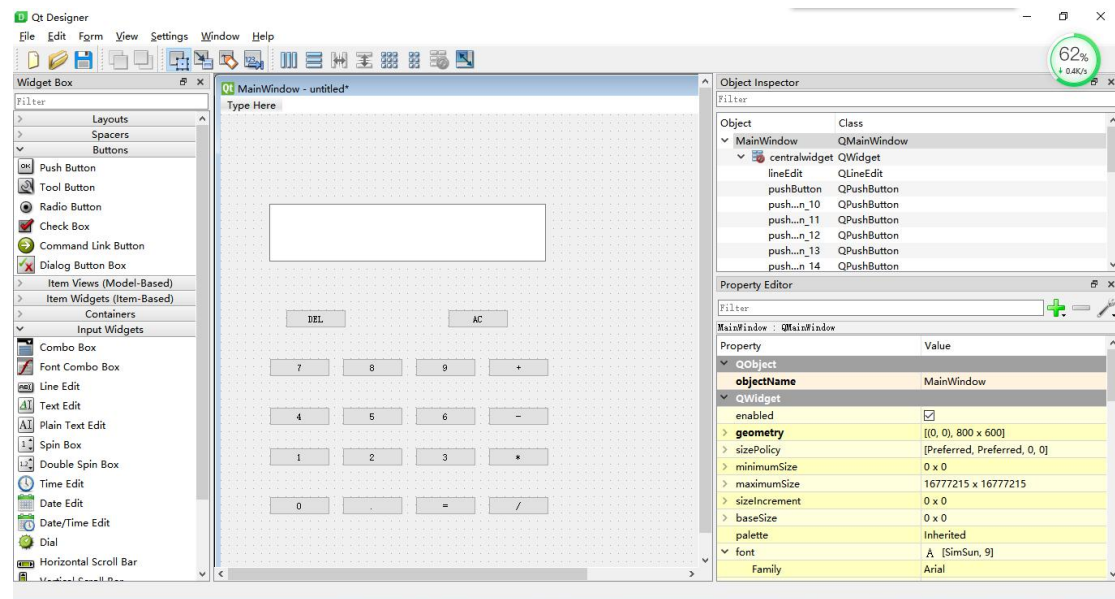


上学期学过 java swing 编程, 大致知道每个控件都是干什么用的吧。如果不知道, 把他们拖出来看看马上就能明白啦!

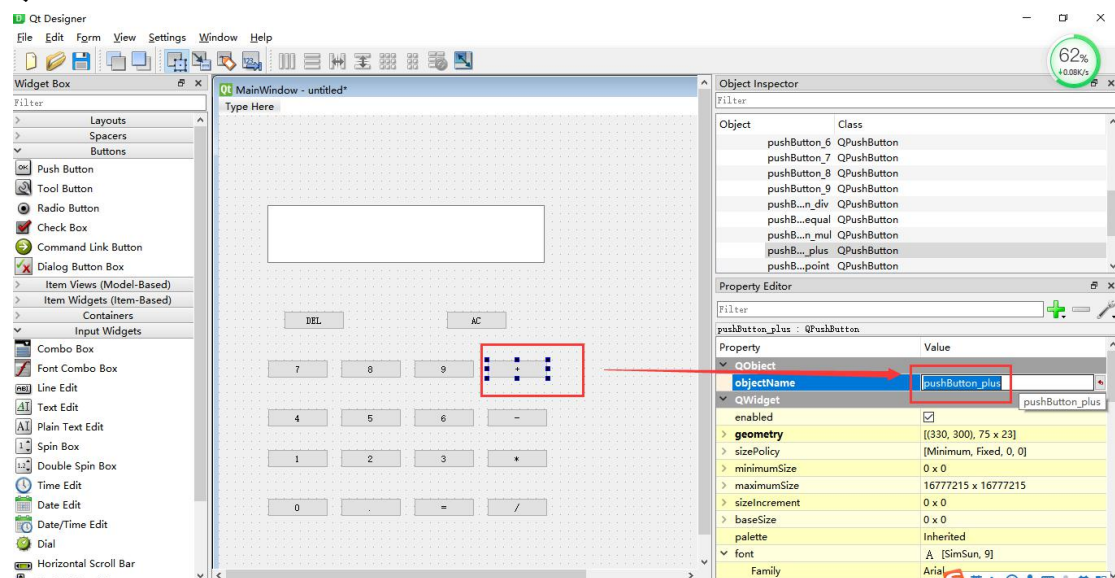
4. 按照计算器的样子, 把控件拖出来



5. 修改每个控件的上面的 text。



6. 修改每个控件上的对象名。这个对象名称是要在以后的逻辑代码中用的，起个容易的名字

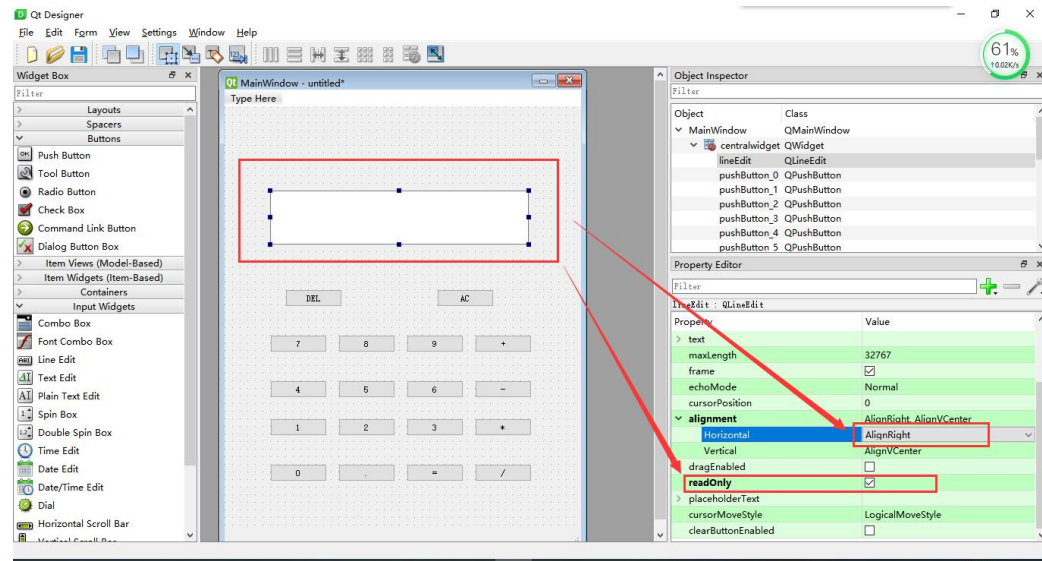


我起的名字分别是：

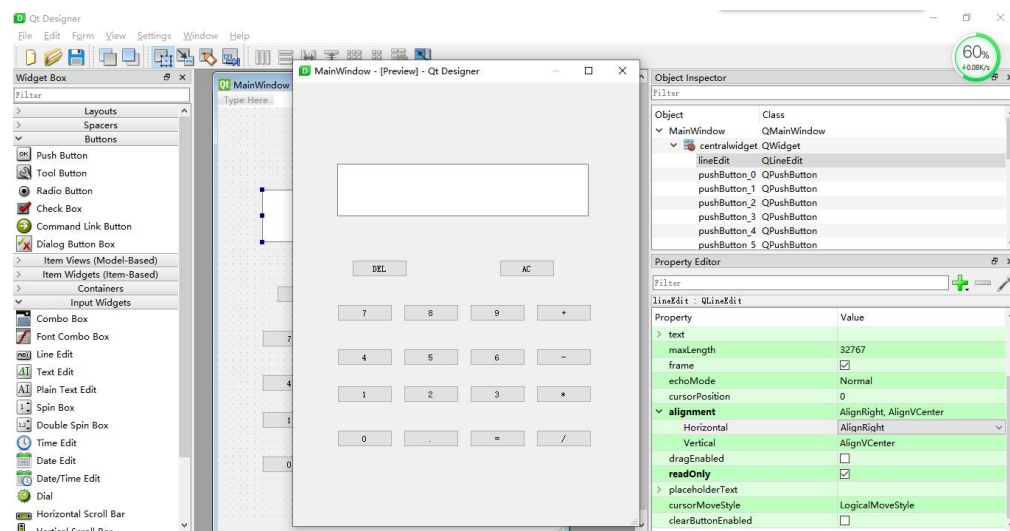
pushButton_0、pushButton_1、pushButton_2、pushButton_3、pushButton_4、pushButton_5、
pushButton_6、pushButton_7、pushButton_8、pushButton_9、
pushButton_point、pushButton_equal、
pushButton_plus、pushButton_sub、pushButton_mul、pushButton_div、

pushButton_DEL、pushButton_AC、
lineEdit

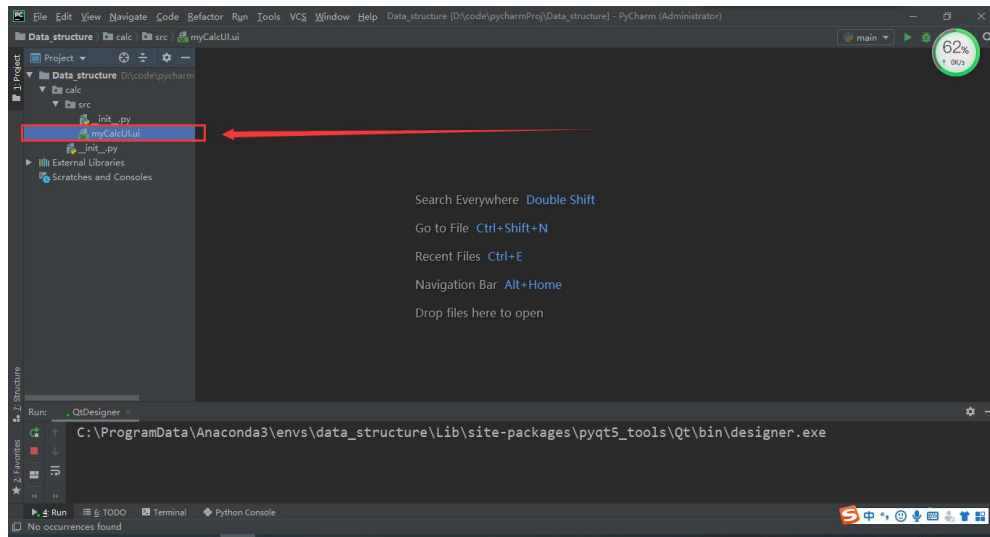
7. 计算器的显示框让他不能编辑，并且靠右显示：



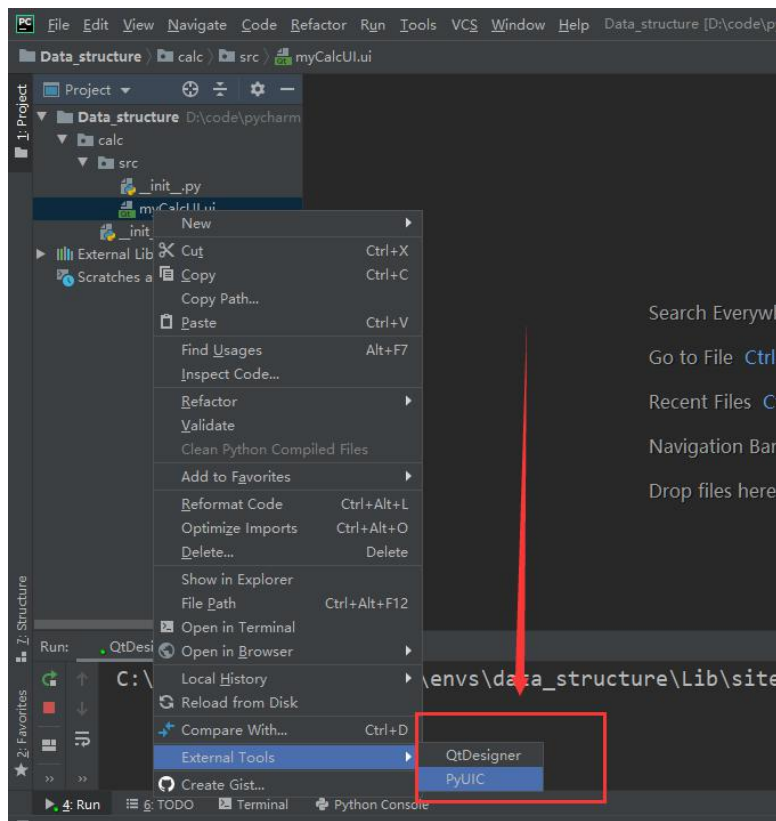
8. 按住 Ctrl+R 预览一下：

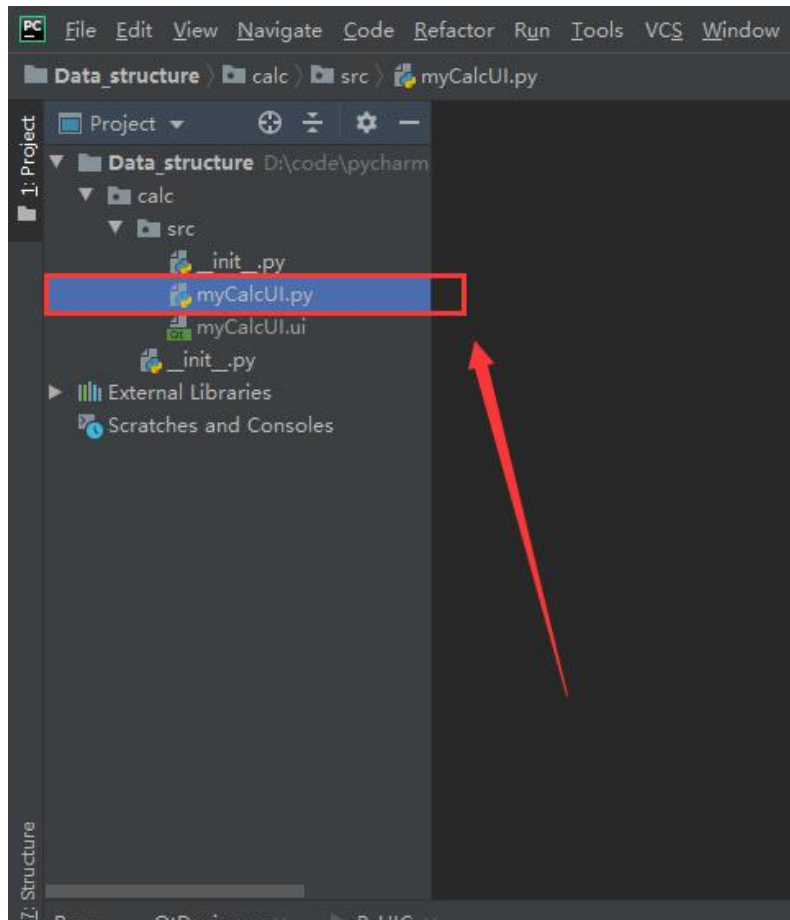


9. Ctrl+S 保存到我们项目的路径里面，文件名字自己取，这时，我们项目中会多出一个“.ui”文件



10. 右键这个.ui 文件，选择我们安装的第二个插件 PyUIC，就能把.ui 文件转换成.py 文件啦！

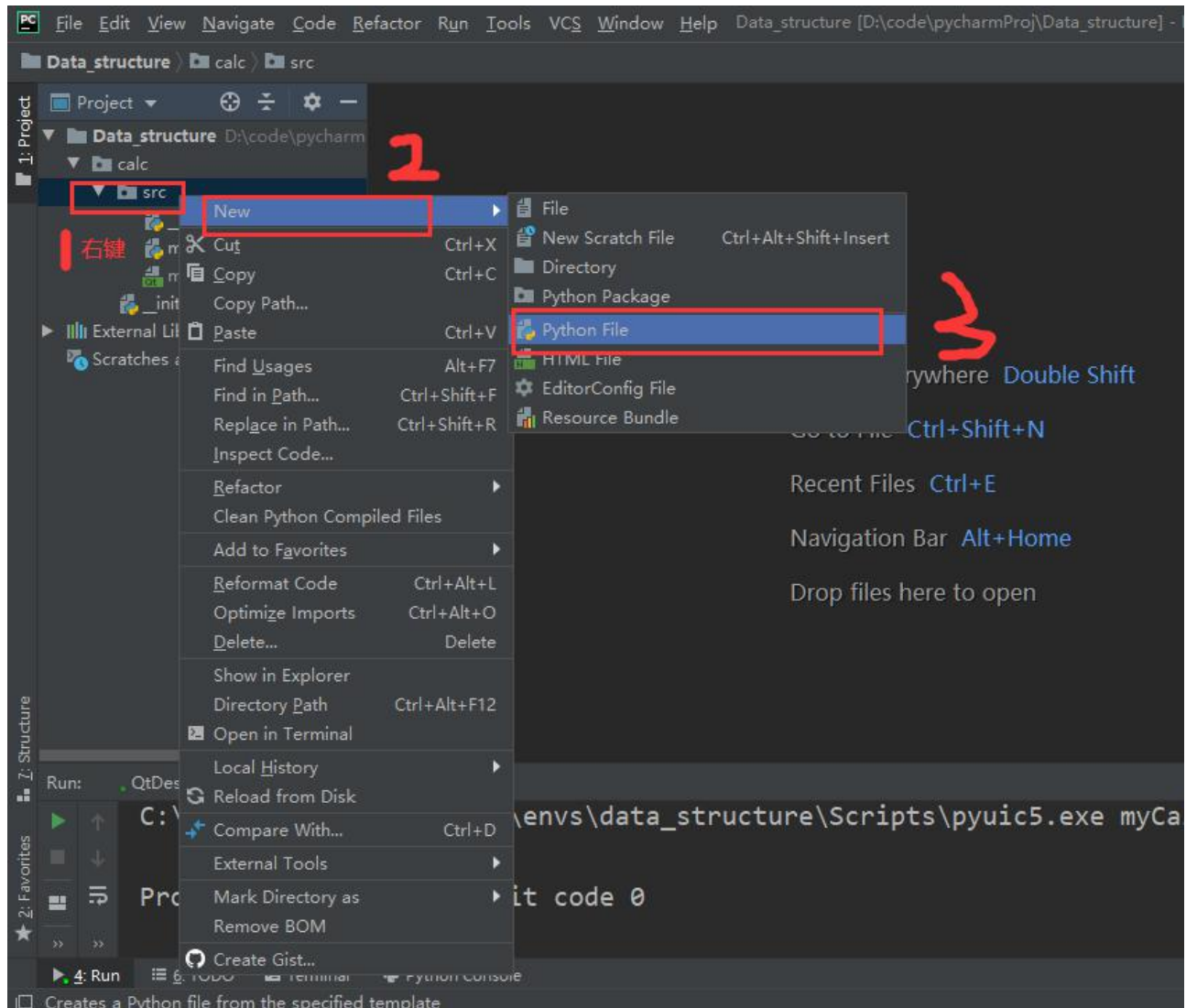




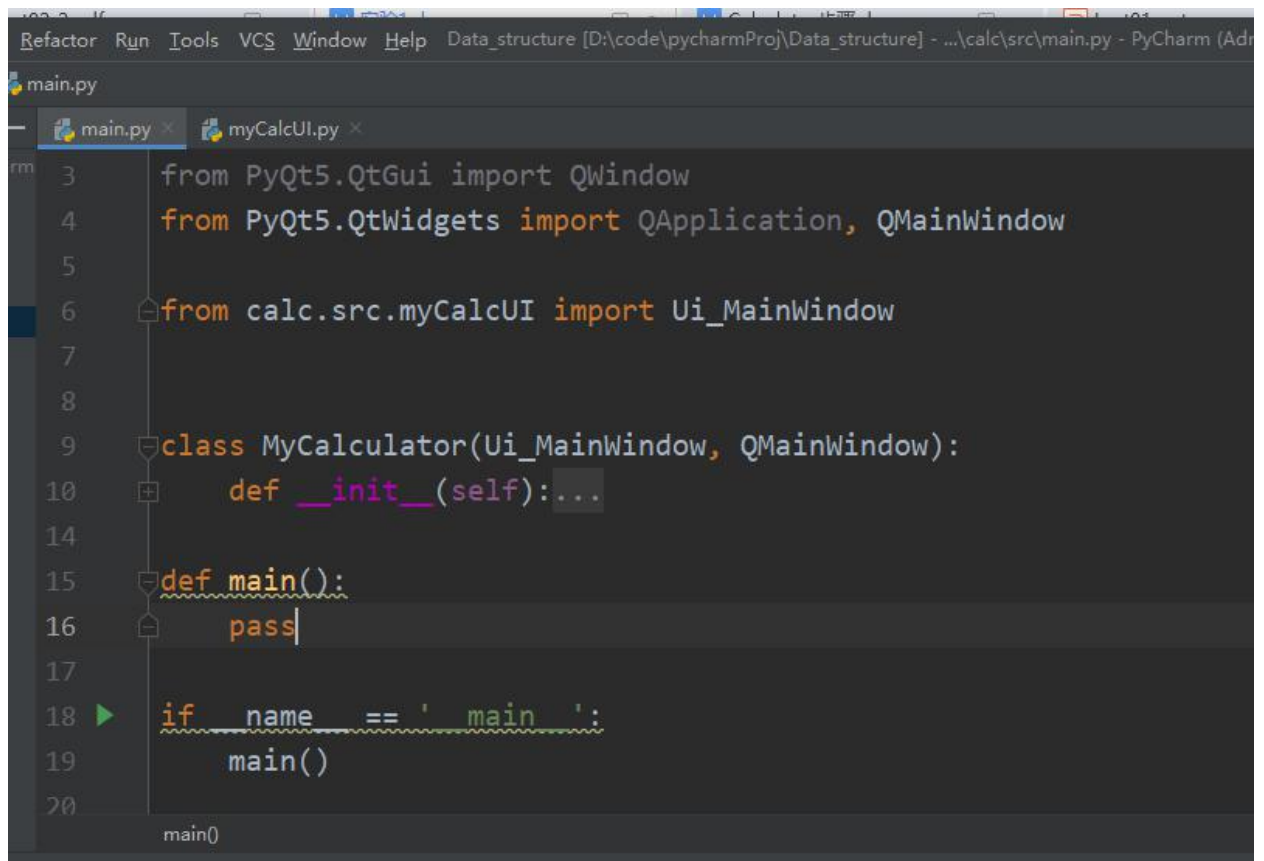
11. 这就是简易计算器的 UI 界面啦！如果要做得和我一样好看点，那你就要学 前端 **css**，这部分教程我放在附录一里面，有兴趣的同学可以看一看。以后的作业，只要功能实现，好不好看无所谓。

逻辑代码部分：

12. 我们新建一个 `main.py` 文件，在这里写逻辑代码

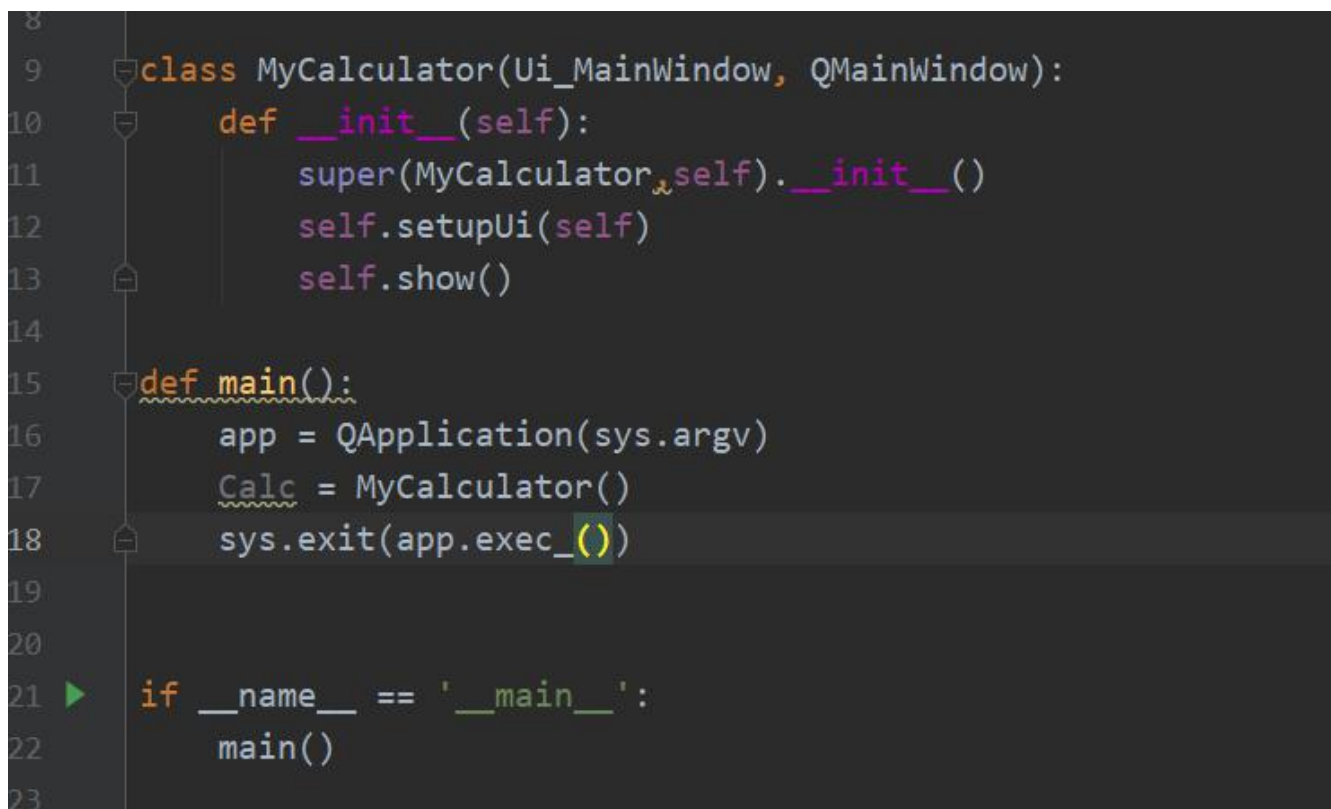


13. 写好主题框架代码：自己写一个类 `MyCalculator` 去继承我们前面通过 UI 创建出来的类。



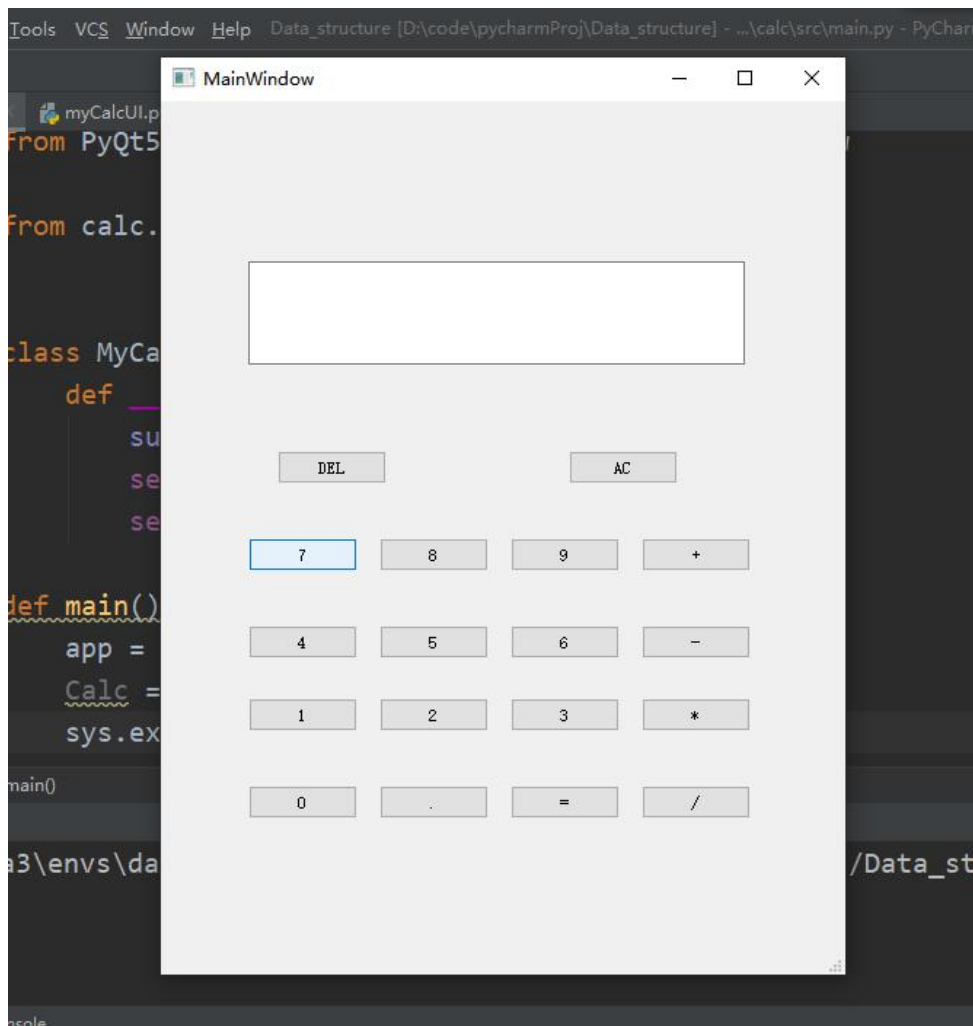
```
3 from PyQt5.QtGui import QWindow
4 from PyQt5.QtWidgets import QApplication, QMainWindow
5
6 from calc.src.myCalcUI import Ui_MainWindow
7
8
9 class MyCalculator(Ui_MainWindow, QMainWindow):
10     def __init__(self):...
11
12
13
14
15 def main():
16     pass
17
18 if __name__ == '__main__':
19     main()
20
main()
```

14. 往里面写一些基础代码，让你的窗体运行出来：

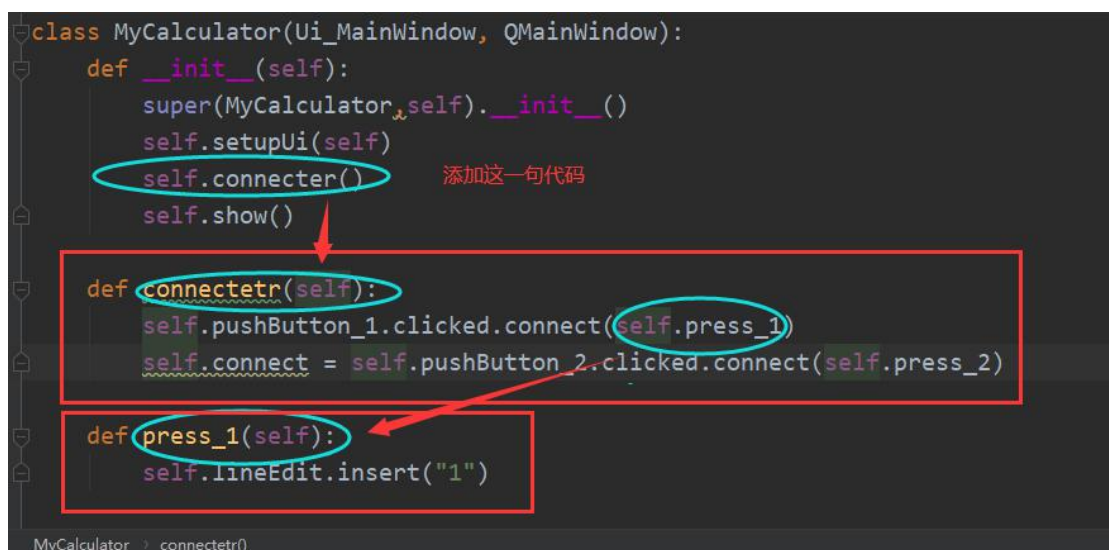


```
9 class MyCalculator(Ui_MainWindow, QMainWindow):
10     def __init__(self):
11         super(MyCalculator, self).__init__()
12         self.setupUi(self)
13         self.show()
14
15 def main():
16     app = QApplication(sys.argv)
17     Calc = MyCalculator()
18     sys.exit(app.exec_())
19
20
21 if __name__ == '__main__':
22     main()
23
```

15. Windows 按 ctrl+shift+F10 运行， mac 按 ctrl+shift+R 运行，界面出现：



16. 接下来写事件监听和处理的代码，也就是你按下每个按键，都发生一件事。比如说，你按下“1”这个按钮，让显示框出现数字“1”



除了“=”按钮，其他按钮依样画葫芦。

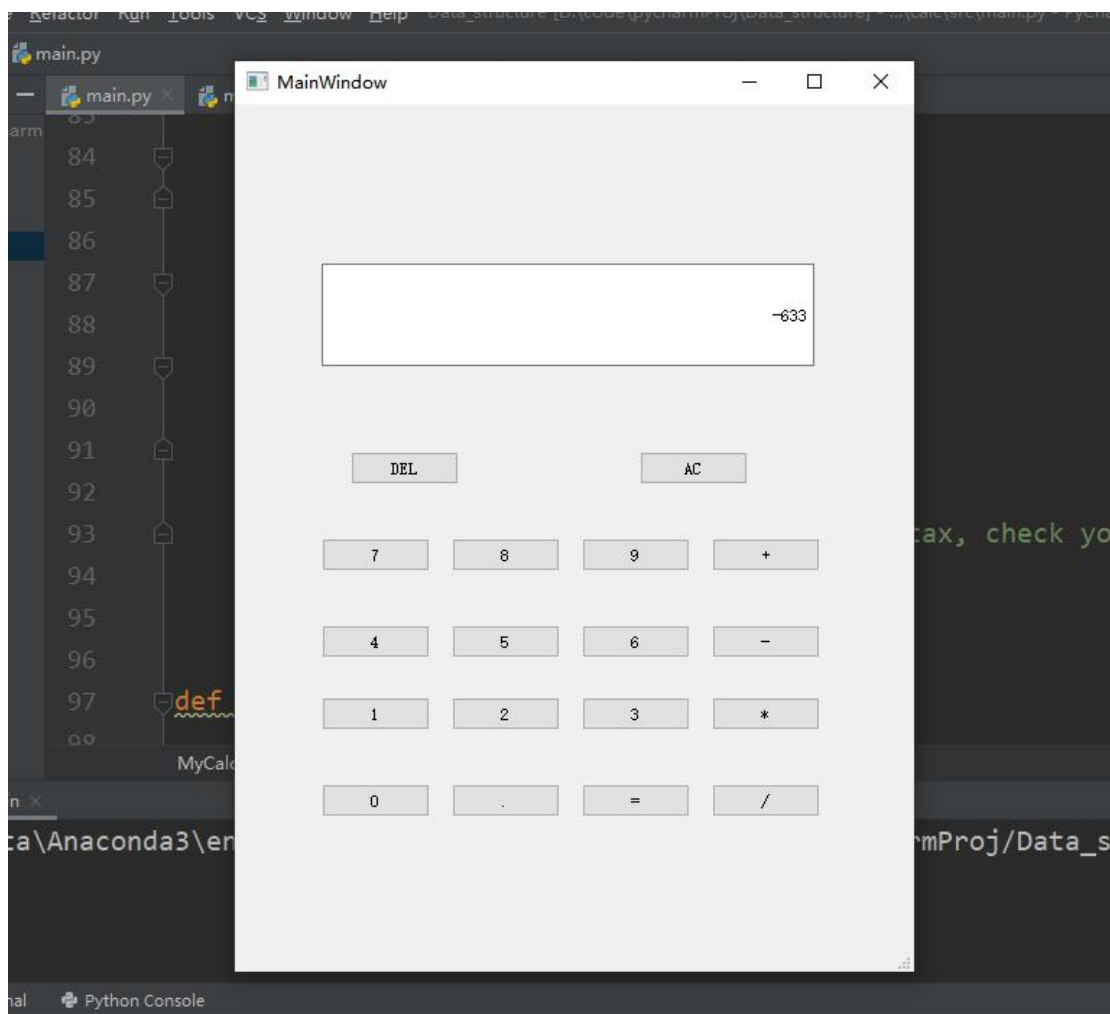
17. 最后按下“=”键的代码：

```

def press_equal(self):
    text = self.lineEdit.text()
    try:
        result = eval(text)
        self.lineEdit.setText(str(result))
    except:
        self.lineEdit.setText("Invalid syntax, check your input!")

```

18. 运行代码（同步骤 15），大功告成！

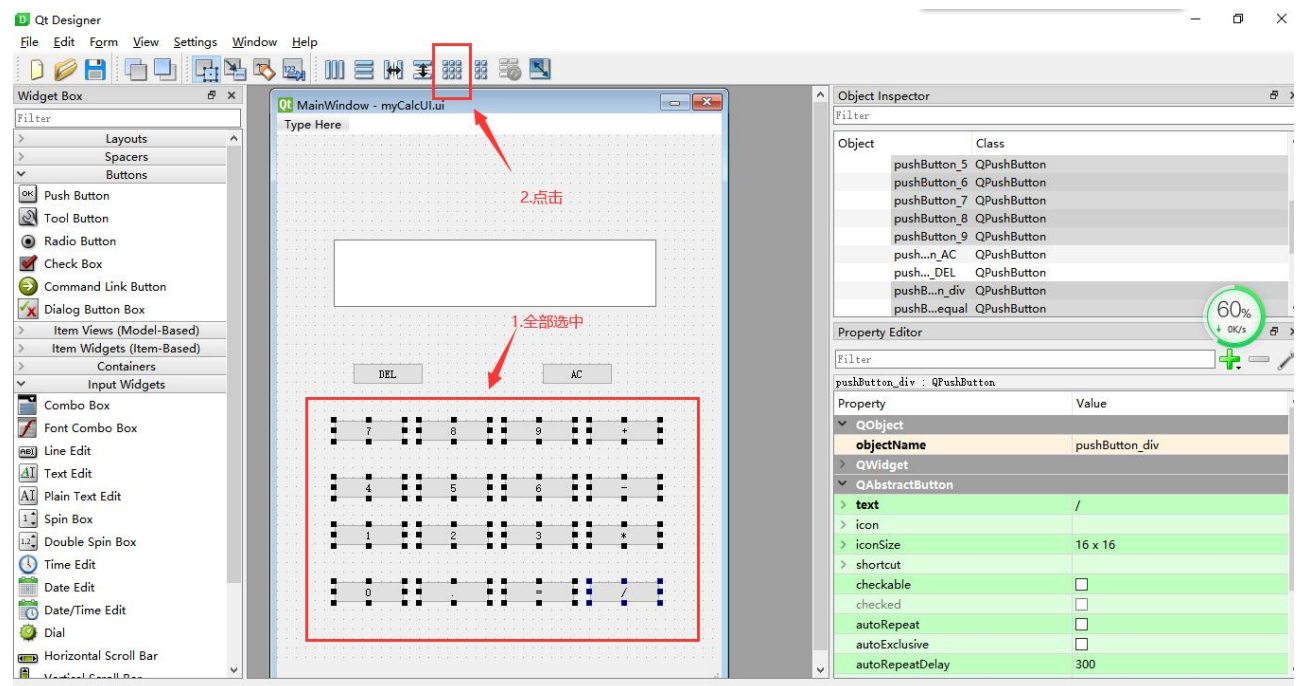


19. 作业做到这样就可以交了，明码标价，功能实现给 90 分，界面好看酌情加分。功能没实现 59 分。

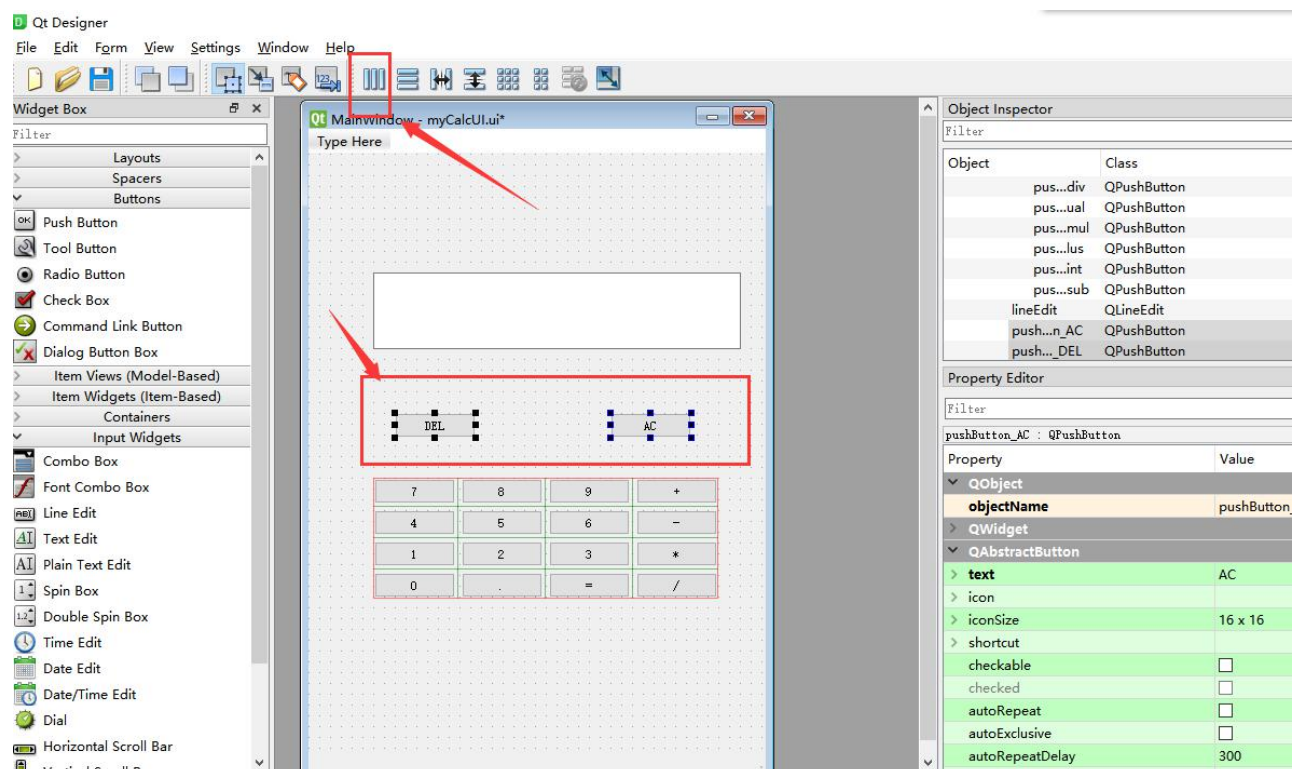
三. 附录一（使用 css 美化）

20. 有兴趣的同学看下怎么美化界面。界面美化又涉及到另外一个领域：前端设计。主要用到了 css 代码。

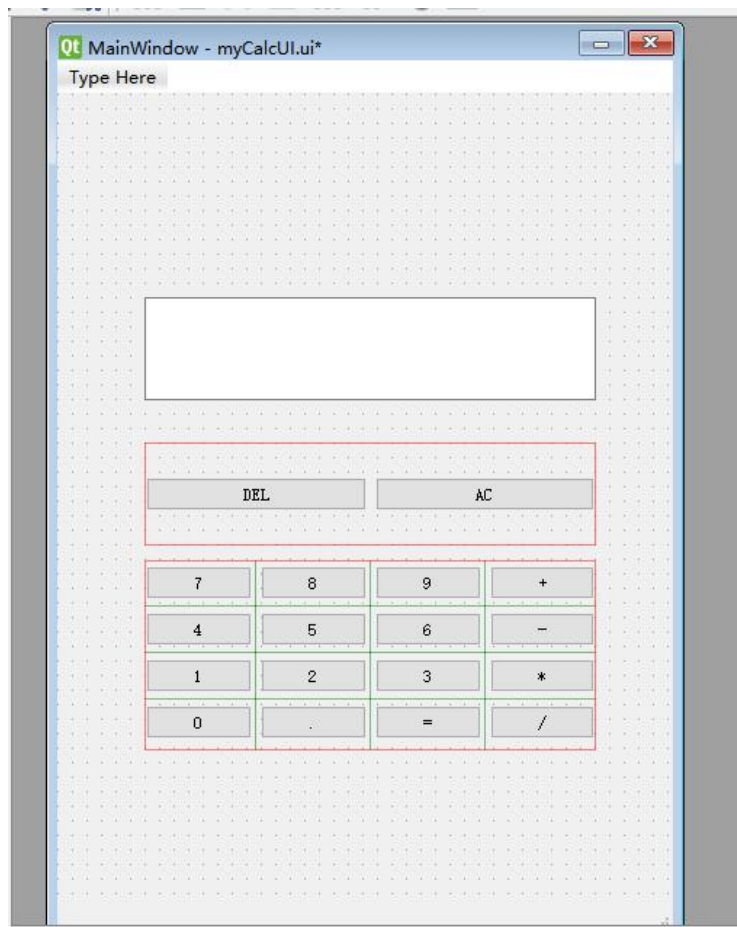
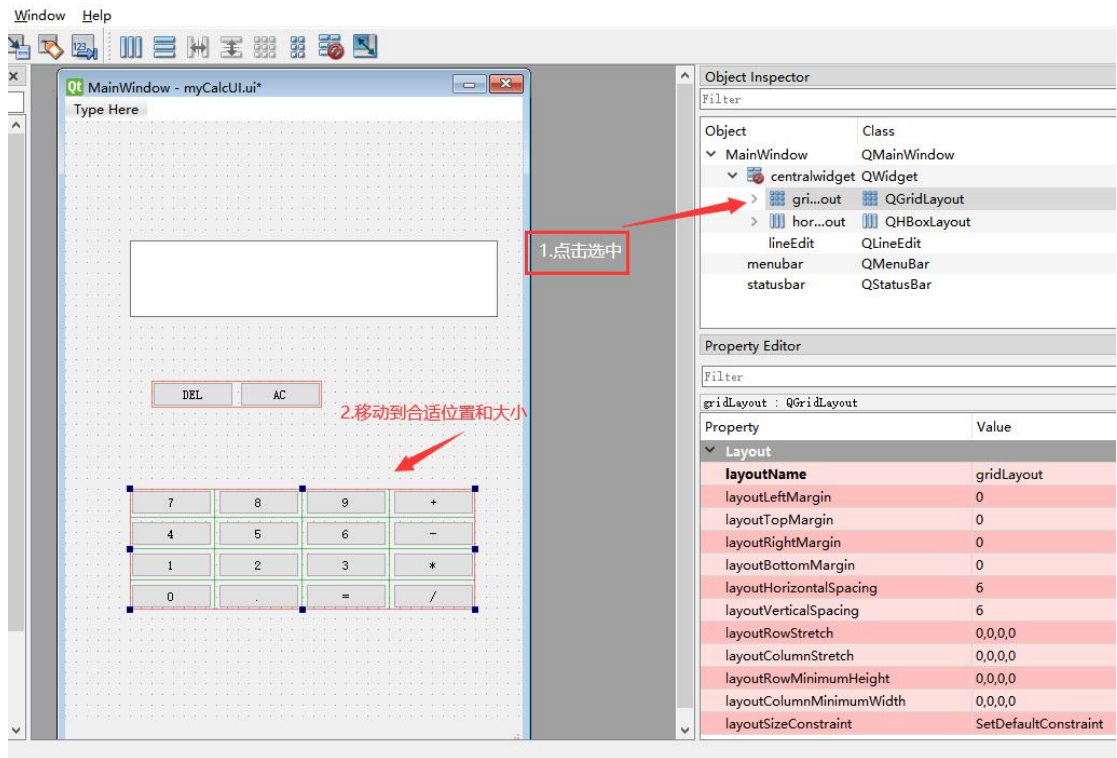
21. 首先使用布局，使界面看起来整齐。到 qtdesigner，选中以下 16 个按钮，用**网格布局**排列。



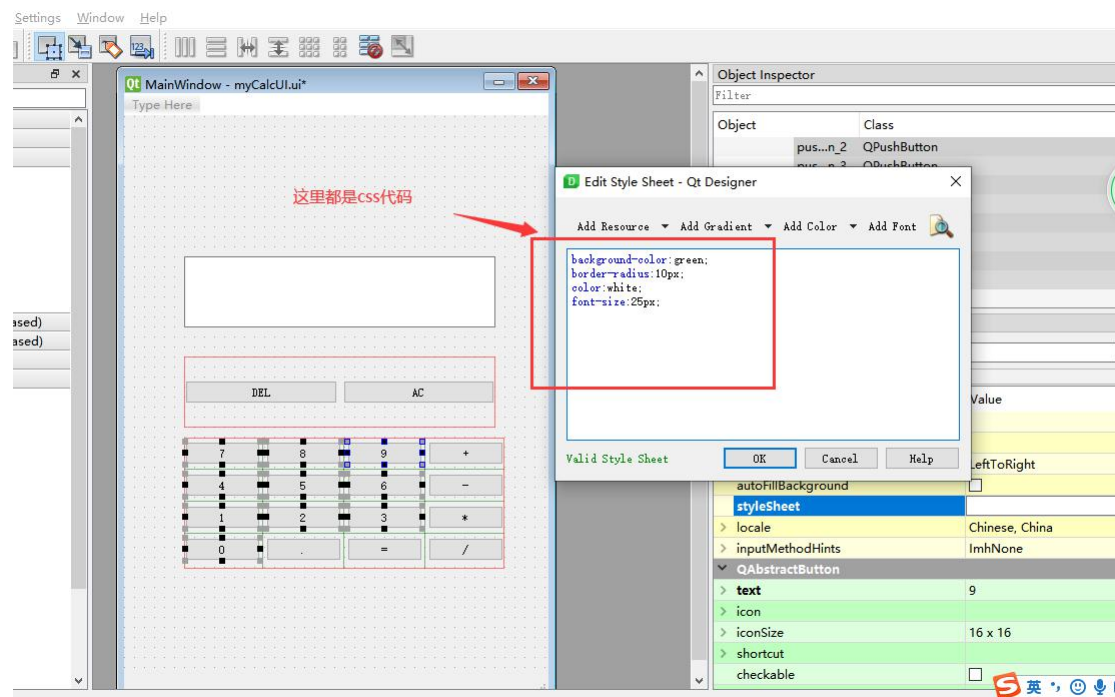
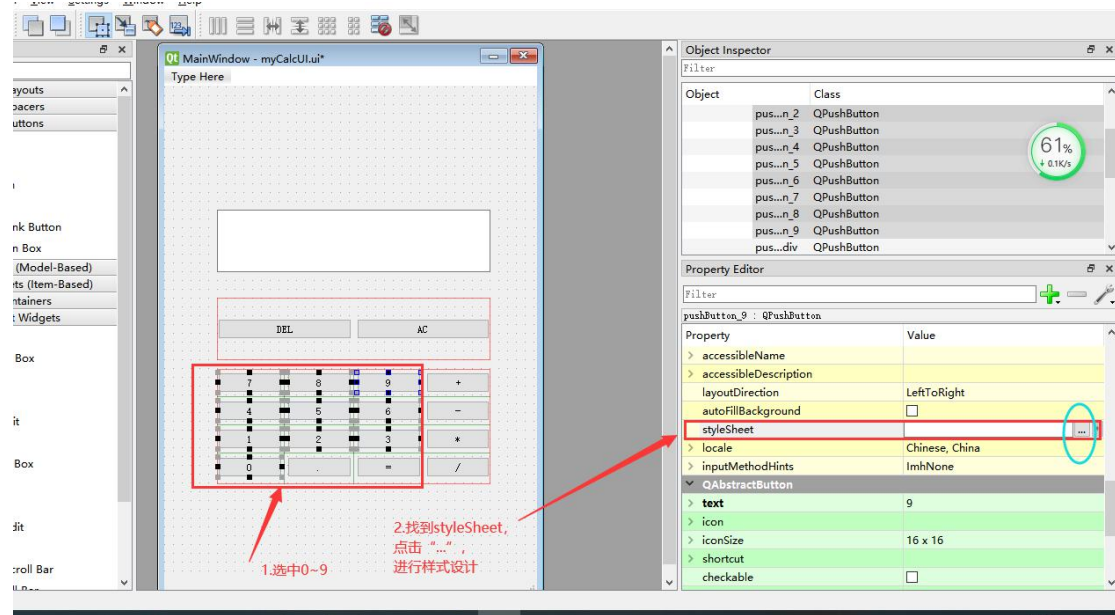
同样，DEL 和 AC 按钮也要，只不过是使用水平布局。

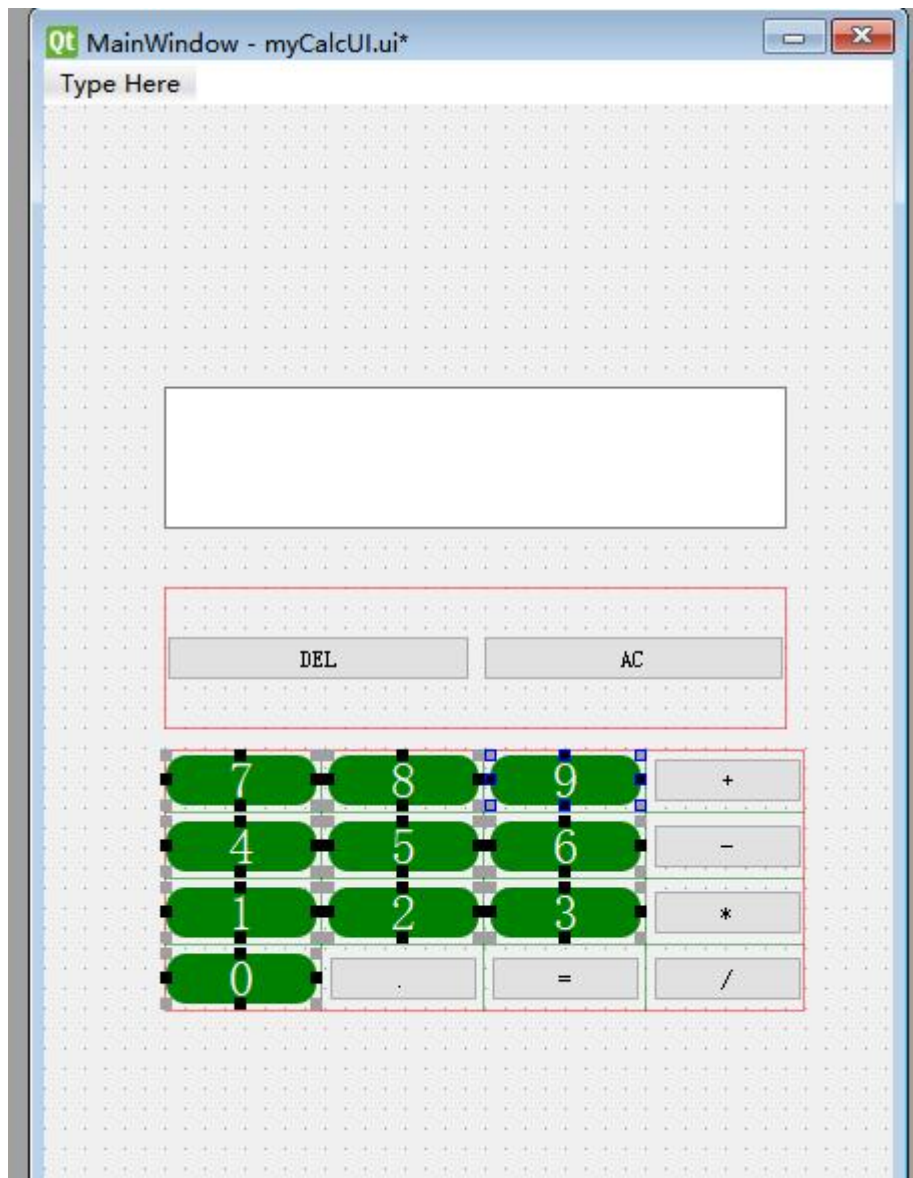


22. 整理界面



23. 设计样式





可以设计字体颜色、大小、背景颜色、边框颜色、边框粗细、色彩渐变……等着你去发掘！

四. 附录二（参考文献）

1. 【配置 QtDesigner】<https://www.jianshu.com/p/5b063c5745d0>
2. 【pyqt5 教程】<http://code.py40.com/pyqt5/16.html>
3. 【css 样式】https://www.w3school.com.cn/css/css_background.asp