**Python Workbook**



**Instructions**

As you progress through the Python Workshop, use this guided workbook to take notes and reflect on your learning. You should take notes on vocabulary, concepts, processes, and key ideas. Feel free to modify the workbook template if you like. This workbook is not a program requirement and will not be graded. However, it is a good idea to capture your learning so that you make the most out of your educational investment.

## Module 1: Introduction to Python

## **Module 1: Introduction to Python**

### Lesson 1: Welcome to Python

* Describe a few things that Python can do with your data.
* Explain what sets Python apart from SQL and other programming languages.

Define the following key terminology of Python.

* Python
* Syntax
* Variable
* Expression
* Script
* Float
* String
* Print
* Boolean
* Loop
* Command Line
* Identifier
* Value
* Describe how Python relies on indentation.
* Describe the difference in how double and single quotes are used in Python.

### Lesson 2: Variable Types

* Identify the types of variables that will be most common to you as an analyst.
* Describe the parts of a variable.
* Describe case sensitivity in Python.
* Identify the rules to follow to declare descriptive variables.
* Provide examples of acceptable variable names.
* Define casting.

### Lesson 3: Multiple Variables

* Write code for assigning values to more than one variable at a time.
* Write code that assigns the same value to more than one variable on the same line.
* Define tuple.
* Identify the steps involved in creating a list in Python and some important tips.
* Write code for a list in Python.

### Lesson 4: Output Variables

* Write code that combines both text and variables in Python.
* Explain the method behind the above code.
* Identify what happens when you try to combine a string and a numeric in Python.

### Module Challenge: Python Review

To successfully complete this challenge, use Python to perform each question below.

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create and print a syntax that says, ‘Python is the best coding language!’

Insert your code here.

1. Create and print a syntax that says, ‘It’s time to go’, using both single quotes then double quotes. Which syntax returns an error.

Insert your code here.

1. Create two variables a and b. Give the value of dog and give b the value of cat.

Insert your code here.

1. Re-declare variable b value to bird.

Insert your code here.

1. Combine an output variable that says “Today is a ” with variable ‘a’. Variable ‘a’ has a value of “good day!”.

Insert your code here.

1. Create a list called sales. In the list include variables a, b, c, and d.
   * 1. The value of a will be orders.
     2. The value of b will be count.
     3. The value of c will be sold.
     4. The value of d will be cancelled.

Insert your code here.

1. Create two variables and add them together using the + symbol. What is the result?
2. Variable ‘a’ will have a value of 15.
3. Variable ‘b’ will have a value of 25.

Insert your code here.

1. Combine the following variables into one result:
2. Variable ‘x’ will have a value of “A bright  ”.
3. Variable ‘y’ will have a value of “ sunshining day!”
4. Variable ‘z’ will have a value of the result of x and y.
5. Print the results.

Insert your code here.

## **Module 2: Python and Basic Operations**

## Module 2: Python and Basic Operations

### Lesson 1: Statements

* Define assignment statement.
* Define multi-line statements.
* Write code for the following;
  + Multiple statements surrounded by parenthesis.
  + Multiple statements surrounded by brackets.
  + Multiple statements surrounded by curly brackets.
  + Multiple statements surrounded by semicolons.
* Describe why comments are valuable in Python and the symbol that is used to begin a comment.
* Write code for a comment in Python.

### Lesson 2: Data Types

* Define constructor function.

Define the following data types and provide examples using code.

* Text type:
* Numeric:
* Sequence:
* Mapping:
* Set:
* Boolean:
* Describe when parenthesis are used in Python versus when brackets are used in Python.

### Lesson 3: Numbers

Define the following numeric types in Python:

* Float
* Integer
* Complex
* Write code for defining number variables and values, using the type () function.
* Write code that combines the type() function with output variables using print statement, and declare variables.
* Describe what it means to convert numbers from one to another using the type() function.
* Use the code your previously wrote to change your numbers to be converted. Also include comments to indicate what is being done.

### Lesson 4: Operators

* Identify the arithmetic operators in Python.
* Write an example of how you would apply mathematical operators in python.
* Define logical operators in Python.
* Write an example of how you would apply logical operators in Python.
* Define identify operators.
* Write an example of how you apply identity operators in Python.

### Module Challenge: Simple Operations

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

To successfully complete this challenge, perform and answer the following:

1. What is an assigned statement?

Insert your answer here.

1. Create an assigned statement.

Insert your code here.

1. What is a multi-line statement?

Insert your answer here.

1. Create a multi-line statement using continuations (brackets, parentheses, semicolons etc.).

Insert your code here.

1. Create a multi-line statement using new line characters.

Insert your code here.

1. What is a string data type?

Insert your answer here.

1. Create a string variable that says, ‘Data Analytics is fun!’ using a constructor function.

Insert your code here.

1. Create a string variable that says, ‘Data Analytics is fun!’ without a constructor function.

Insert your code here.

1. What is an integer data type?

Insert your answer here.

1. Create an integer with a variable of ‘x’ and a value of 100 using a constructor function.

Insert your code here.

1. Create an integer with a variable of ‘x’ and a value of 100 without a constructor function.

Insert your code here.

1. What is a float data type?

Insert your answer here.

1. Create a float with a variable of ‘y’ and a value of 3.6 using a constructor function.

Insert your code here.

1. Create a float with a variable of ‘y’ and a value of 3.6 without a constructor function.

Insert your code here.

1. Can you convert a complex number into another number type?

Insert your answer here.

1. Convert integer variable ‘a’ with a value of 25 to a complex number.

Insert your code here.

1. Declare variable ‘x’ with a value of 112 and variable ‘y’ with a value of 230.
2. Create a code that will add the two variables together. What is the result?

Insert your code here.

1. Create a code that will divide the two variables together. What is the result?

Insert your code here.

1. Create a code that will find the average of the two variables. What is the result?

Insert your code here.

## Module 3: Python Simple Calculations and Collections

## **Module 3: Python Simple Calculations and Collections**

### Lesson 1: Simple Calculations Using Integers

* Identify what expression syntax includes.
* Write code for the following:
  + Addition
  + Subtraction
  + Division
  + Multiplication
  + Power calculation
* Identify the data type that a division produces in Python.

### Lesson 2: Simple Calculations Using Strings

* Identify how to access string elements in Python.
* Define string literals.
* Identify how strings are concatenated.
* Identify how to add a space between two words in a string.
* Write example code for combining a literal and a variable.
* Define slicing and give an example.
* Define indexing and give an example.
* Identify how to find the length of a variable and provide an example.

### Lesson 3: Lists

* Identify the types of data that can be contained in a list.
* Identify how lists are indicated in Python and write an example.
* Identify how strings are indicated in Python and write an example.
* Write example code for calling a list with a specific index number.
* Identify what happens when you call a list with an index number that is out of range.
* Write example code to concatenate string items at their index number using the + operator.
* Write example code to delete Items in a list using the remove() method.
* Write example code to add an item at the end of a list using the append() method.
* Write example code to add an item at the end of a list using the insert() method.
* Write example code to determine how many items a list holds using the len() function.

### Lesson 4: Python Collections

* Define the following collection data types in Python.
  + Lists
  + Tuples
  + Sets
  + Dictionaries
* Define the following keywords for data collection type.
  + Ordered
  + Unordered
  + Unchangeable
  + Allow duplicates
  + No duplicates
* Describe the difference between tuples and lists.
* Write example code for a tuple in Python.
* Identify how sets are indicated in Python.
* Describe how dictionaries store data values and how they are indicated in Python.
* Write example code for adding another key to an existing dictionary.

### Module Challenge: Calculations and Data Collections

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

To successfully complete this challenge, perform and answer the following:

1. Create basic calculations:
   1. Add two integers together.
      * What is the result?

Insert your answer here.

* 1. Subtract two integers.
     + What is the result?

Insert your answer here.

* 1. Divide two values. Have the result be an integer by using floor division.
     + What is the result?

Insert your answer here.

* 1. Create a variable ‘x’, have it’s value be 10+5. Print ‘x’.
     + What is the result?

Insert your answer here.

1. Create two strings and combine them using the + operator.
   1. What was the result?

Insert your answer here.

1. Create variable ‘y’ and have the value be ‘program’. Omit letters o and g. Print the result.

Insert your code here.

1. Create a list called ‘calendar\_months’.

Insert your code here.

* + 1. How long is the list?

Insert your answer here.

* + 1. Print only February, March, and April.

Insert your code here.

1. Create a sample list of your choice, using string variables.

Insert your code here.

1. Create a second sample list of your choice, using string variables.

Insert your code here.

1. Create a third list that concats the two sample lists you created above.

Insert your code here.

* 1. Remove one item from the third list.

Insert your code here.

1. Create a dictionary that has four keys and assign those keys values.

Insert your code here.

* 1. What is the length of the dictionary?

Insert your answer here.

## **Module 4: Python Control Flow and Statements**

## Module 4: Python Control Flow and Statements

### Lesson 1: IF ELSE ELIF Statements

* Identify the logical condition symbols that Python recognizes.
* Write sample code for an IF statement in Python.
* Identify what happens if an indentation is not used with an IF statement.
* If an expression is false and does not return the specified condition, what statement is returned?
* Write sample code for an ELIF statement in Python.
* Write sample code that contains IF, ELIF, and ELSE statements.

### Lesson 2: AND OR Statements

* Define the following statements:
  + AND
  + OR
  + NOT
* Write example code using the NOT statement.
* Write example code using the AND statement.
* Write example code using the OR statement.

### Lesson 3: LOOP Statements

* Define a loop.
* Define a while loop statement.
* Write example while loop code.

### Lesson 4: Functions and Arguments

* Define functions.
* Define user defined functions.
* Write sample function syntax.
* Identify the proper components that should be included when creating and writing functions.
* Define arguments and how to write them.

### Module Challenge: Control Flow

This challenge is going to assess your knowledge in flow and loop statements, as well as functions and arguments.

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a code with the following elements:
   * 1. variable a has a value of 65.
     2. variable b has a value of 150.
     3. IF variable is less than b.
     4. Based on the condition specified using the logical operator, print the specified statement.

Insert your code here.

* + 1. Did the condition return True or False?

Insert your answer here.

1. Create a code with the following elements:
   * 1. variable a has a value of 200.
     2. value b has a value of 50.
     3. IF statement says if b is greater than a, print the statement.

Insert your code here.

* + 1. ELIF statement says if a and b are equal, print the statement.

Insert your code here.

* + 1. ELSE statement says if the if statements and elif statements are false, then print the statements.

Insert your code here.

* + 1. Did the condition return True or False?

Insert your answer here.

1. Create code using the OR operator with the following elements:
   * 1. variable a has a value of 200.
     2. variable b has a value of 50.
     3. variable c has a value of 500.
     4. IF a is greater than b AND c is great than a, print the statement.

Insert your code here.

* + 1. Did the condition return True or False?

Insert your answer here.

1. Create a WHILE loop statement with the following elements:
   * 1. The above code is telling Python that WHILE variable a is less than 15, print the results.

Insert your code here.

* + 1. The increment of the loop result will stop after 1 gets to 15, using increments of one.

Insert your code here.

* + 1. The increment was set using ‘a += 1’.

Insert your code here.

* + 1. Run your code.

Insert your code here.

1. Create a for loop statement with the following elements:
   * 1. The value of the string is ‘good day sunshine’.
     2. Print the sequence for variable x.
     3. Give variable x the value of the string with the loop condition specified.
     4. Run your code.

Insert your code here.

## **Module 5: Python Anonymous Functions, Classes and Objects**

## Module 5: Python Anonymous Functions, Classes and Objects

### Lesson 1: LAMBDA Functions

* Define Lambda functions.
* Define anonymous functions.
* Write sample lambda syntax.
* Identify how functions should be written if needing a return statement.
* Identify the number of expressions and arguments that can exist in Lambda code.

### Lesson 2: Classes and Objects

* Describe the value and features of classes.
* Define objects and what they consist of.
* Identify the function that every class has.
* Write sample code for creating a class.

### Lesson 3: Scopes

* Define namespace.
* Define the scope concept, local.
* Define the scope concept, enclosed.
* Define the scope concept, global.
* Write code for the above scope concepts.

### Lesson 4: Modules

* Define a module in Python.
* Write code for recalling a module.
* Write code that adds to the module.
* Write code for recalling a module using an alias.

### Module Challenge: Functions, Classes and Objects

This challenge is going to assess your knowledge in lambda functions, classes, objects, scopes, and modules. You will be required to use Python for coding applications.

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a lambda function that takes one parameter (a) and return it.

Insert your code here.

1. Create a class called dataclass and assign variable x.

Insert your code here.

1. Create an object of dataclass called d1.

Insert your code here.

1. Use the d1 object to print the value of x.

Insert your code here.

1. Create the correct syntax to import a module named "datamodule".

Insert your code here.

1. Create the correct syntax to import only the customer1 dictionary from the “datamodule”.

Insert your code here.

1. Create a syntax to print your module as an alias.

Insert your code here.

1. Create a syntax that will print all variables and functions in the “datamodule” module.

Insert your code here.

1. Create a syntax importing only the employee1 dictionary of the "datamodule" module.

Insert your code here.

1. Create a syntax that will return all variables and functions in a module.

Insert your code here.

1. Create a syntax that will access a dictionary name “sampledictionary” from the “samplemodule”.

Insert your code here.

1. Create a syntax that will
   * 1. Return a variable using a local scope.

Insert your code here.

* + 1. Return a variable using a global scope.

Insert your code here.

1. Create a syntax that will return two separate values of the variable x both inside and outside the function.

Insert your code here.

* + 1. Variable x is declared outside the function with a value of 100. Print this result last.

Insert your code here.

* + 1. Variable x is declared inside the function named “second\_function” with a value of 500. Print this result first.

Insert your code here.

## **Module 6: Python File Handling**

## Module 6: Python File Handling

### Lesson 1: Open File Mode Arguments

* Identify the key function when working with files in Python.
* Identify the two parameters needed to open files.
* Formulate example syntax for opening files in Python.

Define the different mode arguments below:

* Read “r”
* Append “a”
* Write “w”
* Create “x”
* Identify the default mode that Python assumes if a mode argument is not used.
* Formulate example syntax for the read mode argument if you need to extract a string that contains all characters.
* Formulate example syntax for the read mode argument if you need to include a file path for a text file.

### Lesson 2: Reading Files

* Formulate example syntax using the readline() method.
* Identify what using the readline() method twice does.
* Formulate example syntax using the for argument and describe what this does.
* Formulate example syntax that closes Python files.
* Explain why you need to clean up the output when presenting data.
* Define line splitting.
* Write an example of a splitlines()statement.

### Lesson 3: Append Write Create Files

* Define appending a file.

Describe what the following append modes do:

* ‘a’
* ‘a+’
* Formulate example syntax for the append mode.
* Formulate example syntax for writing a file ovewrite.
* Explain why one should be caution when overwriting a file.
* Formulate example syntax for creating a new file using the open() method and one of the mode parameters.

Define the following Python operators:

* r
* rb
* r+
* rb+
* w
* wb
* w+
* wb+
* a
* ab
* a+
* ab+

### Lesson 4: Delete Files and JSON Types

* Write an example of syntax that removes a text file.
* Can files be retrieved once they are deleted?
* Write example syntax that checks if a file exists and then sets the code to delete it.
* Identify an alternative and quicker way to determine if a file exists.
* Define JSON
* Identify the method needed to read in JSON files.
* Describe what the json.load() does.
* Write an example json syntax.
* Identify the two ways to return objects using the loads() method.
* Identify when you will work with JSON files.

For each of the Python objects listed below, write in the JSON object.

|  |  |
| --- | --- |
| Python | JSON |
| dict |  |
| list |  |
| Str |  |
| None |  |
| In |  |
| Float |  |
| True |  |
| False |  |

### Module Challenge: Mode Arguments and File Handling

This challenge is going to assess your knowledge in file handling, mode arguments, and text files. You will be required to use Python for coding applications.

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

Use Python to create the correct syntax code below.

1. Using the read mode, create a syntax that will open a text file named, ‘samplefile’.

Insert your code here.

1. Create a syntax that will a string that contains all characters in the text file name, “samplefile”.

Insert your code here.

1. Create a syntax that will recall the first two lines of the text file name, ‘samplefile2’.

Insert your code here.

1. Create a syntax that will read in text file name, ‘samplefile2’ line by line.

Insert your code here.

1. Create a syntax to open a JSON file named, ‘jsonsample’.

* if the file does not exist set your print statement to return ‘file does not exist’.

Insert your code here.

1. Create a syntax that will overwrite all data in file name, ‘samplefile3’ to read ‘“All content is deleted”:

Insert your code here.

1. Create a syntax that will create a new text file called, ‘createfile.txt).

Insert your code here.

1. Before you close out of the interpreter, create a syntax that will close the text file name, ‘samplefile2’.

Insert your code here.

## **Module 7: Introduction to Pandas**

## Module 7: Introduction to Pandas

### Lesson 1: Importing Pandas and Series

* Define Pandas.
* Identify what Pandas does.
* Describe how to import Pandas.
* Describe the benefits of setting a series to your data.
* Identify what a series consists of.
* Identify the constructors that series class provides.
* Write example code that creates a basic series from a list.
* Write example code that sets index labels to the series.
* Write code to print a value of your choice in the index.

### Lesson 2: Dictionary, List Series

* Write code that uses keys and object values to create a series using a list index.
* Write code that creates a series from a dictionary and assigns values with a colon.
* Write code that extracts a few items from a dictionary using the pd.Series argument.
* Write code that uses the index () argument to assign the index structure for a list.
* Identify what NAN means.

### Lesson 3: Data Frames

* Define data frames.
* Discuss the difference between a series and a data frame.
* Identify what can make up a data frame.
* Identify how to create a dataframe using Pandas.
* Identify the steps needed to create datasets in Python.
* Identify the steps needed to create an empty frame.
* Write the code needed to create an empty frame.
  + Explain the code for creating an empty frame.
* Write example syntax for creating a set of data that holds employee information.
  + Explain the example syntax code for creating a set of data that holds employee information.
* Write example syntax for creating a dataframe from two different datasets.
  + Explain the example code for creating a dataframe from two different datasets.
* Write example syntax for using the loc attribute to locate and return one or more specific rows from a dataframe.

### Lesson 4: CSV and JSON Files

* Write the syntax for reading in a csv file into Python.
  + Explain what the syntax is telling Python.
* Write the syntax for loading a csv file using the file path name.
* Identify how to print the entire set of data when reading in a csv file, rather than just the first 5 rows.
* Write example syntax for reading in JSON files using Pandas.

### Module Challenge: Pandas, Datasets, and Dataframes

Congratulations on making it through this lesson! This challenge is going to assess your knowledge in importing pandas, creating lists, dictionaries, series, dataframes, and reading in files. You will be required to use Python for coding applications. Good luck!

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a syntax that imports pandas using an alias and create a series.
   1. Define up to 5 values in ‘x’.
   2. Call the series MySeries.
   3. What is the data type that was in the output?  
      Insert your code here.
2. Using the series you created above:
   1. What is the output of index[0]?
   2. What is the output of index[6]?
      1. Did you receive an error? If so, why?

Insert your code here.

1. Create a syntax that imports pandas using an alias and create a series:
   1. Define up to 5 values in ‘x’.
   2. Assign an index for each value in the syntax.

Insert your code here.

1. Create a syntax that imports pandas using an alias and then do the following:
   1. A list called, ‘simplelist’ with 10 numeric values.
   2. Print the list.
   3. What is the data type?

Insert your code here.

1. Create a syntax that imports pandas using an alias and then do the following:
   1. A dictionary called, ‘simpledictionary’ with 10 values.
   2. Print the list.
   3. What is the data type?

Insert your code here.

1. Create a syntax that imports pandas using an alias and create an empty dataframe.

Insert your code here.

1. Create a syntax that imports pandas using an alias and create:
   1. Two data series to make a dataframe called, ‘dataseriesframe.
   2. Use index to return data that is in index row 3.  
      Insert your code here.
2. If you had a csv file called, ‘samplefile’, create a syntax that:
   1. imports pandas as an alias.
   2. reads in the csv file.
   3. reads in the cs file path

Insert your code here.

1. Create a syntax that imports pandas using as alias that reads in an entire json file called, ‘entirefile.json’

Insert your code here.

## **Module 8: Introduction to NumPy**

## Module 8: Introduction to NumPy

### Lesson 1: NumPy Library

* Define NumPy and what it is used for.
* Define an array.
* Identify how to determine the rank of the array.
* Identify what defines the shape of an array.
* Describe how NumPy and Pandas are different.
* Identify the array class of numPy and what that means.
* Describe what homogenous data types means.
* Create code for a specified data type with an array.

Identify the meaning of the following Python single characters:

* i
* b
* u
* f
* c
* m
* M
* O
* S
* U
* V
* Identify how Python indicates a data type.
* Identify what happens when you mix multiple data types in one code.
* Identify how to convert a data type to another data type.
* Write code for a multidimensional array.

### Lesson 2: Array Shapes

* Write example syntax for a one-dimensional array.
* Write example syntax for a two-dimensional array.
* Write example syntax for a three-dimensional array.
* Describe why Python may convert an array to a list.
* Describe advantages of using arrays.
* Describe important things to know about lists over arrays.

### Lesson 3: Array Indexing

* Define array indexing.
* Write example code that shows how to index from a created array.
* Write example code that shows how to index and then add values together.
* Describe what negative indexing allows you to do and write sample code for this.
* Identify the difference between a copy and an array.
* Write example code for creating a copy of an original array.

### Lesson 4: Array Slicing

* Define slicing.
* Describe why you would slice.
* Describe what should be considered when slicing.
* Write example code that shows slicing an array element from an index to the end of the array.
* Write code that shows negative slicing.

### Module Challenge: NumPy Arrays and Indexing

This challenge is going to assess your knowledge in NumPy and arrays. You will be required to use Python for coding applications. You will be provided a worksheet

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a syntax that imports Numpy as np and do the following:

* elements of 1,2,3,4,5
* print the data type of the array

Insert your code here.

2. Create the same syntax as above but identify the data type should be a string.

Insert your code here.

3. Create the same syntax above but change the data type to an integer.

Insert your code here.

4. Create a 2D array using your values of choice, then do the following:

* Check the shape of the array.

Insert your code here.

5. Create an array with the elements of 20, 30, 40, 50, 60, 70:

* Insert the correct syntax for printing the number 50 from the array.
* Insert the correct syntax for pricing the number 20 from the array.
* Insert the correct slicing syntax to include all elements from the second item to (not including) the fifth item.

Insert your code here.

6. Create a syntax importing NumPy as np and do the following:

* Array values are 25, 35, 45, 55, 65
* Slice the array with everything from the second item to (not including) the sixth item.

Insert your code here.

7. Create a syntax using the array above and do the following:

* Create a copy and replace the first index with 105.

Insert your code here.

## **Module 9: Introduction to Visuals**

## Module 9: Introduction to Visuals

### Lesson 1: Import MATPLOTLIB

* Define Matplotlib and describe what it does.
* Describe how you use matplotlib and NumPy together.
* Write code for how you import the pyplot module of the matplot library.
* Write the code for importing NumPY and Matplotllib together.
* Describe what pyplot does and how it is used.
* Describe the following call signatures and data labels.
  + x,y
  + fmt
  + data
* Identify what you should keep in mind while working through visuals in Python.
* Identify the steps to successfully create a plot using Matplotlib.

### Lesson 2: Plotting

* Describe the steps for potting a diagram using matplotlib.
* Write code for plotting a line using matplotlib.pyplot as plt
* Write code for plotting points on a chart that coincide with the data elements but does not draw a connecting line.
* Write code for plotting an array with multiple points.
* Write code for plotting a graph that does not specify the x-axis points.

### Lesson 3: Scatter Plot

* Write example code for creating a scatter plot using the scatter() function.
* Write example code for comparing data from two different datasets using the scatter() function syntax.

### Lesson 4: Bar Pie Charts

* Define the bar()function and describe what it does.
* Describe the steps to use the bar()function.
* Write example code using the bar()function.
* Write example code that created a bar chart with horizontal columns.
* Write example code that creates a pie chart.

### Module Challenge: Creating Charts Using MatPlotLib

This challenge is going to assess your knowledge in visual creation, plotting, and different types of charts. You will be required to use Python for coding applications. Good luck!

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a syntax that imports matplotlib and NumPy, then do the following:

a. Create parameters for an array on the x-axis (xpoints).

b. Create parameters for an array on the y-axis (xpoints).

c. Plot both arrays.

Insert your syntax and plot chart here.

1. Using the syntax above, remove the line and include only markers for the x and y points.

Insert your syntax and plot chart here.

1. Create a syntax that imports matplotlib and NumPy, then do the following:

a. Create parameters for x.

b. Create parameters for y.

c. Plot the arrays as a scatter plot.

Insert your syntax and plot chart here.

1. Using the syntax in #3, create a second set of array data, then plot as a comparison scatter plot.

a. Based on what you see, what is the data telling you?

Insert your response here.

1. Create syntax that imports matplotlib and NumPy, then do the following:

a. Create parameters for x.

b. Create parameters for y .

c. Plot the arrays as a bar chart.

Insert your syntax and plot chart here.

1. Create syntax that imports matplotlib and NumPy, then do the following:

a. Create parameters for x.

b. Create parameters for y.

c. Plot the arrays as a pie chart.

Insert your syntax and plot chart here.

## Module 10: Formatting Visuals

## **Module 10: Formatting Visuals**

### Lesson 1: Markers and Labels

* Define markers and what they are used for.
* Describe how to add markers to a chart in Python.
* Write example code for adding markers to a chart in Python.
* Identify the syntax for the following colors in Python.
  + Red
  + Green
  + Blue
  + Cyan
  + Magenta
  + Yellow
  + Black
* Write example code for making a plot with a green dotted line.
* Describe how to change the size and color of markers in a chart using Python.
* Write code for creating larger markers and a different marker face color on a chart.
* Describe how to add labels to a chart in Python.
* Write code for adding labels to a chart in Python.

### Lesson 2: Formatting Color Schemes

* Describe how to create a bar chart with bars of a variety of colors.
* Write example code for creating a bar chart with a variety of colors and is labeled.
* Write example code for creating a bar chart with labels, colors, and a title.

### Lesson 3: Creating Multiple Plots

* Define the subplot() function and describe what it does.
* Describe what is needed to use the subplot() function.
* Write example code for two plots generated from a single syntax with parameters of your choice.
  + Identify the parameters.
* Add titles to the above code.

### Lesson 4: Color Bars

* Describe why you would add a color bar to a scatter plot chart.
* Describe how to add color bars to charts in Python.
* Write example code for adding a color bar to a chart in Python.

### Module Challenge: Formatting Charts

This challenge is going to assess your knowledge in how to format your charts and plots for strong display of your visuals. You will be required to use Python for coding applications. For all syntax remember to import matplotlib and NumPy. Good luck!

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a line plot that includes the following parameters:
   1. ypoints with declared array values.
   2. Include markers for each point.
   3. Set the marker shape to 30.
   4. Set the marker face color to red.

Insert your line plot here:

1. Create a bar chart that includes the following parameters:
2. Declare variable x as an array of values.
3. Declare variable y as an array of values.
4. Assign a color to each bar.
5. Add a label for the x axis.
6. Add a label for the y axis.
7. Add a title for your chart.

Insert your bar chart here.

1. Using the subplot() function, combine two arrays and chart them within the same syntax:
2. For plot #1:

* Create array x and declare values.
* Create array y and declare values.
* Make this the first subplot and set the rows and columns.

1. For plot #2:

* Create array x and declare values.
* Create array y and declare values.
* Make this the first subplot and set the rows and columns.
* Then, plot both in the same print result.

Insert your plot charts here.

     4. Create a scatter plot with a color bar and include the following:

1. Declare variable x as an array of values.
2. Declare variable y as an array of values.
3. Set the color parameters for your array.
4. Set which plot you will be using and the parameters.
5. Plot the color bar.

Insert your scatter plot chart here.

## Module 11: Python Data Cleaning

## **Module 11: Python Data Cleaning**

### Lesson 1: Missing Data

* Explain why objects will be excluded when working with data in Pandas.
* Define NaN
* Identify what to do if you choose to include NaN values when presenting.
* Identify the argument needed to fill in the missing data and avoid discarding important data.
  + Explain what this will do.
* Identify the import argument needed to import the nan library from NumPy.
  + Describe how nan functions and what it can do.
* Write code for the data output with NaN data.
* Identify the argument needed to tell Python to drop the missing data.
* Write example code for Python to drop missing data and clean the data series.
  + Break down what the code is telling us.
* Identify how to return the cleaned data output.

### Lesson 2: Data Transformation

* Identify what the duplicate() argument returns in a dataframe.
* Describe the options you have with duplicate dataframes and the pros and cons of each.
* Write example code for dropping duplicate data.
* Describe why it is important to include command line comments to syntax when dropping duplicate data.

### Lesson 3: Transforming Data

* Identify the arguments needed to convert all data characters in a data frame to be lowercase.
* Write example code for converting data characters to lowercase.
* Describe how to create a new column of data within a dataframe.
* Describe how to map the data to the new column category.
* Write example code for creating a new column and mapping data to that new column.

### Lesson 4: SEABORN

* Describe what you can do using Seaborn in Python.
* Describe how to use seaborn in Python.
* Identify how to import Seaborn into Python and what is best practice.
* Write example code for basic distribution plot syntax.
* Describe what a normal distribution is.
* Identify what the mean and standard deviation are.

### Module Challenge: Transforming Data

This challenge is going to assess your knowledge in how to clean data and run a distribution plot using Seaborn. You will be required to use Python for coding applications. For all syntax remember to import matplotlib, NumPy, and Pandas were necessary.

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a syntax that will return all NaN values or missing values in a given data series or fram.e Assume the values are string.

Insert your syntax here:

1. Create a syntax that will fill in missing values to zero.

Insert your syntax here:

1. Create a syntax that will identify where duplicate values are within a dataframe,

Insert your syntax here:

1. Create a syntax that will drop duplicates from the data output.

Insert your syntax here:

1. Create a dataframe called ‘DataTransformation’ with the following information:

a. Food type values using string data.

i. use both upper and lower case.

b. Calorie data for each food type.

c. Print the output.

d. Use the string lower argument, make all food type values lower case and name that output ‘lowercasedata’.

e. Create a map category for the food type.

f. Map the category to the food type that was set to lowercase.

g. Print the new output.

Insert your output here:

1. Using Seaborn:

a. Create a distribution plot with 2, random values under 50.

b. Print the plot.

c. Now, print the plot without the histogram.

Insert your plots here:

## Module 12: Python Machine Learning

## **Module 12: Python Machine Learning**

### Lesson 1: Getting Started

* Define machine learning.
* Describe how businesses are using machine learning.
* Identify an example of an array.

Define the following data types:

* Numerical
* Categorical
* Ordinal

Define the following statistical values:

* Mean
* Median
* Mode
* Describe how to use NumPy to calculate the mean of a dataset.
* Write example code to calculate the mean of a dataset.
* Describe how to use NumPy to calculate the median of a dataset.
* Write example code to calculate the median of a dataset.
* Describe how to use scipy to calculate the mode of a dataset.
* Write example code to calculate the mode of a dataset.

### Lesson 2: Standard Deviation

* Define standard deviation.
* Describe how to find the standard deviation using std() method.
* Write example code to find standard deviation in Python.
* Describe how to plot a distribution using the sns.distplot() method.
* Write example code for plotting a distribution using the sns.distplot() method.
* Define variance.
* Describe the relationship between variance and the standard deviation.
* Explain what variance tells us about performance.
* Write example code that uses the var() method to find variance.

### Lesson 3: Percentiles and Data Distribution

* Define percentiles in statistics.

Describe the following quartiles:

* Quartile 1
* Quartile 2
* Quartile 3
* Write example code for using the NumPy percentile() method to find a specified percentile.
* Write example code using the percentile() method that shows how the data compares against the mean.

### Lesson 4: Data Distribution

* Describe what the random() method allows you to do.
* Explain why when using the random() method a different result will return each time.
* Write example code for using the random.choice() method.
* Describe what the random.random() syntax is used for.
* Write code for creating a random large dataset using random.uniform().
* Write example code for creating a histogram in addition to a random dataset using the random.uniform and plt.hist() methods.
* Write example code that creates a normal distribution by using the random.normal() argument.
* Write example code that creates a scatter plot using the scatter() argument.

### Module Challenge: Cleaning and Transforming Data While Plotting

Congratulations on making it through the machine learning lesson! This challenge is going to assess your knowledge in how to perform statistical analysis using raw data. You will be required to use Python for coding applications. For all syntax remember to import matplotlib, NumPy, and Pandas were necessary.

**NOTE: You will be asked to submit these responses to your instructor on a document that is provided in your online classroom.**

1. Create a numerical array and do the following:

a. Calculate the mean.

b. Calculate the mode.

c. Calculate the median.

Insert your syntax here:

1. Create a numerical array and do the following:

a. Calculate the standard deviation.

b. Calculate the variance

c. Plot the data. What does the plot tell you about the data?

Insert your syntax and answer here:

1. Create a random list of data that ranges between 0 and 1000.

Insert your syntax here:

1. Create a random numerical list that includes the following:

a. Float point numbers between 0 and 15.

b. Include 250 values.

c. Create a plot of the list data to include 5 bars.

Insert your syntax and plot here:

1. Create a syntax that performs the following:

a. Creates an array of random values and calls it x. Values should be floating points from 5.0 to 1.0 and 1000 values.

b. Creates an array of random values and calls it y. Values should be floating points from 10.0 to 5.0 and 1000 values.

Insert your syntax here: