VARIABLE AND DATATYPES

## VARIABLE:

Variables are containers for storing data values.

It is a “dynamically typed" in the sense that a variable may contain different types of values over its lifetime.

So there is no need to declare the datatype of the variable particularly in python.

Creating Variables:

A variable is created the moment you first assign a value to it. We don’t need any semicolon in python, as it very simple programming language. Just if u say to print the value using “print” it will execute the variable.

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

output is:

5

John

## Casting:

## It is used to specify the datatype of the variable if need.

## x = str(3)    # x will be '3' y = int(3)    # y will be 3 z = float(3)  # z will be 3.0

## output is :

## 3

## 3

## 3.0

## Case-sensitive:

## Python is a case- sensitive which mean lower case of the alphabet and upper case of the alphabet are distinct and different.

## Example a=”hello” and A=”hi” in this case the hello is not overwritten while if a=10 and a=9 here it will overwrite.

## Variable Names:

## It can be short name of a long name such as x,y,age,hello and so on.

## It also has rules to follow while creating a variable in the python:

## Can start with letter/uppercase.

## Cannot start with numbers. And its case sensitive too.

## We cannot keep a keyword(such as if,else,while….)-as its reserved.

## Variables can only have alphabetic-numerical along with underscores in it.

## Example:

## break=”hi”, 4\_hello=”john”, h\_3$el=”hi” all of its are not accepted.

## While my=”john”, \_my=”john”, MY9=”john” are accepted[legal].

## Multi Words Variable Names:

## Variable with more than multiple words can also be used but its bit difficult for the system to read.

## In that case we use few techniques:

## Camel case-each word except the first word is capital letter.

## myVariableName = "John"

## Pascal case-each word starts with capital letter.

## MyVariableName = "John"

## Snake case-each words are separated with underscore.

my\_variable\_name = "John"

## Many Values to Multiple Variables:

## The number of the variable should match with the number of values else it will throw error.

## If you give x=y=z= “orange”. And if you try to print the x,y and z value then all will be orange as we assigned only 1 value to multiple variable.

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

## output is:

## Orange

## Banana

## Cherry

## Unpack a Collection:

## If you have a list of items in list, tuples and etc.

## You can extract the values from it and give it to variable’s- defined as unpacking.

## fruits = ["apple", "banana", "cherry"] x, y, z = fruits print(x) print(y) print(z)

## output is:

## apple

## banana

## cherry

## Output Variables:

## The Python print() function is often used to output variables.

## You can perform concatenation of the values to such as:

## A=”hello”

## B=”world”

## Print(A+B)

## #the output will be helloworld.

## But if B=3, then it will throw error #as A and B both are of different datatype.

## In that case you can do:

## A=5

## B=”hello”

## Print(A,B) #output is 5 hello.

## We can even perform addition of two number using the same concept.

# Global Variables:

Variables that are created outside of a function.

Can be used by everyone, both inside of functions and outside.

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

Output is:

Python is fantastic  
Python is awesome

## Here the x=”awesome” is a global variable. While the x=”fantastic” is a local variable.

## Local variable has the same variable name of the global variable but its scope is only within a function.

## To create a global variable within a function. We use the keyword “global”.

## def myfunc():   global x   x = "fantastic" myfunc() print("Python is " + x) # output: Python is fantastic

## Here we are saying that x is a global variable so it can have a global as well as local values too.

## DATATYPES:

# Python Data Types:

Python has lot of built in datatypes- means predefined datatypes.

Variable can store values of different datatypes.

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:

## Can get the data type of any object by using the type() function.

## x = 5 print(type(x)) #<class ‘int’>

## Setting the Specific Data Type:

## Example:

## x = frozenset(("apple", "banana", "cherry"))

## x = bytearray(5)

## x = bytes(5)

## x = memoryview(bytes(5))

## x = range(6)

## x = set(("apple", "banana", "cherry"))

## x = complex(1j)

## x = int(20)

## x = list(("apple", "banana", "cherry"))

## x = tuple(("apple", "banana", "cherry"))