

Variable

- **Variable** is a name that is used to refer to memory location. Python variable is also known as an identifier and used to hold value.
- In Python, we don't need to specify the type of variable because Python is a infer language and smart enough to get variable type.
- Variable names can be a group of both the letters and digits, but they have to begin with a letter or an underscore.

Example:

Here we have stored “**Topper World**” in a **var** which is variable, and when we call its name the stored information will get printed.

```
Var = "Topper World"
print(Var)
```

Output:

```
Topper World
```

Note:

- The value stored in a variable can be changed during program execution.
- A Variables in Python is only a name given to a memory location, all the operations done on the variable effects that memory location.

❖ Rules for Python variables

- A Python variable name must start with a letter or the underscore character.
- A Python variable name cannot start with a number.
- A Python variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _).
- Variable in Python names are case-sensitive (name, Name, and NAME are three different variables).
- The reserved words(keywords) in Python cannot be used to name the variable in Python.

❖ Declaring Variable and Assigning Values

- Python does not bind us to declare a variable before using it in the application. It allows us to create a variable at the required time.
- We don't need to declare explicitly variable in Python. When we assign any value to the variable, that variable is declared automatically.
- The equal (=) operator is used to assign value to a variable.

Example:

```
# declaring the var
```

```
Number = 100
```

```
# display
```

```
print( Number)
```

Output:

```
100
```

❖ Types of Variable

There are two types of variables in Python - Local variable and Global variable. Let's understand the following variables.

◆ Local Variable

Local variables are the variables that declared inside the function and have scope within the function.

Example:

```
# This function uses global variables  
def f():  
    s = "Topper World"  
    print(s)
```

Output:

```
Topper World
```

◆ Global Variables

Global variables can be used throughout the program, and its scope is in the entire program. We can use global variables inside or outside the function.

Example:

```
# This function has a variable with name same as s.  
def f():  
    print(s)  
  
# Global scope  
s = "I love Topper World"  
  
f()
```

Output:

```
I love Topper World
```

❖ Delete a variable

We can delete the variable using the **del** keyword. The syntax is given below.

Syntax -

```
del <variable_name>
```

❖ How does + operator work with variables?

- The Python plus operator + provides a convenient way to add a value if it is a number and concatenate if it is a string.
- If a variable is already created it assigns the new value back to the same variable.

Example:

```
a = 10
b = 20
print(a+b)

a = "Topper"
b = "World"
print(a+b)
```

Output:

```
30
TopperWorld
```