**CS160 Web Programming – Python**

**HP05 Organizing Files and Debugging**

9/18/2019 Developed by Apiwat Chuaphan

9/21/2019 Reviewed by Tuan Tran and Clark Ngo

Center for Information Assurance (CIAE) @City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

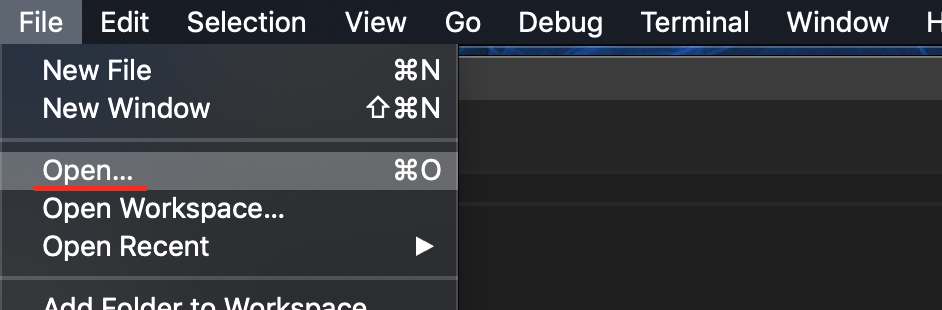
* Create python programs to organize files
* Debug your programs with Exception and Assertion
* Understand a Directory Tree

**Resources**

* [Automate the Boring Stuff with Python](https://login.proxy.cityu.edu/sso/skillport?context=89288)

**Preparation**

1. Open the VS Code and open the amazon-apprenti-2019/CS160/YOURNAME directory



Note: change YOURNAME to your real name. If you do not have the CS160 or YOURNAME folder, please create them under the amazon-apprenti-2019-2 folder.

1. Create a Module5 directory under YOURNAME directory. (If this directory exists, skip this step.)

**Organizing Files**

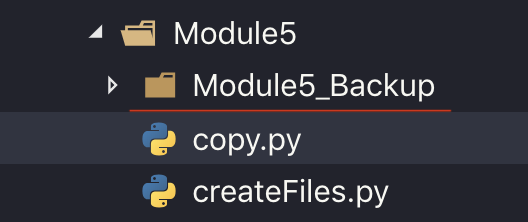
Python can help you organize files a lot easier with automation such as make copies of files, remove certain character from filenames, and compress them into ZIP files.

In this hand-on, we will use the **shutil** module which contains functions to let you copy, move, rename, add delete files in Python program. Warning: make sure to use functions with care, since you might end up with messing up your files.

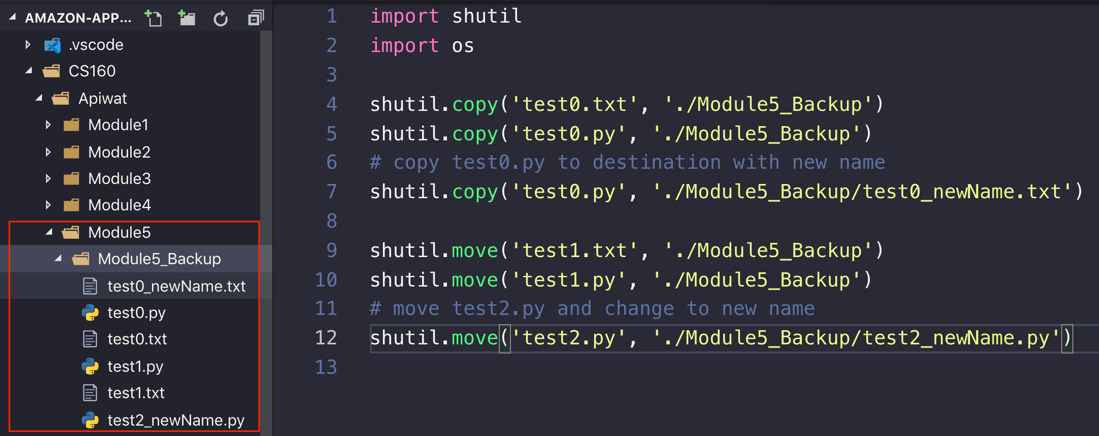
1. Create a file named *create\_files.py* to create 5 text files and 5 python files. Type the following.



1. Run the *create\_files.py* and now we have 10 new files in under Module5 directory, so we can play around with those files. Then, create new folder named “Module5\_Backup” under Module5 directory

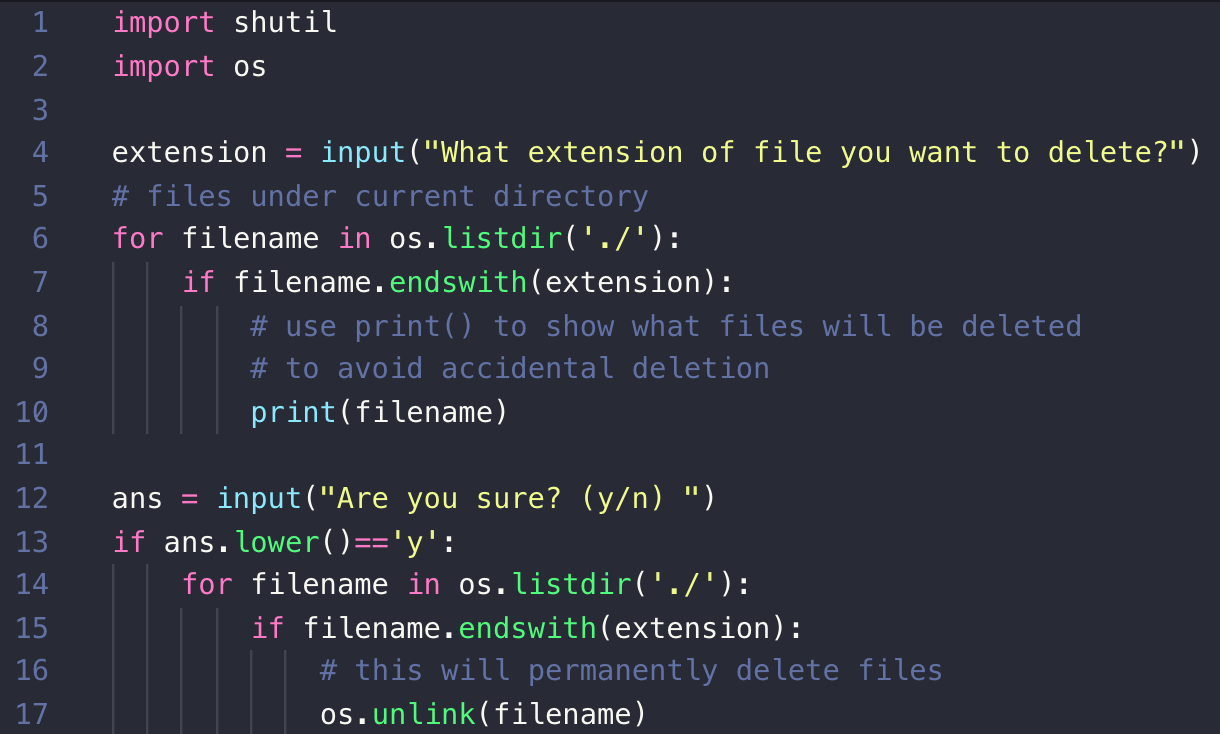


1. Create *copy.py* under Module5 and throw in the code below. The result will be similar to what in the red box after you run copy.py.

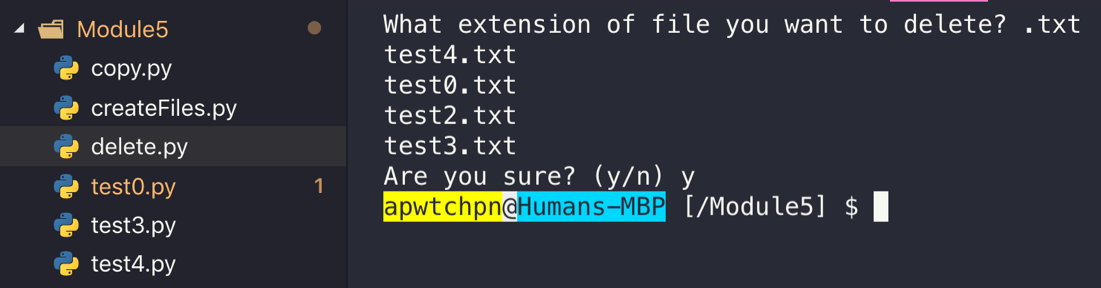


**Note:** you can also use *shutil.copytree()* to recursively copy an entire directory tree.

1. Here comes to deletion of files. The functions from **shutil** will permanently delete files and folder, so be careful with that. Create *delete.py* under Module5



The result will look like this. Notice that all .txt files has been deleted from Module5



1. ZIP files can also be done in Python. Let’s begin with creating z*ip.py* by typing the following code.

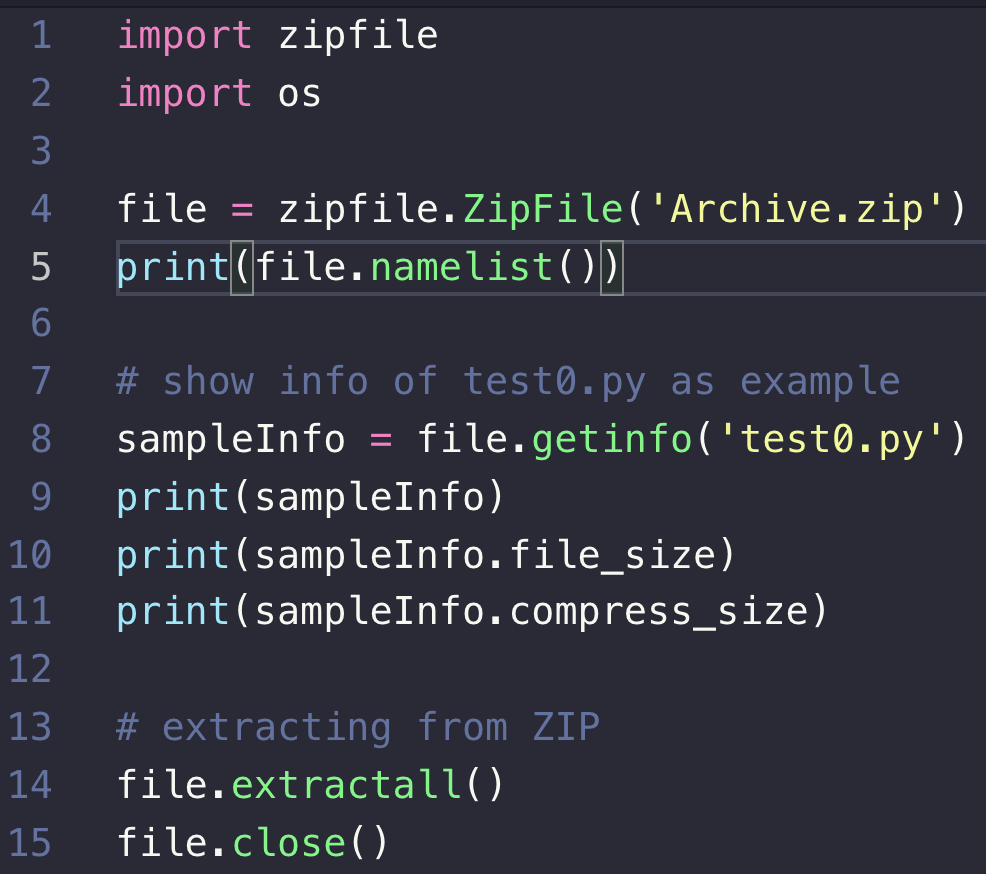


Run the file and notice that you will get Archive.zip under Module5 directory

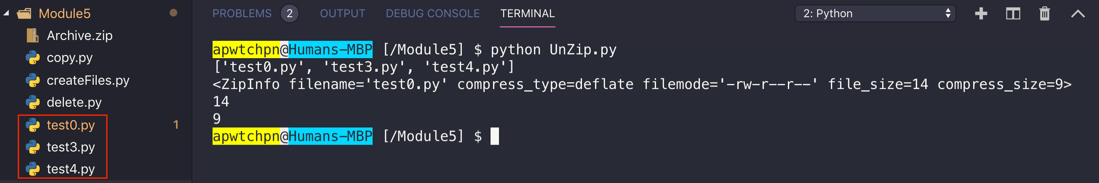
**Notes**: *w* is write, *a* is append

Source: <https://docs.python.org/3/library/zipfile.html#zipfile-objects>

1. Before moving on, please delete test0.py, test3.py, test4.py under Module5 directory because we will extract them from the ZIP file. Create *unzip,py* with the code below.



The result shows that you get those three files after running *unzip.py* and some information on the terminal.

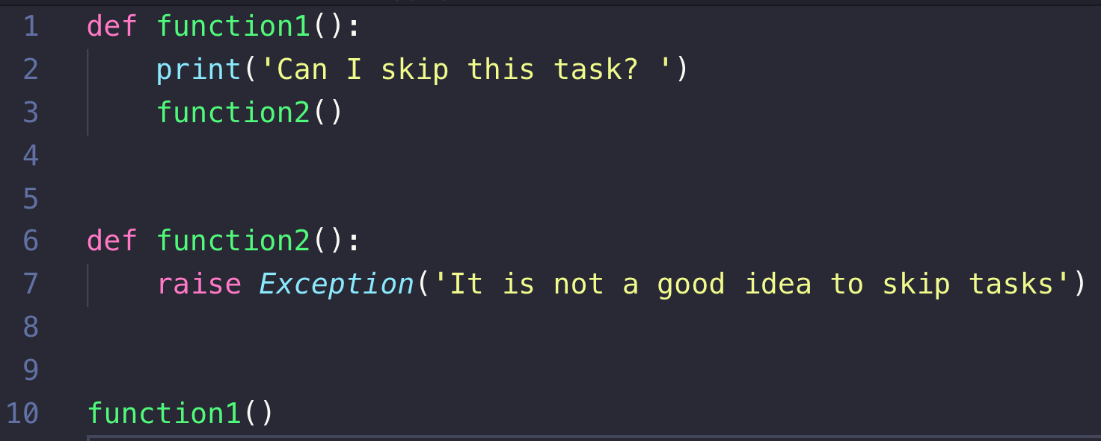


**Debugging**

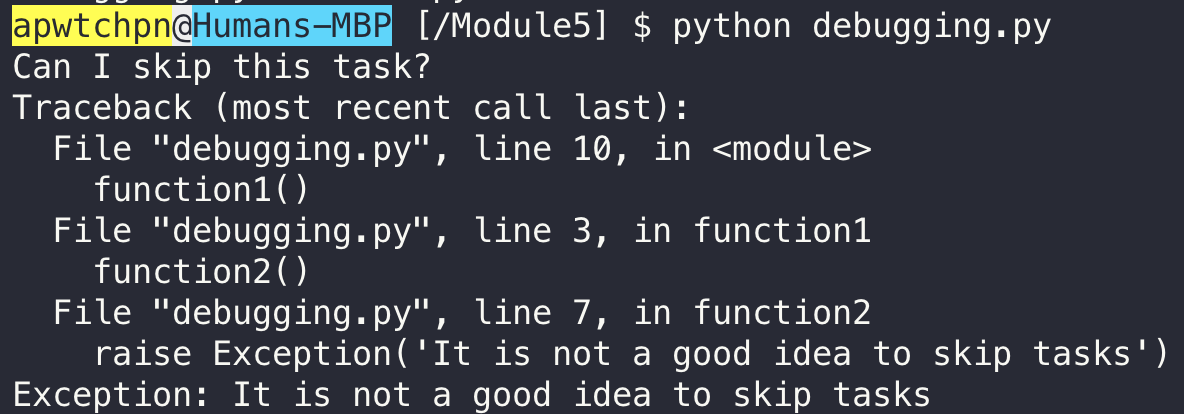
Finding a root cause in your program might be complicated. In this topic, we will cover some tools and techniques to help you fix them more efficiently.

* + Raising exception
  + Assertion statement

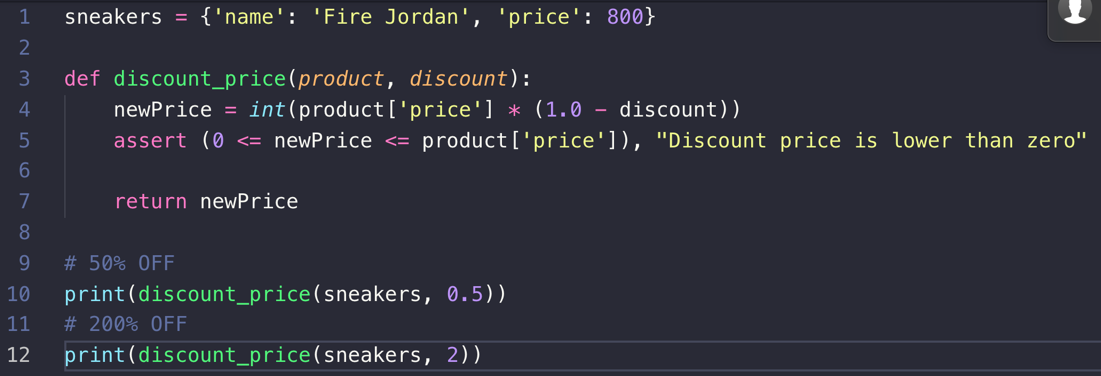
1. Python raises an exception when it executes invalid code. Let’s see how it works by creating *debugging.py* and type the following.



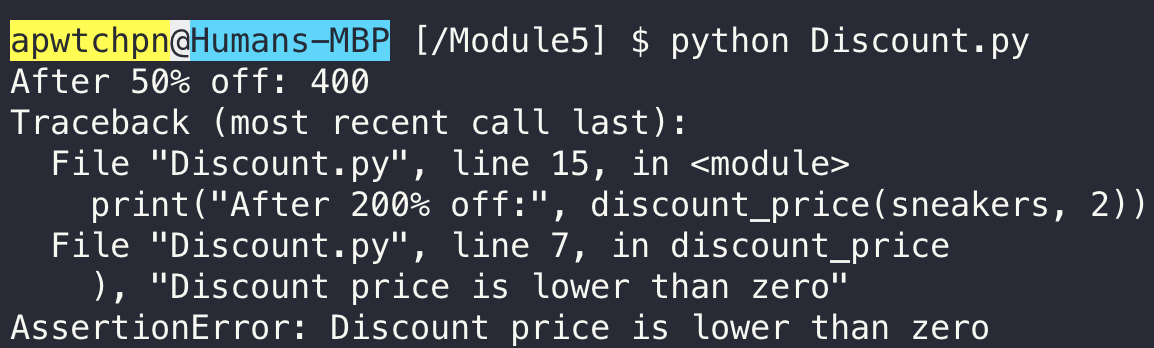
You will see the error message when you execute the program like this. The error happened on line 7 and was called from line 3 and 10.



1. **Assertion** is a sanity check and raise AssertionError if the assert condition evaluates to false, does nothing if the condition is true. Here is the example, create *discount.py* with the following.



The output will be like this.



As you can see discount 50% worked fine, but discount 200% caused an error. How can you pay -800 to get the sneakers?

**Note:** Basically**, Assertions** should be used to check something that should never happen, while an **exception** should be used to check something that might happen.

**Push your work to GitHub**

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hitting the “control” + “~” key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 5”

>>> git push origin YOUR\_BRANCH\_NAME

Note: you should change the YOUR\_BRANCH\_NAME to your own branch name. It should be firstname-lastname (e.g. maria-gracia).

If you cannot remember, run the command “git status” to check.

**Create a pull request**

* Go to the remote repository in GitHub
* Create a pull request