**Week 2: SS1, Python Class**

**Topics: *Syntax, Comment, Variables, Data type, numbers, casting, string and Booleans***

**Syntax**

Syntax refers to the rules that define the structure of a language.

Python syntax can be executed by writing directly in the Command Line:

**Example**:

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

**Example**:

**Python Indentation**

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

**Example**:

# Comments

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.

Example:

## Variables

Variables are containers for storing data values.

In python, a variable is created the moment you first assign a value to it.

**Example**:

**Built-in Data Types**

In programming, data type is an important concept.

Variables can store data of different types, and different types 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, range |
| Mapping Type: | dict |
| Set Types: | set, frozenset |
| Boolean Type: | bool |
| Binary Types: | bytes, bytearray, memoryview |
| None Type: | NoneType |

|  |  |  |
| --- | --- | --- |
| **Example** | **Data Type** |  |
| x = "Hello World" | str |  |

|  |  |  |
| --- | --- | --- |
| x = 20 | int |  |
| x = 20.5 | float |  |
| x = 1j | complex |  |
| x = ["apple", "banana", "cherry"] | list |  |
| x = ("apple", "banana", "cherry") | tuple |  |
| x = range(6) | range |  |
| x = {"name" : "John", "age" : 36} | dict |  |
| x = {"apple", "banana", "cherry"} | set |  |
| x = frozenset({"apple", "banana", "cherry"}) | frozenset |  |
| x = True | bool |  |
| x = b"Hello" | bytes |  |
| x = bytearray(5) | bytearray |  |
| x = memoryview(bytes(5)) | memoryview |  |
| x = None | NoneType |  |

**Python Numbers**

There are three numeric types in Python:

* int
* float
* complex

Variables of numeric types are created when you assign a value to them:

## Note: Complex Numbers

Complex numbers are written with a "j" as the imaginary part:

Example = 9+5j

## Random Number

Python does not have a random() function to make a random number, but Python has a built-in module called random that can be used to make random numbers:

**Example**:

**Specify a Variable Type**

There may be times when you want to specify a type on to a variable. This can be done with casting. Python is an object-orientated language, and as such it uses classes to define data types, including its primitive types.

Casting in python is therefore done using constructor functions:

* int() - constructs an integer number from an integer literal, a float literal (by removing all decimals), or a string literal (providing the string represents a whole number)
* float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
* str() - constructs a string from a wide variety of data types, including strings, integer literals and float literals

## Strings

Strings in python are surrounded by either single quotation marks, or double quotation marks.

'hello' is the same as "hello".

You can display a string literal with the **print()** function:

Example:

## Boolean Values

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:

Example:

**Class Work**

Create the first variable and assign a value of 5.7 to it, create the second variable and assign a value of “9.9” to it

Create the third variable and assign a value of **multiplication of first variable and second variable**

Use the in build print() function to display your result in the console.

**Assignment**

Create the first variable and assign a value of 5 to it, create the second variable and assign a value of “5” to it

Create the third variable and assign a value of **addition of first variable and second variable**

Use the in build print() function to display your result in the console.