Pablo Montijo

March 8st, 2021

Foundations of Programming: Python

Assignment08

<https://github.com/pablomarcel/IntroToProg-Python-Mod08>

**Introduction**

In this document, I present the theory of classes and objects. I also discuss concepts such as fields, properties, constructors, methods and docstrings. Finally, I explore examples with Git commands and a use case with GitHub Desktop. A Python Script is shown at the end along with the testing result in the Windows Console.

**What is the difference between a class and the objects made from a class?**

The class is the blueprint, and the object is a copy made based off the blueprint. An object is a concrete realization of the class. Also, the class occupies a define address in memory, whereas the objects created from the class occupy a different address.

**What are the components that make up the standard pattern of a class?**

As shown in figure 1, the main components are the class name, the base class if any, the docstring, the class fields, the class constructor, the constructor attributes, the properties, and the methods.

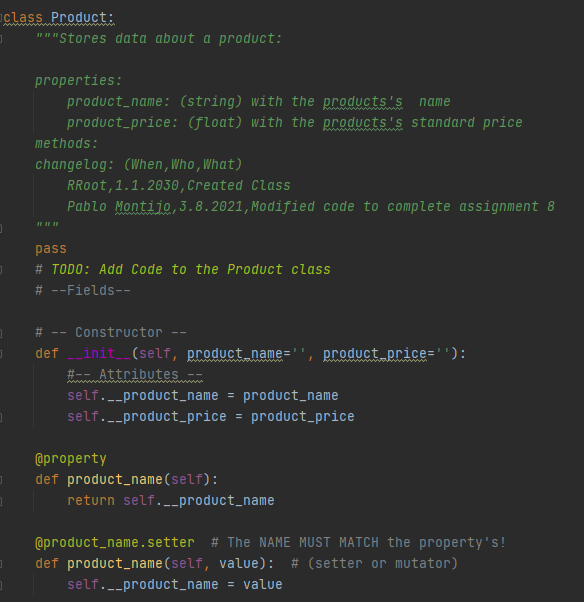


Figure 1 Shows a class structure.

**What is the purpose of a class constructor?**

To initialize an object with the assigned parameters. As shown in figure 2, whenever there is a call to create a new object of type ‘Product’, the constructor assigns the *product\_name* and the *product\_price* parameter to the object’s attributes.

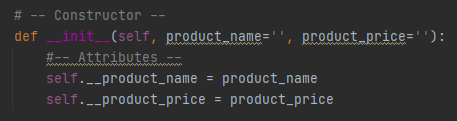


Figure 2 Shows an example of a constructor.

**When do you use the keyword "self?"**

Whenever we are working with non-static methods or put it in a different way when there is need to work with methods in object instances made off the class. The constructor in figure 2 is an example of a case where the keyword ‘self’ is used.

**When do you use the keyword "@staticmethod?"**

When there is need to work with class methods to be called out directly without the need to create an object or instantiate the class (Root, R. Programming with Python, Module 08 Notes, page 13). Figure 3 shows an example of a static method. The method is part of class ‘FileProcessor’. We don’t need to create an object based off ‘FileProcessor’, therefore the ‘staticmethod’ keyword is used.

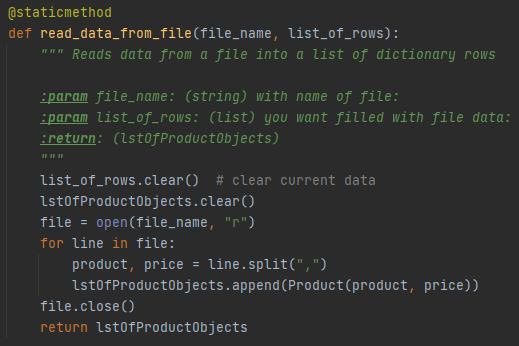


Figure 3 Shows an example of formatted text, per code in figure 1

**How are fields and attributes and property functions related?**

The attributes are called out in the class constructor. The property are functions that get and set the values for the class attributes via the getter and setter methods. The fields are variables used in the class. Class fields are present in all the objects instantiated from the class. The constructor in figure 2, shows an example of attributes used in a class.

**What is the difference between a property and a method?**

Both are methods, but properties are functions limited in scope to assign to and get values from object’s attributes. The scope of methods is broader, not just limited at handling attribute data, for example, manipulating lists as shown in figure 4.

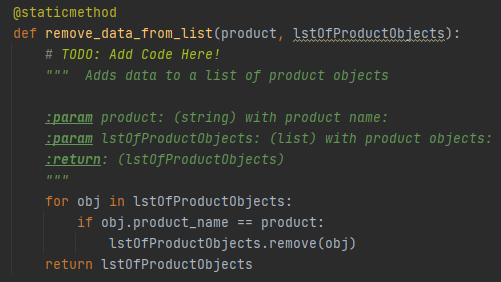


Figure 4 Shows a method

**Why do you include a docstring in a class?**

It tells other developers the scope of a class and how it is supposed to be used. Also, it is helpful to keep the code organized and inform of the properties, methods and change log for a particular class. Figure 5 shows and example of docstring for class ‘FileProcessor’. It communicates the description, methods and change log.

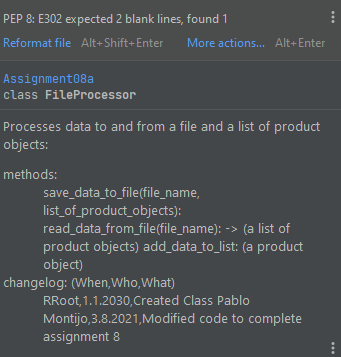


Figure 5 Shows an example of a ‘docstring’.

**What is the difference between Git and GitHub?**

Git is a version control system that uses the Git syntax to initialize and clone databases and, pull from and push to databases to handle document changes. GitHub is a company that materializes a version control system in the form of a desktop application and a website to conveniently manage changes in documents. Figure 6 and 7show the use of Git Commands for a personal project called ‘TurtleVsHare’. Git created a copy of the repository in a local machine to make the documents available for editing.

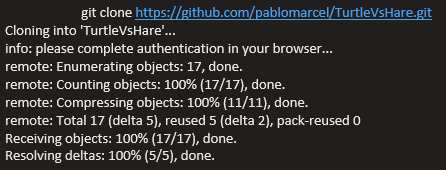


Figure 6 Shows the Git clone Command.

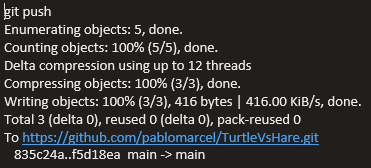


Figure 7 Shows the Git push Command.

**What is GitHub Desktop?**

Is a desktop application that creates folders in a local machine linked to repositories. Figure 8 shows the history for the repository created for assignment 08. The first step was to upload the python script. Since this is all new, everything is highlighted in green.

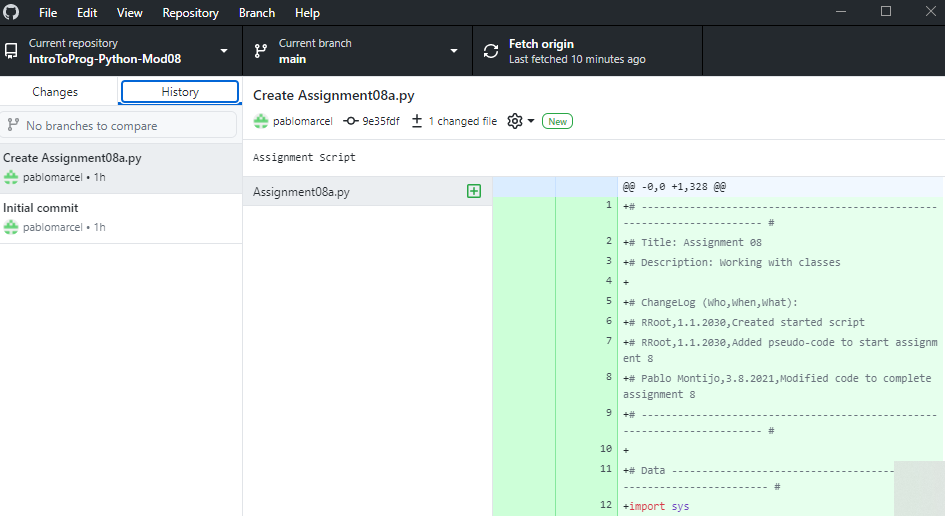


Figure 8 Shows a screenshot of GitHub Desktop

**Python Script**

***Main Body.***

The main code handles the while loop and handles the user choices. Option 1 directs the user to add a new product. Option 2 removes an existing product. Option 3 saves to file. Option 4 reloads data from file and Option 5 exits.

***Input Output.***

This section handles the user input and incorporates the try-except blocks to catch type errors. A partial listing of class IO is shown in figure 8.

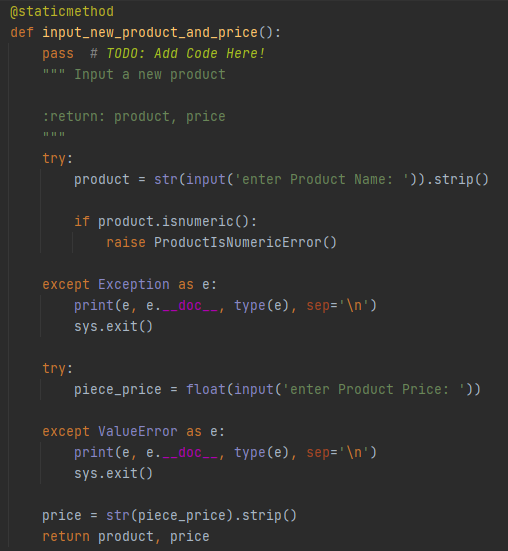


Figure 9 Shows the input method in the IO Class.

***Processing.***

This section handles the manipulations with the list of objects. The ‘FileProcessor’ class contains 4 methods. The ‘read\_data\_from\_file’ method puts the contents of the text file in a list of product objects. The ‘add\_data\_to\_list’ method appends an object to the list. The ‘remove\_data\_from\_list’ method removes an object from the list. And finally, the ‘save\_data\_to\_file’ reads the objects list and translate its contents to a text file.

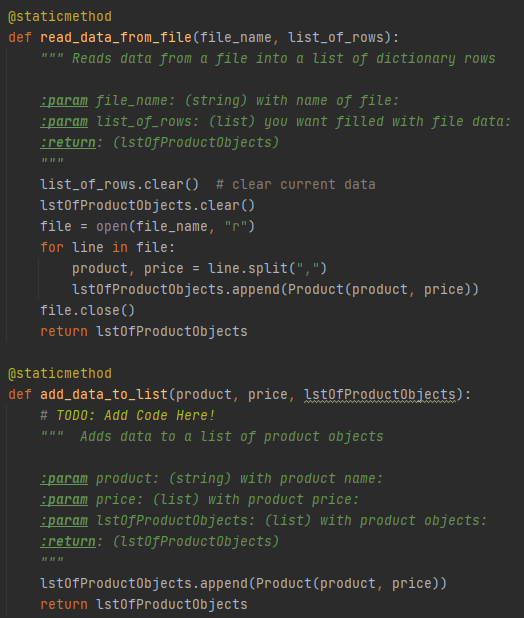


Figure 10 Shows 2 methods of the ‘FileProcessor’ class.

***Error Handling.***

This section handles the custom exception classes. Class ‘ProductIsNumericError’ comes alive whenever a user enters numbers when asked for a Product.

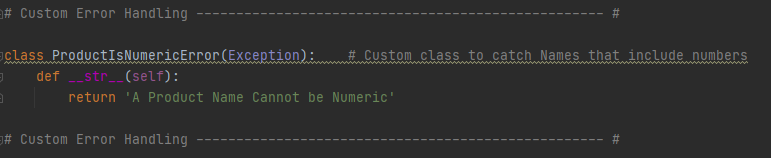


Figure 11 Shows custom exception classes.

***Data***

Class Product handles the object creation. As shown in figure 12, the class includes the docstring, the constructor, and the properties. Class FileProcessor uses class Product when managing manipulations on the list of objects.

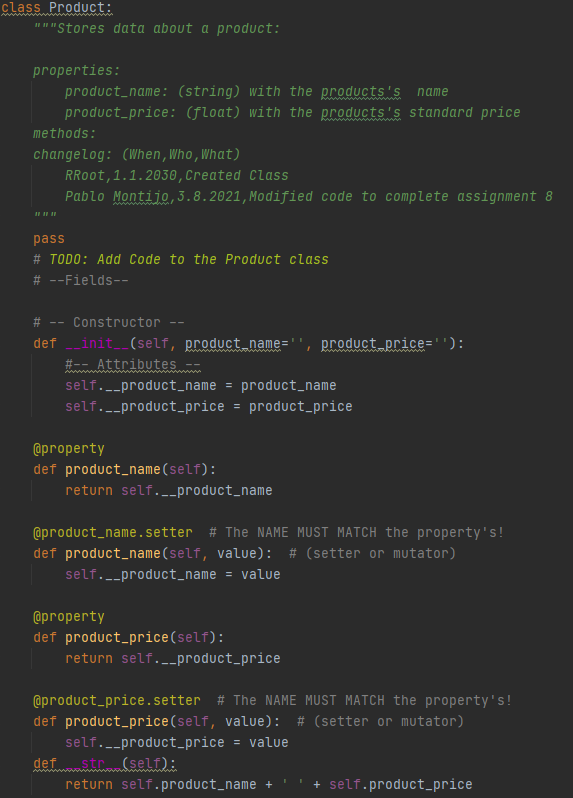


Figure 12 Shows class Product

**Console Test.**

The console test shows the code is behaving as expected per the summary below.

The user selects option 1 ‘Add a new Product’.

The user selects option 3 ‘Save Data to File’.

The user selects option 4 ‘Reload Data from File’.

The user selects option 2 ‘Remove an existing Product’.

The user selects option 3 ‘Save Data to File’.

The user selects option 5 ‘Exit Program’.

python Assignment08a.py

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

guitar,18

desk,1

table,22.0

ipad,99.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Product

2) Remove an existing Product

3) Save Data to File

4) Reload Data from File

5) Exit Program

Which option would you like to perform? [1 to 5] - 1

enter Product Name: iphone

enter Product Price: 33

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

guitar,18

desk,1

table,22.0

ipad,99.0

iphone,33.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Product

2) Remove an existing Product

3) Save Data to File

4) Reload Data from File

5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Save this data to file? (y/n) - y

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

guitar,18

desk,1

table,22.0

ipad,99.0

iphone,33.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Product

2) Remove an existing Product

3) Save Data to File

4) Reload Data from File

5) Exit Program

Which option would you like to perform? [1 to 5] - 4

Warning: Unsaved Data Will Be Lost!

Are you sure you want to reload data from file? (y/n) - y

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

guitar,18

desk,1

table,22.0

ipad,99.0

iphone,33.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Product

2) Remove an existing Product

3) Save Data to File

4) Reload Data from File

5) Exit Program

Which option would you like to perform? [1 to 5] - 2

enter Product to remove: desk

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

guitar,18

table,22.0

ipad,99.0

iphone,33.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Product

2) Remove an existing Product

3) Save Data to File

4) Reload Data from File

5) Exit Program

Which option would you like to perform? [1 to 5] - 3

Save this data to file? (y/n) - y

Press the [Enter] key to continue.

\*\*\*\*\*\*\* The current Tasks ToDo are: \*\*\*\*\*\*\*

guitar,18

table,22.0

ipad,99.0

iphone,33.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Menu of Options

1) Add a new Product

2) Remove an existing Product

3) Save Data to File

4) Reload Data from File

5) Exit Program

Which option would you like to perform? [1 to 5] - 5

Goodbye!

**Summary**

In this activity I reviewed the theory about classes, objects, constructors, properties, and methods. The python script for this activity helped to better understand how objects work. As shown in the python script, the manipulation with objects and classes is straightforward and convenient. There are unlimited possibilities when working with classes and objects because, any number of object attributes and methods can be used. And most importantly, code reuse saves time. The code in assignment 08 was reused from assignment 06 with minor changes. Finally, as a side activity, I presented examples about Git commands and GitHub Desktop. Version control in the desktop environment turn out to be a nice to have feature.