第十章-文件和异常

- 从文件中读取数据
  - 读取整个文件
    - 示例
      - with open（'pi_digits.txt'）as file_object：
      - contents = file_object.read（）
      - print（contents）
    - 关键字with在不再需要访问文件后将其关闭
    - 我们调用了open（）但不调用close（）的话，python就会在适当的时候关闭文件
    - 方法read（）读取这个文件的全部内容，并将其作为一个字符串存储在变量contents中
    - read（）到达文件末尾时会返回一个空字符串，所以会有一个空行，如果不想要的话就用contents.rstrip（）除掉
  - 文件路径
    - 在Windows系统中，文件路径使用反斜杠\而不是斜杠/
  - 逐行读取
    - 把上面的第二行改为
      - for line in file_object:
      - 但是这么打印每一行都会多一个空白行，是因为源文件中每行的末尾都有一个看不见的换行符，可以用rstrip（）去除
  - 创建一个包含文件各行内容的列表
    - 把上面的第二行换为lines = file_object.readlines（）
    - readlines（）方法从文件中读取每一行，并将其存储在一个列表中，该列表被存储到变量lines中
  - 读取完文件之后就可以用各种方法利用它
    - 注意：读取完的文件类型是字符串，要用int（）或float（）转换为对应的类型
  - 包含一百万位的大型文件
    - 可以用切片，只读取部分
  - 圆周率中包含你的生日吗?
    - if birthday in pi_string:
    - 拓展：那么在第几位呢?
- 写入文件
  - 写入空文件
    - 需要在调用open（）函数的同时提供第二个实参，有：读取模式（'r'）、写入模式（'w'）、附加模式（‘a’）、让你能够读取和写入的模式（‘r+’）。如果没有实参，默认是

只读模式打开文件
    - **注意**：如果以写入模式打开文件时，如果指定的文件已经存在，python将在返回文件对象前清空该文件（要规避的话见“附加到文件”）
  - 写入多行
    - write（）函数不会再你写入的文本末尾添加换行符（哪怕你在编程时已经换行了），所以要加上\n来换行
  - 附加到文件
    - 如果只是给文件添加内容，而不是覆盖原先的内容，需要用附加模式打开文件。python在这模式下会将你写入到文件的行都附加到文件末尾。
    - 如果文件不存在，也会给你添加一个新的空文件
- 异常
  - try-except代码块
    - 规定了一旦遇到了错误该怎么办
  - else代码块
    - try-except-else，如：如果除法运算成功，就打印结果，就可以在else那里加一个print函数
      - 为什么不能直接在try那里写?
        - 书上：只有可能引发异常的代码才需要放在try语句当中，而仅在try代码成功执行时才需要执行的代码都应该放在else代码块当中
  - 分析文本
    - 无版权的文学作品：gutenberg.org
    - 方法split（）
      - 以空格为分隔符将字符串分拆为多个部分，并将这些部分都存储在一个列表中
  - 使用多个文件
    - filename是一个列表，列表中有各种文件名
    - 结合try-except板块可以达到这样一个效果：当某些文件名不存在的时候跳过，继续执行（提高了效率）
  - 失败时一声不吭
    - pass语句，可在代码块中使用它让python什么都不做（except）
- 存储数据
  - 使用json.dump（）和json.load（）
    - json.dump接受两个实参：要存储的数据以及可用于存储数据的文件对象
      - （A，B）存储A到B当中
    - json.load（）加载刚刚存储的信息
      - numbers = json.load（f_obj）
  - 保存和读取用户生成的数据

> - 思路就是用户有输入就保存为json文件

- 重构
  > - 代码能够正确地运行，但可做进一步的改进
  > - 思想：分而治之，一个函数做一种事情