**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

Ex:-

print("Hello, World!")

**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.

Ex:-

#This is a comment  
print("Hello, World!")

**Multiline Commants:-**

#This is a comment  
#written in  
#more than just one line  
print("Hello, World!")

Variables

Variables are containers for storing data values.

Creating Variables

Python has no command for declaring a variable.

A variable is created the moment you first assign a value to it.

Example

x = 5  
y = "John"  
print(x)  
print(y)

**Variable Names:**

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total\_volume). Rules for Python variables:

* A variable name must start 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 (A-z, 0-9, and \_ )
* Variable names are case-sensitive (age, Age and AGE are three different variables)
* A variable name cannot be any of the [Python keywords](https://www.w3schools.com/python/python_ref_keywords.asp).

Ex:-

myvar = "John"  
my\_var = "John"  
\_my\_var = "John"  
myVar = "John"  
MYVAR = "John"  
myvar2 = "John"

**Also we assign Multiple values:**

x, y, z = "Orange", "Banana", "Cherry"  
print(x)  
print(y)  
print(z)

**Global Variables:**

Variables that are created outside of a function (as in all of the examples above) are known as global variables.

Global variables can be used by everyone, both inside of functions and outside.

**Ex:-**

x = "awesome"  
  
 def myfunc():  
   print("Python is " + x)  
  
 myfunc()

**Python 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 |

**Getting the data type:-**

You can get the data type of any object by using the type() function:

**Ex:-** x = 5  
 print(type(x))

**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:

**Ex:-**

x = 1    # int  
y = 2.8  # float  
z = 1j   # complex

**Python 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:

Ex:-

Print(“hello”)

Print(‘hello’)

**Strings are arrays:**

Like many other popular programming languages, strings in Python are arrays of bytes representing unicode characters.

However, Python does not have a character data type, a single character is simply a string with a length of 1.

Square brackets can be used to access elements of the string.

**Ex:-**

a = "Hello, World!"  
print(a[1])

**Looping through Strings:**

Since strings are arrays, we can loop through the characters in a string, with a for loop.

**Ex:-**

for x in "banana":  
   print(x)

**String Length:**

To get the length of a string, use the len() function.

Ex:-

a = "Hello, World!"  
print(len(a))

# **Slicing Strings:**

You can return a range of characters by using the slice syntax.

Specify the start index and the end index, separated by a colon, to return a part of the string.

Ex:-

b = "Hello, World!"  
print(b[2:5])

Slice from the start:

b = "Hello, World!"  
print(b[:5])

Slice to the end:

b = "Hello, World!"  
print(b[2:])