What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Python Program Example :-

If-else.py :-

```
a = 33
b = 33
```

```
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

elseif.py file code:-

```
bike="yamaha"
if bike=="hero":
    print("bike is hero")
elif bike=="suzuki":
    print("bike is suzuki")
elif bike=="yamaha":
    print("bike is yamaha")
else:
    print("please choose correct answer")
```

output:-

bike is Yamaha

For loop example :-

forloop1.py

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

output:-

apple

banana

cherry

forloop2.py example :-

```
for x in range(2, 30, 3):
  print(x)
```

output:-

2

5

8

11

14

17

20

23

26

29

whileloop1.py Example :-

```
i = 1
while i < 6:
  print(i)
  i += 1
```

Output:

1

2

3

4

5

Python function example:-

myfunction.py

```
def my_function():
  print("Hello from a function")

my_function()
```

output:-

Hello from a function

Function Example with Arguments:-

myfunction1.py:-

```
def my_function(a, b):
  c=a+b
  print("addition of a and b =",c)

my_function(5,8)
```

output :-

addition of a and b = 13

Python Lists

List Items

List 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.

Listexample.py :-

```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```

output :-

['apple', 'banana', 'cherry']

Tuple Items

Tuple items are ordered, unchangeable, and allow duplicate values.

Tuple items are indexed, the first item has index `[0]`, the second item has index `[1]` etc.

Ordered

When we say that tuples are ordered, it means that the items have a defined order, and that order will not change.

Unchangeable

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

Allow Duplicates

Since tuples are indexed, they can have items with the same value:

tupleexample.py:-

```
thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)
```

output:-

('apple', 'banana', 'cherry', 'apple', 'cherry')

Set Items

Set items are unordered, unchangeable, and do not allow duplicate values.

Unordered

Unordered means that the items in a set do not have a defined order.

Set items can appear in a different order every time you use them, and cannot be referred to by index or key.

Unchangeable

Set items are unchangeable, meaning that we cannot change the items after the set has been created.

Once a set is created, you cannot change its items, but you can remove items and add new items.

Duplicates Not Allowed

Sets cannot have two items with the same value.

setexample.py :-

```
thisset = {"apple", "banana", "cherry", "apple"}

print(thisset)
```

output:-

{'apple', 'banana', 'cherry'}

Dictionary Items

Dictionary items are ordered, changeable, and do not allow duplicates.

Ordered or Unordered?

As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.

When we say that dictionaries are ordered, it means that the items have a defined order, and that order will not change.

Unordered means that the items do not have a defined order, you cannot refer to an item by using an index.

Changeable

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

Duplicates Not Allowed

dictionaryexample.py :

```
thisdict =  {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict)
```

output:-

{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}