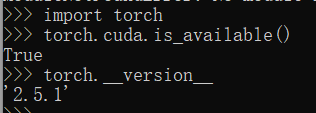
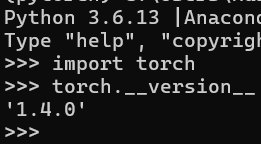
1. 安装了CUDA 12.6 、cuDNN、NumPy和Pytorch（通过Anaconda安装）



然后听说版本太高了很多东西不兼容，重新安了个低版本的，因为兼容性等问题弄了半天。出现的问题包括但不限于（由此获得了一些配环境的经验）：

1. pip下载太慢（切换THU下载源解决）
2. mkl-service和numpy版本过高，导致用tensorflow时说警告我没安装mkl-service。（单独卸载mkl-service重装解决）

最终保留版本：



1. 通过[《动手学深度学习》 — 动手学深度学习 2.0.0 documentation](https://zh-v2.d2l.ai/)学习：

学习了tensor。包括：定义方式、运算、广播机制、索引、切片。

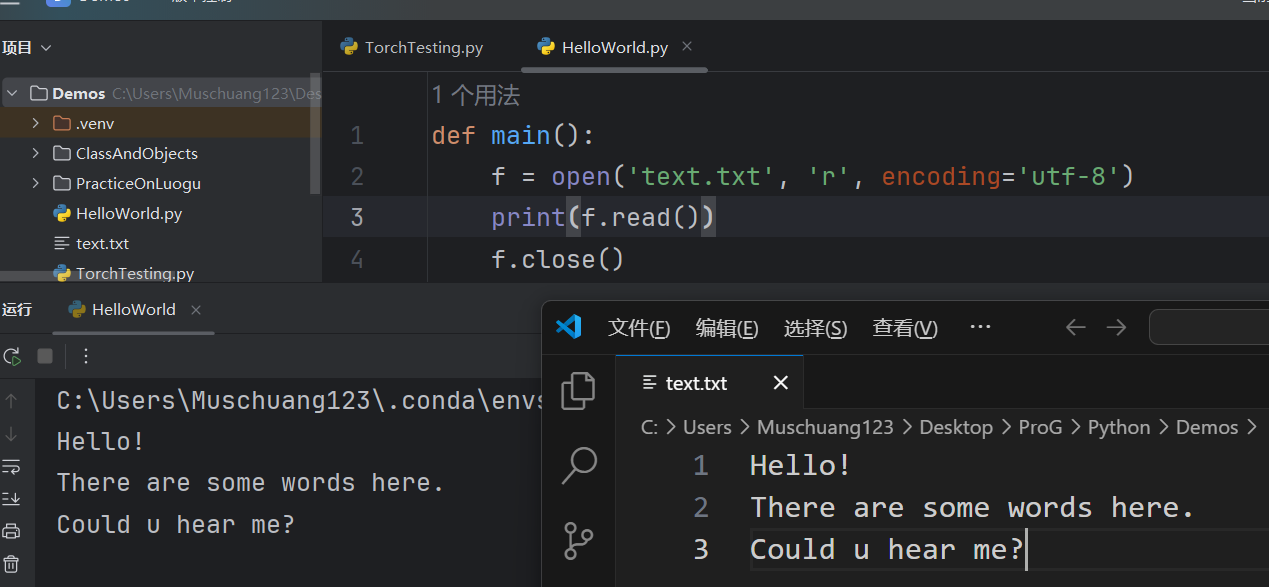
* 1. 通过我的探索，发现tensor是可以多层访问并切片的。



* 1. 并且还可以多层切片再切片：

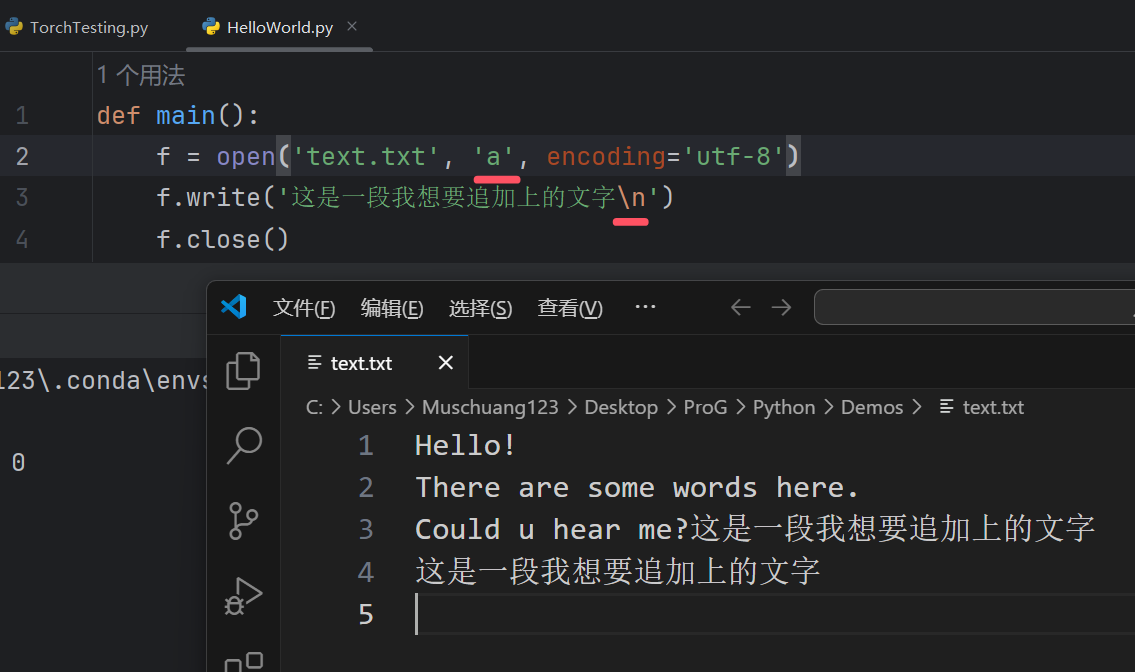


1. 文件操作：读取一个文本文件（文件编码格式是UTF-8）

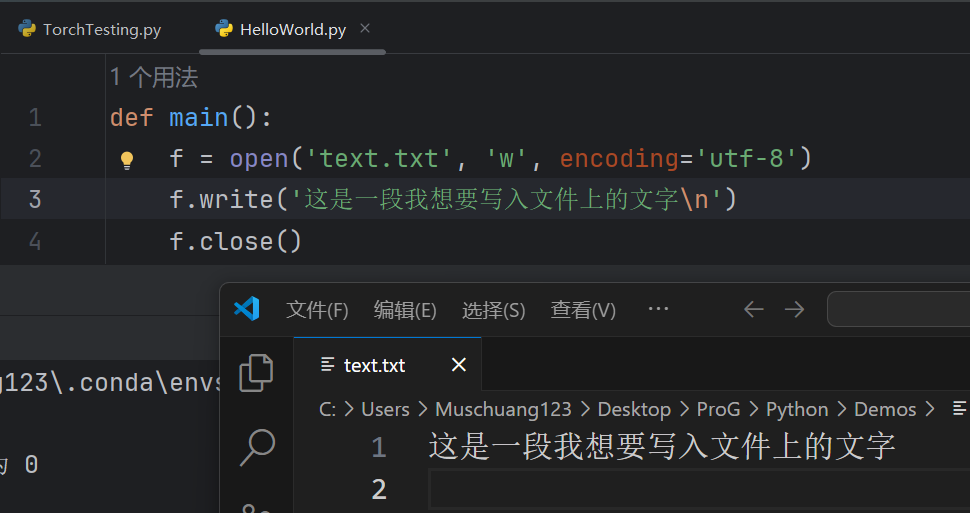


在文件后面追加内容：

我运行了两遍，可以看到追加内容是直接附在文件后面的，而不是另起一行，所以我们应当保持在文件结尾留一行空行的好习惯（？）



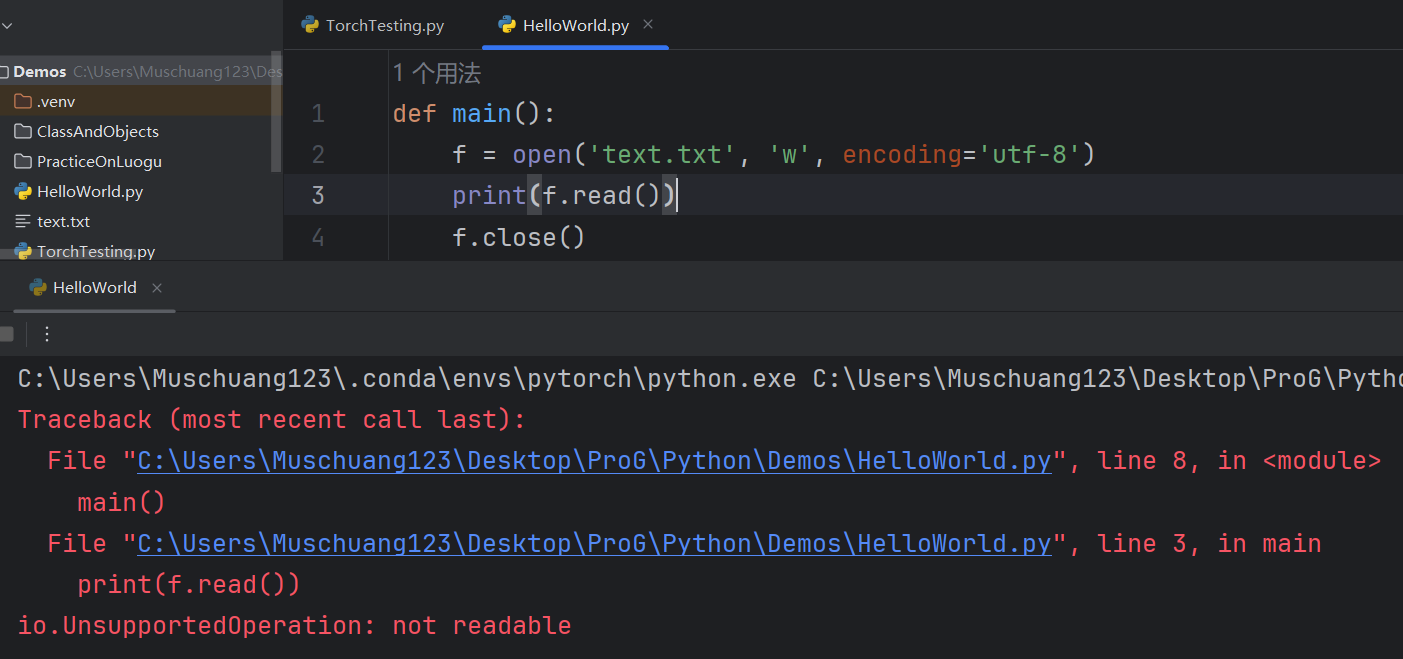
在文件中写入文字：（可以看到之前的文字被直接覆盖掉了）



另外只读模式下是不可以写入的：会产生异常。

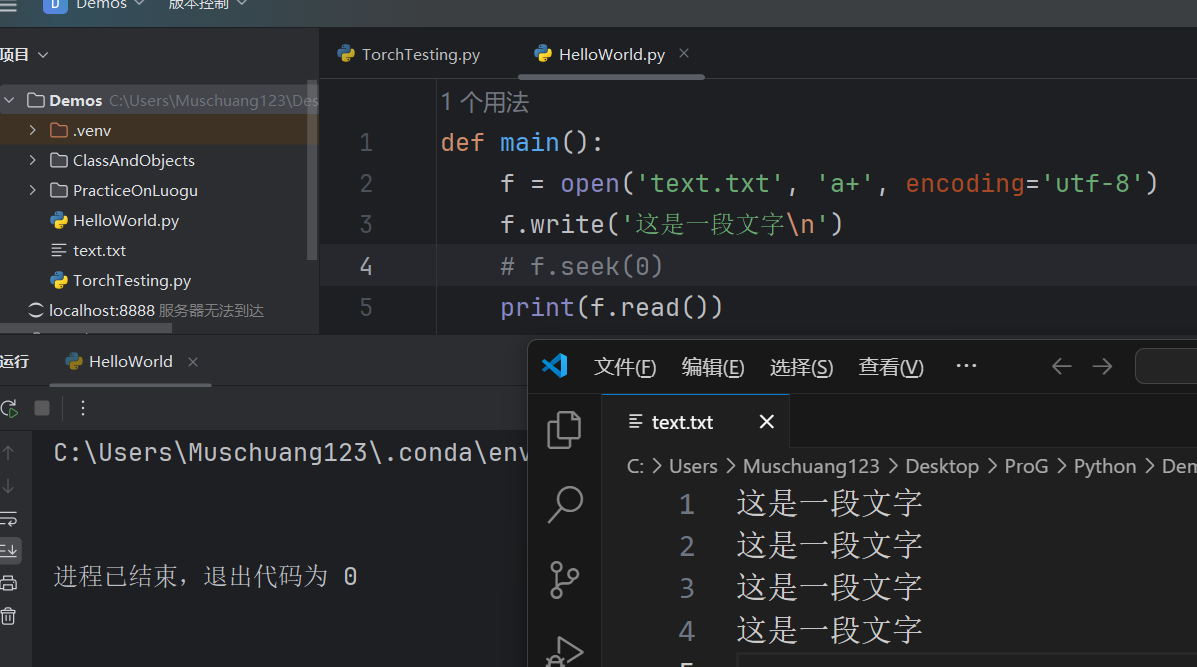


同样地，写入模式下也不可以读取：

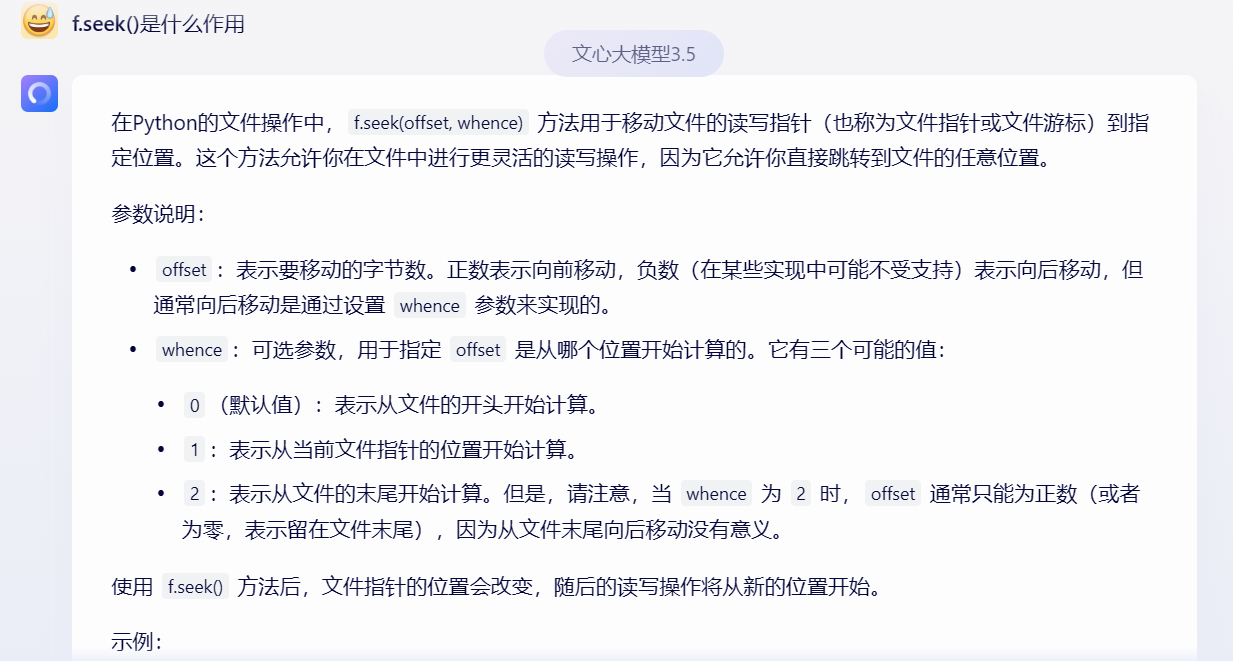


不过操作模式后加一个 ‘+’ 就可以做到又读又写：

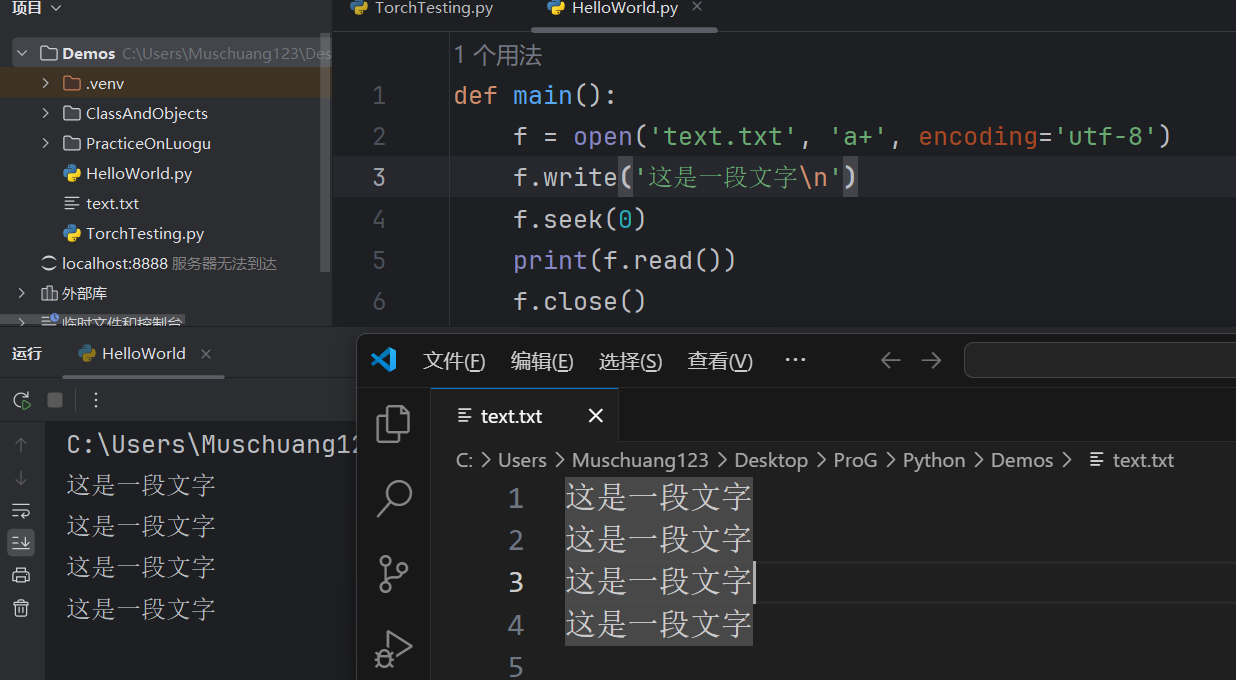
在实验过程中，我发现文件操作时有一个类似于指针的东西，当写入内容时，指针会移到文本末尾，而read()是从指针位置读取的，所以如果不移动指针，先写再读会发现读不到任何东西：



而我们可以通过seek()移动指针。



运行结果如下：



另外seek()中的offset移动指针操作只能在二进制的打开模式下使用。经过我实验，在默认模式下打开会报错：

**io.UnsupportedOperation: can't do nonzero end-relative seeks**

\*：目前只是对文件操作有初步了解，更多函数与方法会在用到时深入探究。

4. 异常处理：

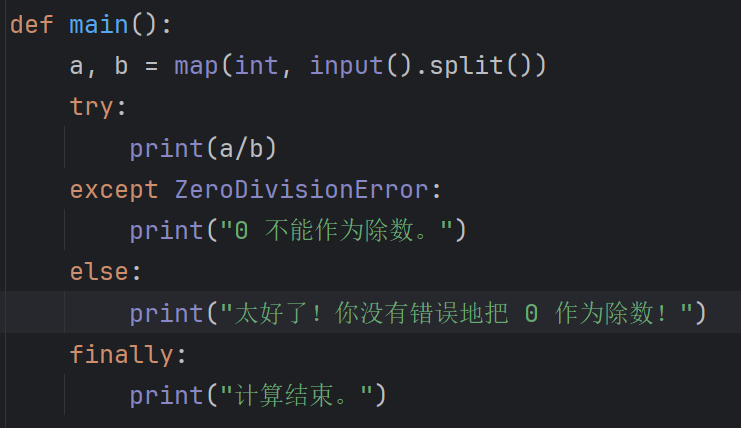
在写程序的过程中往往会出现异常，异常发生时程序就会停止，所以我们需要在异常发生时处理异常，使程序的鲁棒性（Robustness）变得更强。

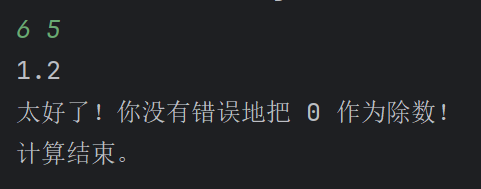
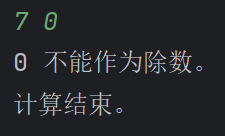
\*：另外Python是门解释型语言，我在文件操作实验的过程中，发现即使程序停止运行了，但是在停止之前的操作是已经做了的，我认为这是非常危险的。这不同于先编译后运行的语言，如果我通过单纯的运行程序来检查我的错误，就会导致错误出现之前的程序被运行了好几遍。有时这是我不想看到的。AI为我提供了解决方案：使用Mock对象、异常处理、代码审查、静态分析工具。

所以接下来我要学习异常处理。



进行了实验：



所以与Java的try-catch语句相同，只要知道异常类型就可以处理抛出的异常，但是比Java强大的一点是，Python还有个else语句块，可以在没有发生异常时执行一些语句。

异常的类型可以通过运行代码得知。

Python内置了许多异常类型，以下是一些常见的：

1. **Exception**：所有内置异常的基类。
2. **AttributeError**：尝试访问对象不存在的属性时引发。
3. **IOError**（在Python 3中已更名为OSError）：输入/输出操作失败时引发。
4. **ImportError**：无法导入模块或包时引发。
5. **IndexError**：使用非法索引访问序列（如列表）元素时引发。
6. **KeyError**：尝试访问字典中不存在的键时引发。
7. **KeyboardInterrupt**：用户中断执行（通常是Ctrl+C）时引发。
8. **MemoryError**：内存不足，无法分配新对象时引发。
9. **NameError**：尝试访问未声明的局部变量或不存在的全局变量时引发。
10. **TypeError**：操作或函数应用于不适合的类型时引发。
11. **ValueError**：操作或函数接收到具有正确类型但不适合的值时引发（如将字符串转换为整数时字符串包含非数字字符）。
12. **ZeroDivisionError**：除法或取模运算的第二个操作数为零时引发。

未来

1. 吴恩达机器学习的视频并未提供代码示例，我手痒痒，之后我将通过李沐的书学习：

[《动手学深度学习》 — 动手学深度学习 2.0.0 documentation](https://zh-v2.d2l.ai/)

1. 我认为很多语法学过之后用不到就忘记了，而且非常枯燥。所以我不再单独学习语法，而是用到了再学。
2. 我下载了一个CV的开源项目，后面几周，我将会剖析这个项目，并对Pytorch如何实现CV项目进行更深入的了解。

[Roujack/mathAI: 一个拍照做题程序。输入一张包含数学计算题的图片，输出识别出的数学计算式以及计算结果。This is a mathematic expression recognition project.](https://github.com/Roujack/mathAI)

1. 还有我发现配环境真的好费事，会遇到各种各样的问题，是比较消耗时间的。目前我的Jupyter Notebook和Pycharm还没有做到很好的融合，打示例做实验的时候还是比较卡手。我会尽快完善工具，提升效率。