Monroe Township Library Coding Bootcamp

Class 10 Notes

* Classes continued…
* Subclasses & Inheritance
* Python modules & frameworks
* Resources for further learning/practice

**Subclasses/Inheritance:**

* You can create a subclass by including a previously created class in parentheses in the new class’ definition
* A subclass inherits all properties and methods associated with its parent class
  + For example, if we created the method func() in a class, and we created a second class that inherits from the first, we would be able to call the func() method on any instance of that class
  + Parent class methods can also be replaced by creating a method of the same name in the subclass
* You can use the super() function to get the class directly above
  + For example, if we want to implement a similar \_\_init\_\_ method, we can call super() and grab the \_\_init\_\_ method from the parent class so that we don’t have to retype all the same code

**Python Modules & Frameworks:**

* Importing modules and frameworks gives you access to useful functions and classes that can make problem-solving easier, and helps avoid redundancy in code
  + For example, if you are working on an app that analyzes data from the Internet, there are already well-established modules that allow you to make URL requests with your code, you don’t need to spend time trying to write that part from scratch and you can focus on the unique things your app does instead
* There are hundreds of modules in the Python standard library (which comes included with your Python installation) and thousands more that you can easily install
  + External modules & frameworks can be found on <https://pypi.org/>
  + Most modules can be installed from your computer’s command line using the command:  
    pip install <name of module>
  + Modules will also include installation instructions on their PyPi page

**Popular Python Modules:**

* **Web Development:**
  + requests : makes http requests to get data from websites
  + django : a web development framework for creating websites and web apps, as well as backend data management (used by Instagram and Tinder)
  + flask : another web development framework that is more lightweight and easier to use than Django for less complex web projects
* **Data Science:**
  + numpy : for performing complex mathematical operations, matrix operations, etc.
  + pandas : for data analysis and manipulation of data
  + matplotlib : good for data visualization like graphs and charts
* **Machine Learning/AI:**
  + tensorflow : open-source module developed by Google, used for creating neural networks and running machine-learning algorithms
  + pytorch : developed and maintained by Facebook, primarily used for Natural Language Processing systems
* **GUI/Desktop App Development:**
  + tkinter : a basic, easy-to-use GUI module for developing all types of desktop applications and software
  + PyQt5 : a slightly more complicated GUI framework with lots of customization options for developing modern looking desktop apps

**Resources for Further Learning:**

* **Online courses:**
  + SoloLearn
  + Codecademy
  + Udemy
* **Coding practice:**
  + CodeKata
  + CodeWars
  + LeetCode
  + Github (analyze and contribute to projects)
* **Books:**
  + The Big Book of Small Python Projects – Al Sweigart
  + Python Crash Course – Eric Matthes
  + The Pragmatic Programmer – Andrew Hunt

**Check class files at** [**github.com/monroecoding**](https://github.com/monroecoding)

Chris DiFazio

Adult Services Librarian

cdifazio@monroetpl.org