ex16：

这节主要学习了有关处理文件方面的一些函数

包括close，read，truncate，write等等

同时进行了应用

源代码：

from sys import argv

script, filename = argv

print(f"We're going to erase {filename}.")

print("If you dong't want that, hit CTRL-C(^C).") # 键盘终端操作

print("If you do want that, hit RETURN.") # 回车继续

input("?") # 仅起到暂停的作用，并不存入1变量中

print("Opening the file...")

target = open(filename, 'w') # 使用open函数并且使用了‘w’模式，即写入模式

print("Truncating the file, Goodbye!")

target.truncate() # 清空文件内容，每次运行会先清空再写入//在‘w’模式下，不需要truncate，可删

print("Now I'm going to ask you for three lines.")

line1 = input("line 1: ")

line2 = input("line 2: ")

line3 = input("line 3: ")

print("I'm going to write these to the file.")

target.write(line1)

target.write("\n")

target.write(line2)

target.write("\n")

target.write(line3)

target.write("\n")

print("And finally, we close it.")

target.close() # 关闭文件，类似保存

同时做了巩固练习

使用input读入文件名，并简化了部分步骤

源代码：

print("So, what is your filename?")

filename = input(">")

print(f"We're going to erase {filename}.")

print("If you don't want that, hit CTRL-C(^C).")

print("If you do want that, hit RETURN.")

input("?")

print("Opening the file...")

target = open(filename, 'w')

print("Truncating the file. Goodbye!")

# target.truncate()

print("Now I'm going to ask you for three lines.")

line1 = input("line 1: ")

line2 = input("line 2: ")

line3 = input("line 3: ")

print("I'm going to write these to the file.")

target.write(line1 + "\n" + line2 + "\n" + line3 + "\n")

print("And finally, we close it.")

target.close()

需要注意的是

文件处理完毕后要close掉，否则会造成内存泄漏

另外在‘w’模式下使用open函数，不需要使用truncate

原因在于‘w’模式下的open在发现文件中存在内容时会自动truncate掉。

Pydoc中有相关解释

内容如下：  
mode is an optional string that specifies the mode in which the file

is opened. It defaults to 'r' which means open for reading in text

mode. Other common values are 'w' for writing (truncating the file if

it already exists), 'x' for creating and writing to a new file, and

'a' for appending (which on some Unix systems, means that all writes

append to the end of the file regardless of the current seek position).

In text mode, if encoding is not specified the encoding used is platform

dependent: locale.getpreferredencoding(False) is called to get the

current locale encoding. (For reading and writing raw bytes use binary

mode and leave encoding unspecified.) The available modes are:

========= ===============================================================

Character Meaning

--------- ---------------------------------------------------------------

'r' open for reading (default)

'w' open for writing, truncating the file first

'x' create a new file and open it for writing

'a' open for writing, appending to the end of the file if it exists

'b' binary mode

't' text mode (default)

'+' open a disk file for updating (reading and writing)

'U' universal newline mode (deprecated)

========= ===============================================================