Python Lessons

Introduction:
- ✔ Python is a popular programming language.
- ✔ It was created by "Guido Van Rossum".
- ✔ Python programming language was released in year 1991.
- ✔ It is used for: web development, software development, mathematics and system scripting.

Uses:
- ✔ Python can be used with Database Systems.
- ✔ Python can also read and modify files.
- ✔ Python can be used to handle big data and perform complex mathematics.
- ✔ Python can be used for repaid prototyping or for production ready software development.
- ✔ Python can be used alongside software to create workflows.
- ✔ Python can be use on server side to create web applications.

Why to Choose Python:
- ✔ Python works on many platforms. Like Windows, Mac and Linux.
- ✔ Python has a simple syntax similar to English language.
- ✔ Python executes on interpreter system, it means that code be executed as they have been written.
- ✔ Python has syntax that allows developers to write the program with fewer lines than some other programming languages.
- ✔ Python can be treated in procedural way, an object oriented way and or a functional way.
- ✔ It is possible to write python code by using Integrated Development Environment (IDE).
- ✔ Some of the popular Python IDEs are: PyCharm, Eclipse and Netbeans.
- ✔ Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- ✔ Python relies on indentations, using white-space to define scope such as scope of loops, functions and classes, other programming languages often use curly brackets for this purpose.
- ✔ Python uses new line to complete a command as opposed to other programming languages which often use semicolons or parentheses.

Python Syntax:
- ✔ Python syntax can be executed by writing directly in the command line.
- ✔ Python syntax can also be executed by creating a python file on the server.
- ✔ Using the *.py extension and executing on the command line.
- ✔ Indentation refers to the spaces at the beginning of the code line.
- ✔ In other programming languages the indentation in code is for readability purpose.
- ✔ The indentation in python is very important.
- ✔ Python uses indentation to indicate a block of code.
- ✔ Python will display an error if you forgot the indentation in the block of code.
- ✔ The number of spaces has to be at least one space.
- ✔ You have to use same number of spaces in the same block of code otherwise python will give you an error.

Python Lessons

Python Variables:
- ✔ Variables are created when you assign a value to it.
- ✔ Python has no command for declaring a variable.

Python Comments:
- ✔ Python comments starts with (#) symbol.
- ✔ After (#) symbol every line of code will be turned into python comments.
- ✔ Python has commenting capability for the purpose of in-code documentation.
- ✔ Comments can be used to explain Python code.
- ✔ Comments can be used to make the code more readable.
- ✔ Comments can be used to prevent execution when testing code.
- ✔ Comments can be placed at the end of a line, and Python will ignore the rest of the line.
- ✔ A comment does not have to be text that explains the code, it can also be used to prevent Python from executing code.
- ✔ Python does not have really a syntax for multi-line comments.
- ✔ To add a multi line comment you could insert a (#) hash for each line.
- ✔ You can also use multiline string.
- ✔ Python will ignore string literals that are not assigned to string variables.
- ✔ You can add a multiline string ("""") triple quotes in your code.
- ✔ As long as the string is not assigned to a variable, Python will read the code, but then ignore it, you have made a multiline comment.

Python Variables:
- ✔ Variables are containers for storing data values.
- ✔ Python has no command for declaring a variable.
- ✔ A variable is created the moment you first assign a value to it.
- ✔ Variables do not need to be declared with any particular type and can even change after they have been set.
- ✔ If you want to specify the data type of a variable this can be done with casting.
- ✔ You can get the type of the variable with the type() function.
- ✔ String variables can be declared either by using single or double quotes.
- ✔ Variables name are case-sensitive.
- ✔ The variables can have short names (like x or y) or can have descriptive name (like Age and Gender).

Rules for Python Variables:
- ✔ A variable name must starts with a letter or the underscore character.
- ✔ A variable name cannot start with a number.
- ✔ A variable name can only contain alpha-numeric characters and underscores.
- ✔ Variable name are case-sensitive. Age and AGE are different variables.
- ✔ Variables name with more than one word can be difficult to read. There are several techniques you can use to make them more readable.
- ✔ Camel Case: Each Word, except the first, starts with the capital letter.
- ✔ Pascal Case: Each word starts with capital letter.
- ✔ Snake Case: Each word is separated by an underscore character.
- ✔ Python allows you to assign values to multiple variables in one line. Make sure the number of variables matches the number of values, or else you will get an error.
- ✔ You can also assign single same value to multiple variables.

Python Lessons

- ✔ If you have a collection of values in a list, tuple etc. Python allows you to extract the values into variables. This is called unpacking.
- ✔ The Python print statement is often used to output variables.
- ✔ To combine text and variables Python uses (+) plus operator.
- ✔ You can also use (+) plus operator to add a variable to another variable.
- ✔ For numbers in Python (+) plus operator will work as a addition operator.
- ✔ Variables that are created outside of a function are known as global variables.
- ✔ Global variables can be used by anyone both inside of a function or outside of it.
- ✔ If you create a variables inside of a function then it scope will be local, and hence it will be called local variable.
- ✔ To create a global variable inside a function you can use the global keyword.
- ✔ Also use the global keyword if you want to change a global variable inside a function.

Python Data Types:
- ✔ Built-in Data Types:
- ✔ In programming, data type is an important concept.
- ✔ Variables can store data of different types and different type can do different things.
- ✔ Python has the following data types built-in by default in these categories.
- ✔ Text Type: str()
- ✔ Numeric types: int(), float(), complex()
- ✔ Sequence types: list, tuple and range.
- ✔ Mapping type: dict.
- ✔ Set types: set, frozenset
- ✔ Boolean type: bool
- ✔ Binary types: bytes, bytearray and memoryview
- ✔ You can get the data type of any object by using the type() function.
- ✔ In Python, the data type is set when you assign a value to a variable.
- ✔ If you want to specify the data type you can use the constructor of the specified data type.

Pythons Numbers:
- ✔ There are three types of numerics are there:
- ✔ Int.
- ✔ Float.
- ✔ Complex.
- ✔ Variables of numerics are created when you assign a value to them.
- ✔ To verify the type of an object in Python, use the type() function.
- ✔ Int or Integer, is a whole number positive or negative without decimals of unlimited length.
- ✔ Float or Floating Point Number is a number, positive or negative, containing one or more decimals.
- ✔ Float can also be a scientific numbers with an "e" to indicate the power of 10.
- ✔ Complex numbers are written with a "j" as the imaginary part.
- ✔ You can convert from one type to another with the int(), float() and complex() methods.
- ✔ You cannot convert complex data type variable to another data type variable.
- ✔ Python has a built-in module called random that can be used to make random numbers.
- ✔ Python is an Object Oriented Language, and as such it uses classes to define data types including its primitive types.
- ✔ Casting in Python is therefore is done using constructor functions. Inf(), float(), complex() and str().

Python Lessons

Python Strings:
- ✔ Strings in Python are surrounded by either single quotation marks or double quotations mark.
- ✔ You can display a string literal with the print() function.
- ✔ Assigning a string to a variable is done with the variable name followed by an equal sign and the string.
- ✔ You can assign a multiline string by using the three quotations marks.
- ✔ Like many other programming languages strings in Python, are arrays of bytes representing unicode characters.
- ✔ Python does not have a character data type, a single character is simply a string with a length of one.
- ✔ Square brackets can be used to access elements of the String.
- ✔ Strings are arrays, we can loop through characters in a string.
- ✔ To get the length of the string, use len() function.
- ✔ To check if a certain phrase or character is present in s string, we can use in keyword.
- ✔ To check if a certain phrase or a character is not present in a string, we can use the keyword not in.
- ✔ You can return a range of characters by using the slice syntax. Just specify the start index and the end index, which will be separated by a colon to return a part of the string.
- ✔ By leaving out the start index the range will start at the first character.
- ✔ By leaving out the end index, the range will go to the end.
- ✔ Use the neagative indexes to start the slice from the end of the string.
- ✔ Python has a set of built-in methods that you can use on string.
- ✔ The lower() method returns the string in lower case.
- ✔ The upper() method returns the string in upper case.
- ✔ White-space is the space before and or after the actual text and very often you want it to remove this white-space.
- ✔ The strip() methods removes any white-space from the beginning or the end.
- ✔ The split() method returns a list where the text between the specified separator becomes the list items.
- ✔ To concatenate or combine two strings you can use the (+) plus operators.
- ✔ We can combine strings and numbers by using format() method.
- ✔ The format() method takes the passed arguments, format them, and places them in the string where the placeholders {} are presents.
- ✔ The format() methods takes unlimited arguments and are placed into the respective placeholders.
- ✔ You can use index numbers {0} to be sure the arguments are placed in the correct placeholder.
- ✔ To insert characters that are illegal in a string, use an escape character.
- ✔ An escape character is a backslash \ followed by the character you want to insert.
- ✔ An example of illegal character is double quote inside the string that is surrounded by double quotes.
- ✔ You will get an error if you use double quotes inside a string that is surrounded by double quotes. To fix this problem use the escape character \".
- ✔ Other characters we can use are:
- ✔ Single Quote.
- ✔ Backslash.
- ✔ New line.

Python Lessons

- ✔ Carriage Return.
- ✔ Tab.
- ✔ Backspace
- ✔ Form Feed.
- ✔ Octal value.
- ✔ Hexa Value.
- ✔ Python has a set of string method built-in method that you can use on string.
- ✔ All string methods returns new value. They do not change the original string.

Python Booleans:
- ✔ Boolean represents one of two values: True or False.
- ✔ In programming you often need to know if an expression is True or False.
- ✔ You can evaluate any expression in Python, and get one of two answers: True or False.
- ✔ When you compare two values, the expression is evaluated and Python returns the Boolean answer.
- ✔ When you run a condition in an if statement, Python returns True or False.
- ✔ The bool() function allows you to evaluate any value, and give you True or False in return.
- ✔ Almost any value is evaluated to True if it has some sort of content.
- ✔ Any string, is True except an empty string.
- ✔ Any number is True, except 0.
- ✔ Any List, Tuple, Set and Dictionary are True except empty one.
- ✔ Python has also many built-in functions that return a boolean value like the isinstance() function which can be used to determine if an object is of a certain data type.

Python Operators:
- ✔ Operators are used to perform operations ion variables and values.
- ✔ Python divides the operators in the following groups.
- ✔ Arithmetic Operators.
- ✔ Assignment Operator.
- ✔ Comparison Operators.
- ✔ Logical Operators.
- ✔ Identity Operators.
- ✔ Membership Operators.
- ✔ Bitwise Operators.

Python Arithmetic Operators:s
- ✔ Arithmetic operators are used with numeric values to perform common mathematical operations.
- ✔ Arithmetic Operators are:
- ✔ Addition (+).
- ✔ Subtraction (-).
- ✔ Multiplication (*).
- ✔ Division (/).
- ✔ Modulus (%).
- ✔ Exponential (**).
- ✔ Floor Division (//).

Python Lessons

Python Assignment Operator:
- ✔ Assignment operator are used to assign values to variables.
- ✔ Assignment operators are:
- ✔ Equal-to operator (=).
- ✔ (+=).
- ✔ (-=).
- ✔ (*=).
- ✔ (/=).
- ✔ (&=).
- ✔ (//=).
- ✔ (**=).
- ✔ (&=).
- ✔ (|=).
- ✔ (^=).
- ✔ (>>=).
- ✔ (<<=).

Python Comparison Operator:
- ✔ Comparison operators are used to compare two values.
- ✔ Comparison operators are:
- ✔ Equal (==).
- ✔ Not Equal (!=).
- ✔ Greater Than (>).
- ✔ Less Than (<).
- ✔ Greater Than Equal To (>=).
- ✔ Less Than Equal-to (<=).

Python Logical Operators:
- ✔ Logical operators are used to combine conditional statements.
- ✔ Logical Operators are:
- ✔ and.
- ✔ Or
- ✔ not

Python Identity Operators:
- ✔ Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with same memory location.
- ✔ Identity Operators are:
- ✔ is.
- ✔ is not.

Python Membership Operator.
- ✔ Membership operators are used to test if a sequence is present in an object.
- ✔ in.
- ✔ not in.

Python Lessons

Python Bitwise Operator:
- ✔ Bitwise operator is used to compare binary numbers.
- ✔ Bitwise operators are:
- ✔ Bit and (&)
- ✔ Bit or (|)
- ✔ Bit not (~)
- ✔ Bit xor (^)
- ✔ Zero Fill Left Shift (<<).
- ✔ Zero Fill Right Shift (>>).

Python Lists:
- ✔ Lists are used to store multiple items in a single variable.
- ✔ Lists are one of four built-in data types in Python used to store collections of data, the other three are Tuple, Set and Dictionary, all with different qualities and usage.
- ✔ Lists are created using square brackets.
- ✔ Lists items are ordered, changeable and allow duplicate values.
- ✔ List items are indexed, the first item has index [0], the second item has index [1] etc.
- ✔ Lists items can be ordered and cannot be changed because list items have a defined order.
- ✔ If you add new items to the list, the new item will be placed at the end of the list.
- ✔ There are some list methods, that will change the order but in general, the order of the items will not change.
- ✔ Lists in python are changeable, we can add, edit and even remove an item from the list after it was created.
- ✔ To get the list items count use the len() method.
- ✔ List items can be of any data type.
- ✔ Lists are defined as object with the data type "list"
- ✔ It is also possible to use the list() constructor when creating a new list.

Python Collections:
- ✔ Python Programming Language has four (4) collections data types. They are:
- ✔ List: Liss are ordered, changeable and can contain duplicate values.
- ✔ Tuple: Tuple are ordered but unchangeable and allows duplicate values.
- ✔ Set: Set are unordered and unindexed and cannot have duplicate values.
- ✔ Dictionary: Dictionary is ordered, changeable and cannot have duplicated values.
- ✔ As of Python version 3.7, dictionaries are ordered.
- ✔ In Python version 3.6, dictionaries are unordered.
- ✔ List Items are indexed and accessed by an index number. The first item has index 0.
- ✔ Negative indexing means start from the end. -1 refers to the last item, -2 refers to the second last item.
- ✔ Range can be specified by specifying start and end ranges, Python will return a new list within the defined ranges.
- ✔ Check the specified item in list you have to used in keyword.
- ✔ To change the value of a specific index, just mention that index in square brackets and then assign new value to that index.
- ✔ If you want to change some list items that create a new list with the specific index ranges.
- ✔ If you add more value to the specified index range then additional values will be added after range value and previous value will be moved further.

Python Lessons

✔ The length of list will be changed when the number of items is greater than the specified after changing some items.
✔ The insert() method will insert new item at the specified index of the list.
✔ The append() method will add new value to the end of the list. Aka new index will be created.
✔ To merge one list into another list you can use extend() method for that purpose. Elements will be added to the end of the extending list.
✔ The exend() method can handle any ilterable object like, Set, List, Dictionary and Tuple.
✔ To remove an item from list you can use remove() method. This method will remove a specified item.
✔ The pop() method will remove the specified index from the list. If you forget to specify the index number, the the method pop() will remove the last index from the list.
✔ You can use del keyword to perform the same operation like pop() method does. Del keyword can also format the list completely.
✔ The clear() method will clear all the elements from the list, no items will be left after clear() method executed.
✔ You can loop the list by using for loop. You can use range() and len() function to loop through index number.
✔ You can also use while loop to iterate the list.
✔ List Object have a sort() method, which sorts the list in alphabetically.
✔ To sort in reverse pass the parameter reverse with value true in sort() method.
✔ You can also customise the sort() funtion with you own function, simply use key parameter with you r function in sort() method. Then sort() method with execute used defined function with the implementation for sorting.
✔ Sort() method is case-sensitive by default. We can use functions as key to sort() method. Before sorting.
✔ Reverse() method will reverse the sorting from default ordering.

Python File Handling:
✔ File handling is an important part of any web application.
✔ Python has several functions for creating, reading, updating and deleting files.
✔ Open() function is the most commonly used function in Python language for files.
✔ Open() function requires two parameters which are filename and writing mode.
✔ The four types of modes for files are:
✔ r: open a file for reading purpose only. It also return an error if file does not exist.
✔ X: create a new file if it does not exist, error if already exist.
✔ A: update the file if exist and if not, it will create a new file and then update the file.
✔ W: write to a file if it exist or create a new file.
✔ T: text mode it is used in binary mode.
✔ B: images in binary mode.
✔ If you just want to read a file then only filename is efficient for reading purposes only.
✔ File should be present on the same folder as python file are on, or you have to specify the full file path to be read.
✔ Open() function returns the file object which contain read() method for reading.
✔ You can also pass number of paragraphs to be read as the parameter to the read() method.
✔ There is another function which read only first line in the file, this function is readline().
✔ Loop through the file to read file line by line by readline() method.
✔ Close() function closes any opened file, it is good practice to close the opened file.

Python Lessons

- ✔ Closing a file is important because in buffering, the changes you have made into the file cannot be appeared if file is not closed.
- ✔ You can write to a file if you use a or w mode in open() method. A mode will append to the next line while w will overwrite the file.
- ✔ To remove a file you have to import the os module and the the remove() method from that module.
- ✔ You can also remove the entire empty folder by using rmdir() method of the os module.

List Comprehension:
- ✔ List comprehension provides a shorter syntax if you want a new list among a given list. Without list comprehension you have to write for loo with condition statement inside.

Python MySQL:

Refs:

Continue From Python Copy Lists Items:
https://www.w3schools.com/python/python_lists_copy.asp

Python Files: https://www.w3schools.com/python/python_file_handling.asp

Python MySql: https://www.w3schools.com/python/python_mysql_getstarted.asp