**Download Your Daily Notes**

**My own views on Data Types**

 Modules are pre-defined files in Python that contain code representing basic functionalities such as classes, methods, and variables. They consist of a collection of functions and classes grouped together in files within a directory. Modules are sometimes referred to as libraries. Their purpose is to provide predefined methods that enhance code efficiency and reduce redundancy.  
  
By utilizing modules, code can be organized and repetitive functions commonly used in the code can be encapsulated. This reduces code duplication and improves code readability.  
  
Some examples of commonly used modules in Python include:  
  
OS: Provides functions for interacting with the operating system, such as file management and directory operations.  
  
Time: Offers functionality related to time, including time measurement, manipulation, and formatting.  
  
Math: Contains mathematical functions and constants for performing various mathematical operations.  
  
Matplotlib: A library used for creating visualizations and plots in Python.  
These modules, among others, provide additional functionality and features that can be easily integrated into Python code, making it more efficient and concise.

**Daily Notes - Data Types**

 Modules are pre-defined files in Python that contain code representing basic functionalities such as classes, methods, and variables. They consist of a collection of functions and classes grouped together in files within a directory. Modules are sometimes referred to as libraries. Their purpose is to provide predefined methods that enhance code efficiency and reduce redundancy.  
  
By utilizing modules, code can be organized and repetitive functions commonly used in the code can be encapsulated. This reduces code duplication and improves code readability.  
  
Some examples of commonly used modules in Python include:  
  
OS: Provides functions for interacting with the operating system, such as file management and directory operations.  
  
Time: Offers functionality related to time, including time measurement, manipulation, and formatting.  
  
Math: Contains mathematical functions and constants for performing various mathematical operations.  
  
Matplotlib: A library used for creating visualizations and plots in Python.  
These modules, among others, provide additional functionality and features that can be easily integrated into Python code, making it more efficient and concise.

**Daily Notes - Integers**

 When a module is imported in a Python program, the Python interpreter retrieves the module from one of the following locations:  
  
Program Directory: The module is fetched from the directory where the program is located.  
  
PYTHONPATH Variable Directory: The module can also be fetched from a directory specified in the PYTHONPATH variable. This variable contains a list of directories where Python searches for modules.  
  
Default Directory: If the module is not found in the program directory or the directories specified in the PYTHONPATH variable, Python will look for it in the default directory.

**Daily Notes - Floating point numbers**

 This discusses float data type in Python which handles decimal numbers with high precision. Float can be called as a function with or without an argument, and the function tries to convert the argument into a float. However, not all values can be converted into float data type. It also provides examples of formatting float numbers using different tokens in Python.

**Daily Notes - Unpacking Argument Lists**

 This covers the concept of strings in Python. Strings are a sequence of Unicode characters that form a single manageable string and are represented by the immutable str data type. The str function is used to create a string, and when no argument is supplied, it returns an empty string. The function is often used to convert other data types to strings.  
  
The example code provided demonstrated how to use strings and input to produce a suitable output. The ‘+=‘ operator was also introduced, which adds values to an existing variable, and the end of line escape sequence was used to make code more readable. Enclosing expressions in brackets is another way to make code easier to read.

**Daily Notes - Lambda Expressions**

 Lambda expressions allow you to create small anonymous functions with the lambda keyword, which can be used wherever function objects are required. They are restricted to a single expression but can reference variables from the containing scope. Lambda functions can be used to pass a small function as an argument or to return a function. An example of this is shown, where a lambda function is used to sort a list of tuples based on the second element of each tuple.

**Daily Notes - Conventions about the content and formatting of documentation strings**

 The conventions for formatting documentation strings are outlined, including the importance of a concise summary and the use of blank lines to separate the summary from the rest of the description. Function annotations are also discussed, as well as the importance of coding style, including the use of 4-space indentation and following PEP 8 guidelines.

**Activity 1 - Consolidating Learnings**

 This has been completed and uploaded. The first one simply applies the sin calculation to a number inputted by me. The second one calls a module containing the name of a car and color that was created by me, it is called in a different file and then displayed to show it's functionality.

**My Views on the Day**

 1. Learning about modules, the ones that already exist and how to create a new one.  
  
2. Activity 1  
  
3. None  
  
4. None

**Daily Notes - Day 3 Reflections**

 1. Learning about modules, the ones that already exist and how to create a new one.  
  
2. Activity 1  
  
3. None  
  
4. None

**Activity 2**

 This activity has been completed, screenshotted and uploaded.