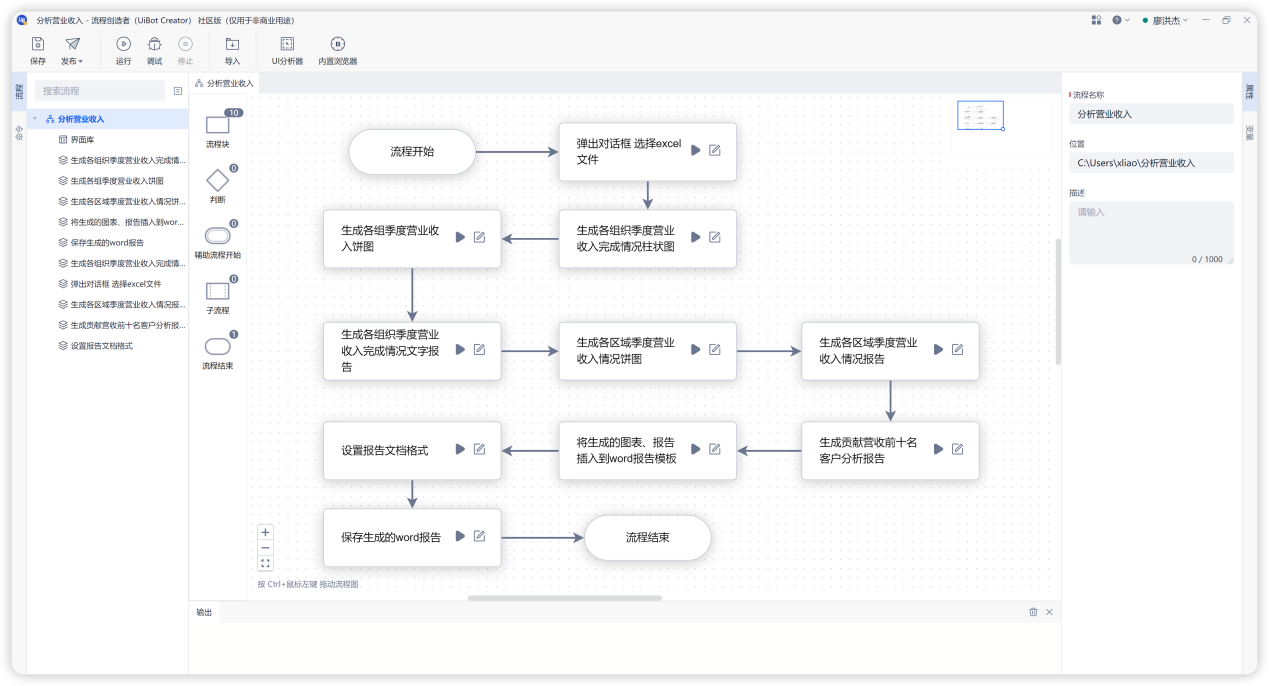
## 准备工作

本RPA对大部分的数据的操作如数据清洗、数据统计、生成图表、填写word报告等操作，均通过编写python插件的方式实现。原因是python简单易学，有许多来也RPA没有的优点，能实现来也RPA不能实现的操作。如在本RPA中，需要用到python的数据分析库pandas、生成图表库echarts，这些都是来也RPA自带组件无法实现或者实现起来非常困难的操作。

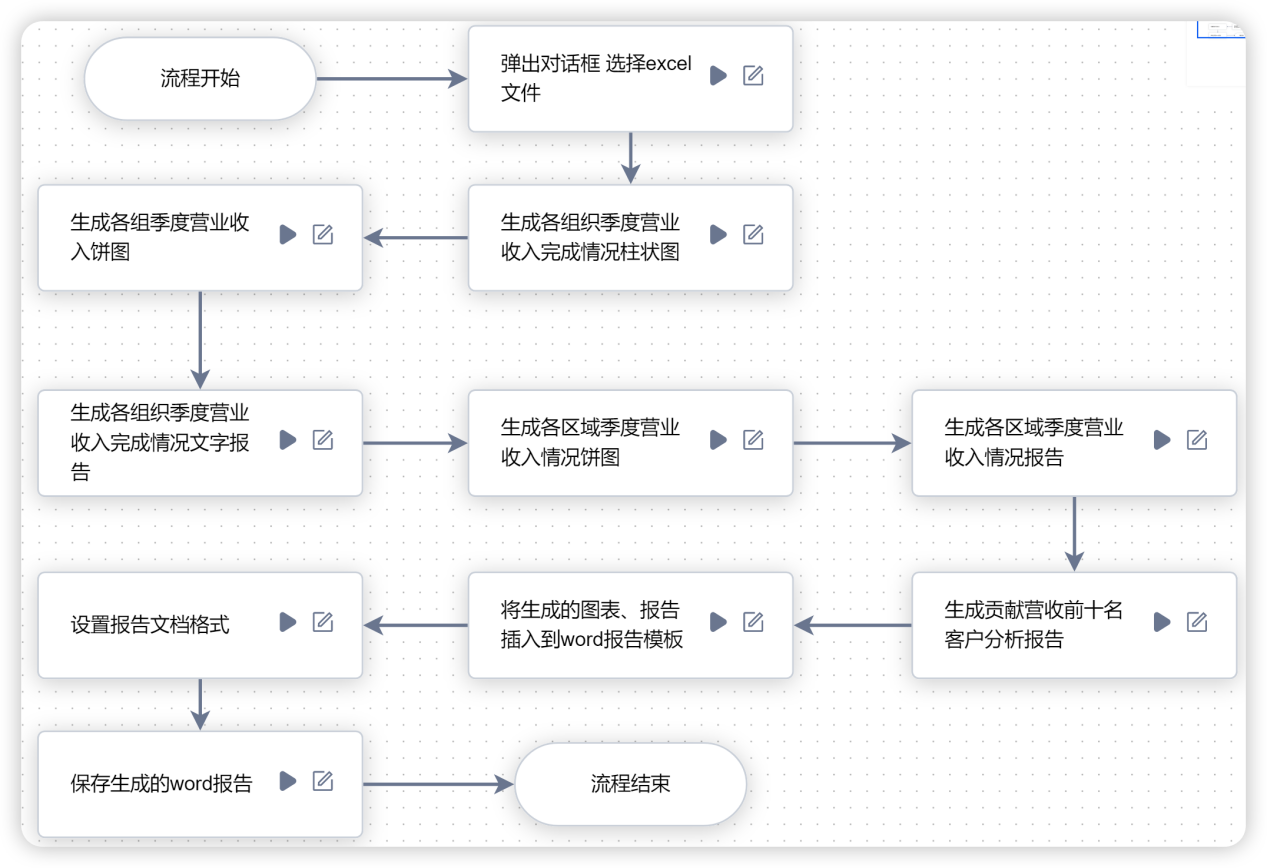
电脑上安装适合的编辑器，如VS Code，用于查看插件代码文件。VS Code下载地址： <https://code.visualstudio.com/Download>

使用来也的流程创造者打开营业收入统计分析RPA，显示界面如下

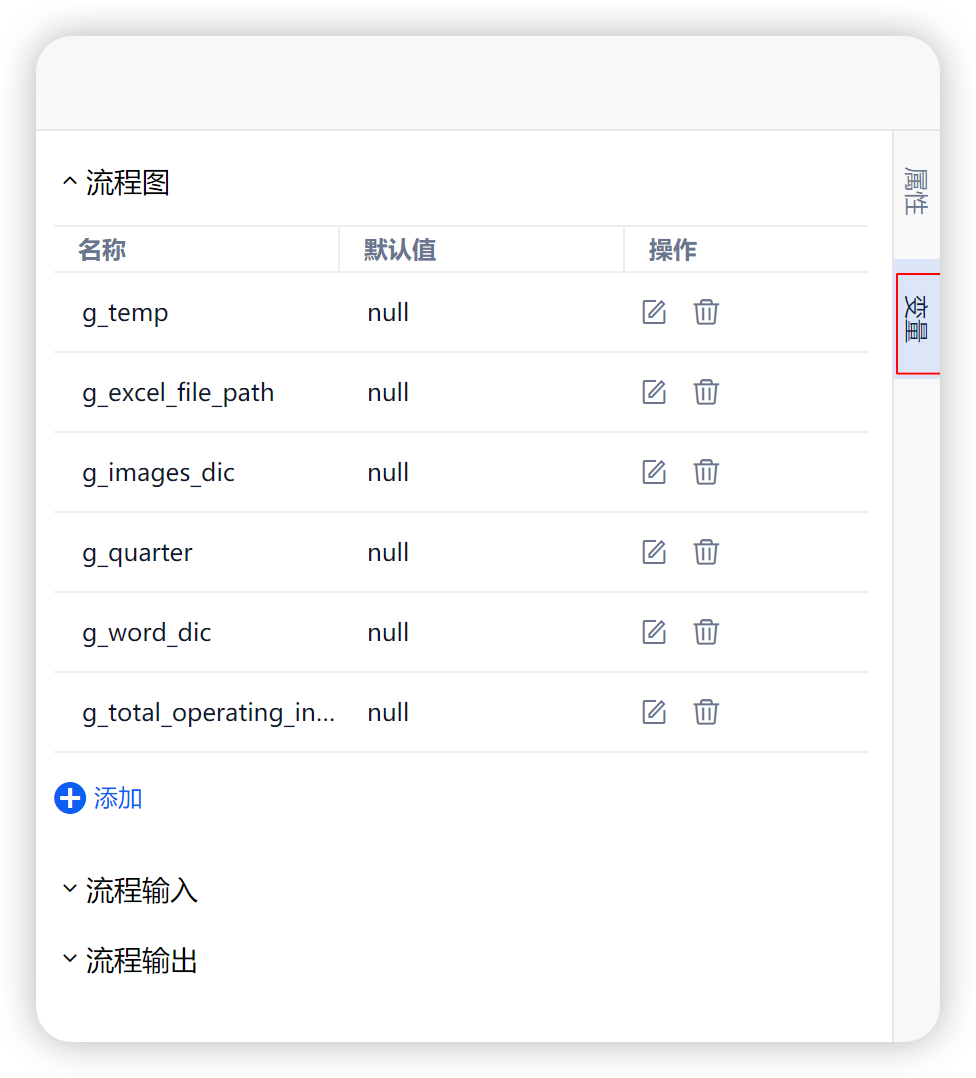


## 流程和变量讲解

流程图界面显示了整个RPA的运行流程，大致的流程是:选择excel文件->生成对应的图片->生成对应的报告->将报告插入模版文件->保存报告文件



在右侧的属性区，设置了一些流程图变量，用于各个流程块的结果传递。流程图变量在所有流程块中都可以访问到，流程块变量可以通过返回值的方式传递到下一流程块。获取到流程图变量和上一个流程块传递过来的变量数据才能是整个流程连贯的执行。



g\_excel\_file\_path 选择的excel文件路径

g\_images\_dic 设置一个字典，用于存储生成的图表路径

g\_quarter 设置一个变量，用于接收存储用户输入的报告季度信息

g\_word\_dic 设置一个文字字典，用于存储报告中的数据在后续替换模版文件里的占位符

g\_total\_operating\_income 设置一个变量，用于存储营业总收入的数据，在计算贡献营收前十名客户分析的时候需要用来计算每个客户站营业总收入的占比

## 具体流程讲解

### 弹出对话框 选择excel文件



第2行：该流程主要是弹出文件选择框，用于选择要分析的excel文件，并将文件路径赋值给”gexcel\_file\_path”变量，该变量是流程图变量（全局变量），因此以下所有流程块都可以访问到这个变量，无需在每个流程块中再选择excel文件。

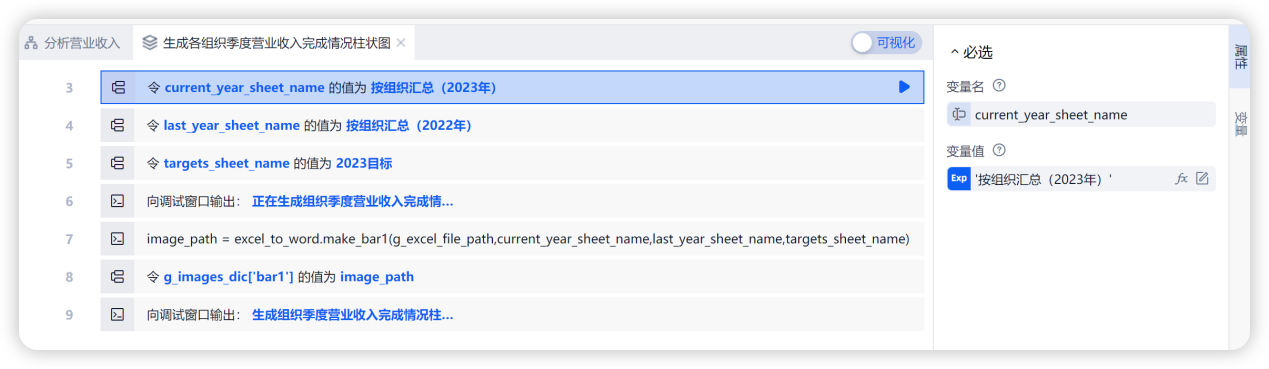
*备注：在来也的RPA中，约定全局变量的变量名以g\_开头，便于其他人阅读和理解代码。*

第3、4行：初始化g\_images\_dic和g\_word\_dic变量为字典。

第5行：弹出输入框，将用户输入的数据赋值给g\_quarter变量

第6行：将g\_word\_dic添加一对键值数据，键是’季度’，值是变量g\_quarter的内容，例如用户输入了’三’，那么g\_word\_dic的数据结构为{‘季度’:’三’}

### **生成各组织季度营业收入完成情况柱状图**



第3-5行，定义了3个变量，分别是3个excel的工作博名称，因为生成营业收入完成情况和目标完成情况需要用到这三个工作簿的数据，用户可以在右侧的属性栏自己修改成实际的工作簿名称。

第6行：打印调试信息以便知道当前程序的运行情况

第7行：调用插件excel\_to\_word的make\_bar1方法，向该方法传递excel路径和前面定义的3个变量，并将插件执行的结果（生成的图表路径）添加到字典g\_images\_dic中，索引键是’bar1’。

#### 查看excel\_to\_word插件的内容

插件位置在RPA目录下的extend\python

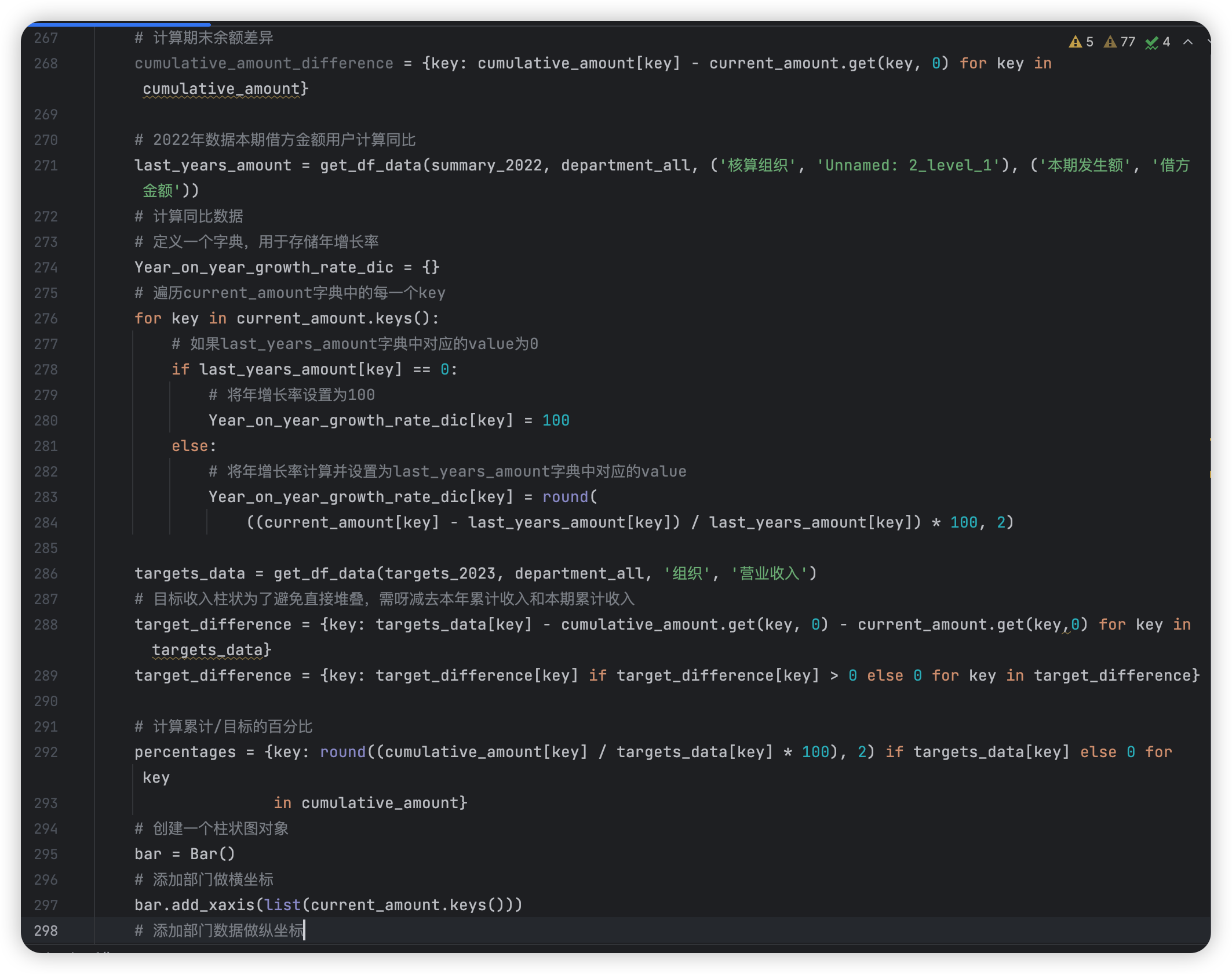
该插件使用python编程语言编写，excel\_to\_word.py是excel\_to\_word插件的主文件，里面包含了主要执行的代码函数；excel\_to\_word.lib文件夹是插件运行时所依赖的其他代码库存放的位置，images文件夹是生成的图表截图存放的位置，Template.docx是报告模版。

调用该插件的方法为excel\_to\_word.方法名(传入参数)，所需的参数个数在插件内的具体方法中定义。

使用VS code打开excel\_to\_word.py，搜索make\_bar1 即可找到生成柱状图的方法。该方法读取了本期工作簿sheet、同期数据工作簿sheet和目标数据工作簿sheet进行去除空行、排除不需要统计的部门等操作，获取数据进行计算本期发生额、本年累计、同比数据、目标数据等操作，将得到的结果使用echatrs生成图表。









### 生成各组织季度营业收入完成情况文字报告



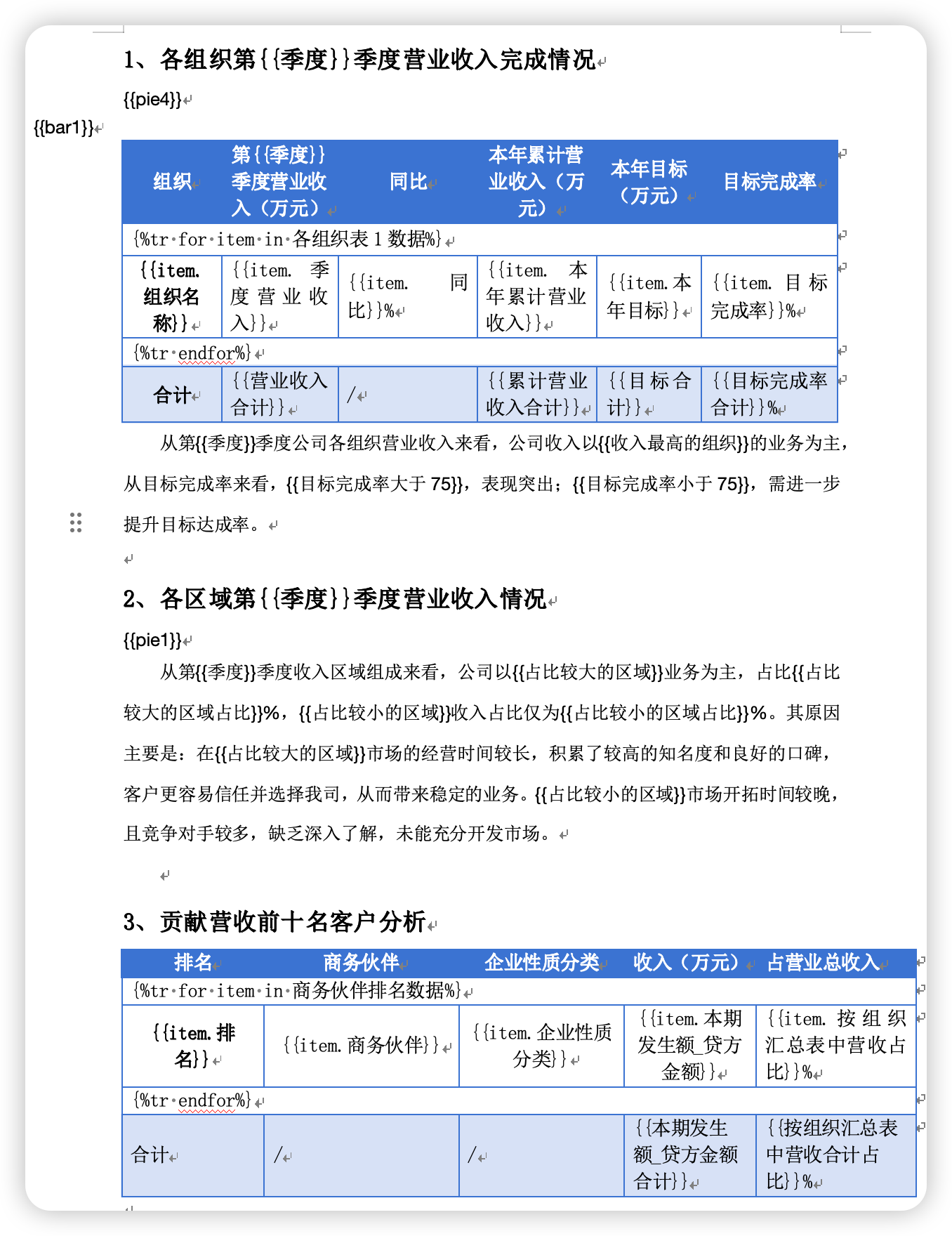
该流程与“生成生成各组织季度营业收入完成情况柱状图”的流程类似，区别在于该流程的主要数据处理方法是调用excel\_to\_word的make\_paragraph1方法，返回的数据是Template.docx模版文件中第一个段落所需的表格和段落数据，将这些数据添加到流程图变量g\_word\_dic字典中。后续的生成图表和生成分析文字报告的流程块基本与这两个流程类似，不再赘述。

### **将生成的图表、报告插入到word报告模板**



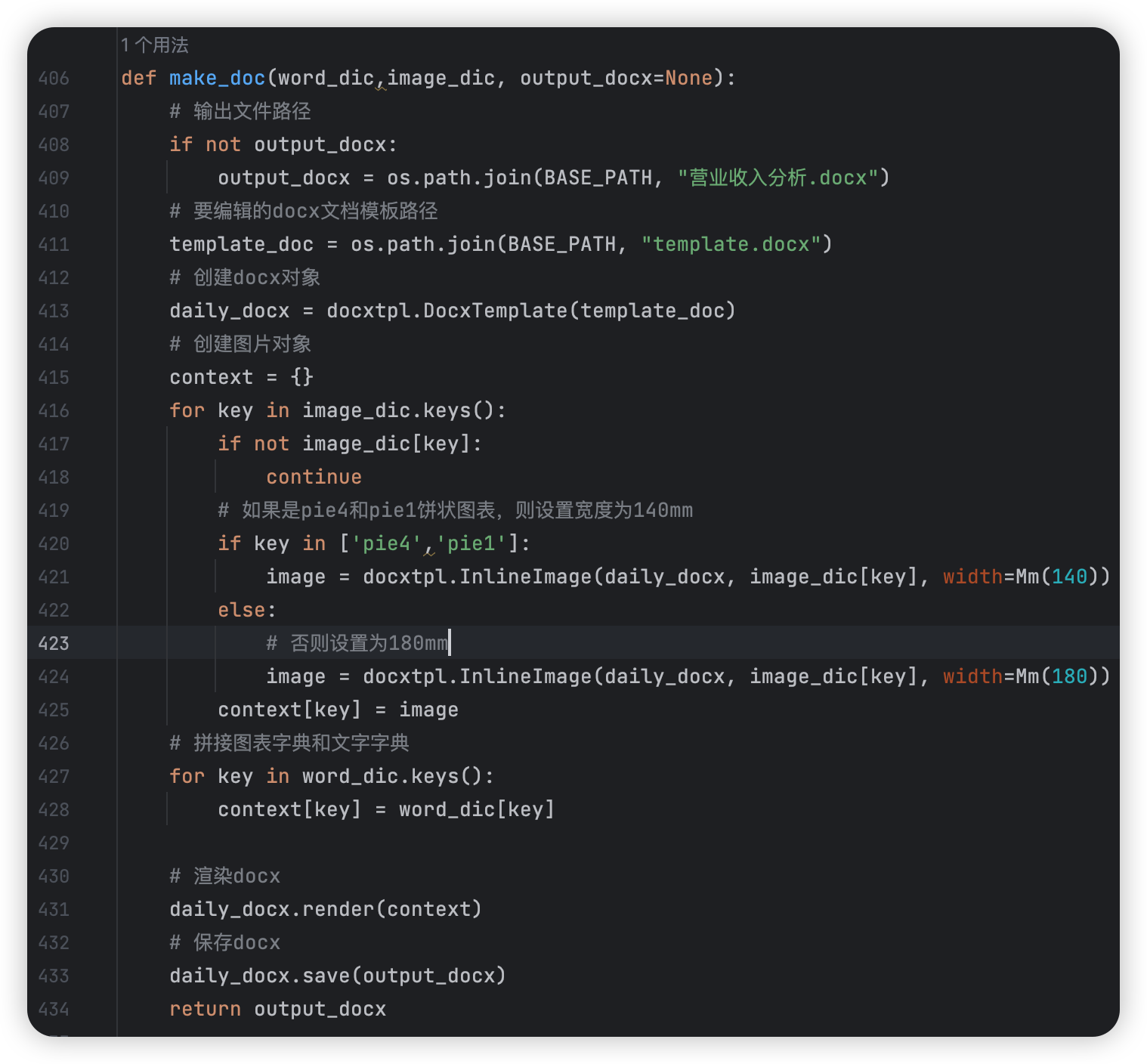
在之前的流程中，我们已经将所有生成的图表图片位置信息存放到了g\_images\_dic字典中，所有的报告所需的填充数据存放到了g\_word\_dic字典中。在这个流程中，将g\_images\_dic和g\_word\_dic传递给make\_doc方法，该方法将图片插入到模版的指定位置，将文字填充到指定位置，将表格数据遍历循环填充到表格的指定位置，最后存储为word文件。具体的代码实现可以阅读make\_doc方法的代码。

word模版如下：

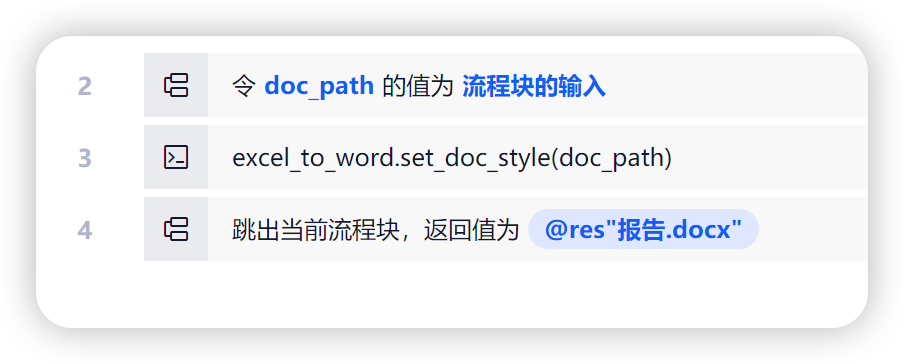


word模版中设置了需要替换的文字占位符和表格占位符，程序会根据数据自动填充到模版指定的位置。

生成word的make\_doc方法代码如下:



### **设置报告文档格式**



设置文档格式是调用了set\_doc\_style方法，该方法将传入的word文档读取，然后根据规则判断设置表格单元格的颜色，最后保存文件。

代码如下：

