

PYTHON

Variables: A variable is a name for the memory location which stores a particular value of some sort of data . Python has pre defined methods to get the memory location of the variable and Data type of the Data stored in the variable .

Standards of naming a variable:

- The variable name should start with a lower case character unless it is declared as a constant.
 - a , name , list , number ... <--- valid & recommended
 - Name , Number , List <--- valid , but not recommended
 - PI , EPSILON <--- valid(constants)
- Variable name cannot start with a numerical value or special character.
 - @name , \$number ,)list <---- invalid
- One can use camel case or underscore(_) separated variable names for assigning values.
 - firstName , numbersList , students_fee

A variable is assigned a value by an assignment operator(=)

Eg : a = 10 ,name = "Python" , fee = 100000

Methods on Variables:

type(variable_name) ---> provides with the data types of data stored in the variable_name

id(variable_name) ---> provides with the unique number that represents the memory location of the variable_name

Multiple Assignment:

Assigning a value to multiple variables.

Eg: a=b=c=24

Assigning multiple values to multiple variables.

Eg: p,q,r=10,20,30

The value will be assigned in the order in which the variable appears.

Data Types in Python :

Python has 5 Data types of which two are independent and the remaining are derived or dependent data types .

The data types in Python:

- Numbers
- Strings
- Lists
- Tuples
- Dictionaries

The first two Data Types are independent and the later are Dependent / Derived Data types.

Note : In Python Boolean (True , False) are not considered as Data Types but they still exist as status flags.

Numbers: Number data type stores numeric values only.

Python supports four numeric type:

Int: This takes only the integer values.

Eg: 10,-43

Long: This can take integer values size greater than 10 and octal and hexadecimal values.

Eg:56356754L , DEFGHF1453F

Float: This take only value with point (.) .

Eg: 123.43,-12.64

Complex: This will take complex number if form A+iB.

Where A=Real value and B=imaginary value

Eg: 3+4j, 6+3j, 4j

Number Type Conversion: A number in python at some time is to be converted to another to satisfy the requirement according to the operators. To perform conversion python provide in-built functions for converting.

int(a): It convert the value of a into a plain integer.

long(a): It convert the value of a into a long integer.

Float(a): It convert the value of a into a floating –point number .

Complex(x): It convert the value of a into a complex number with a taken as the real part and imaginary part as 0.

Complex(a, b): It convert the value of a and b into complex number where a will be real part and b will be the imaginary part of complex number.

Python internally has in-built mathematical functions that can be performed on integer values.

Mathematical Functions: The functions used for integers are following:

ceil(a): It will return the smallest integer greater or equal to a.

fabs(a): It will return absolute value of a.

factorial(a): It will return factorial value of a value.

Floor(a): It will return largest integer less than or equal to x.

Fmod(a,b): It will return the remainder when a is divided by b.

Sqrt(a): It will return the square root of a.

exp(a): It will return exponential value of a

power(a,b): It will return value of a with the power of b. $a^{**}b$.

pi: It will return the pi value which is $\pi=3.14$

These are the functions which are frequently used. There are still different functions like trigonometric, random numbers that are used in python which will be discussed in future.

Operators: Operators are the special symbols used to perform computations actions, conditional matching etc.,

Eg: 3/5 where 3,5 are operands and / is a operators.

Type of Operators:

Python supports different types of operators:

- Arithmetic Operators
- Comparison Operators
- Logical Operators
- Assignment Operators
- Bitwise Operators
- Conditional Operators
- Membership Operators
- Identity Operators

Arithmetic Operators: These are the operators which are used to perform basic Mathematic calculations.

Addition(+): Operator used for addition of two or more values.

Eg: 4+5

Subtraction(-): Operator used for subtraction of two or more number.

Eg: 5-4

Multiplication(*):Operator used for product of two numbers.

Eg: 8*6

Division(/): Operator used for division of number with other.

Eg: 4/2

Modulus(%): Same as Division but give remainder as output.

Eg: 5/2=1

Exponent(**): Operators for power of number.

Eg: $5^{**}3=125$

Floor division($//$): Same as division but don't consider value after point.

Eg: $24/5=4$

Comparison Operators: This perform comparison between the values and return output in the form of Boolean (True/False). These are also called as Relational Operators.

($==$): If both values are same then it will return True other wise False.

Eg: $a=5$ $b=2$ ($a==b$)=False

($!=$): If both values are different it return True and viceversa.

Eg: $a=5$ $b=2$ ($a==b$)=True

($>$): Check the number is greater or not and return True is Yes and Viceversa.

Eg: $a>b$ =True

($<$): Check the number is smaller or not and return True if Yes and viceversa.

Eg: $a<b$ =False

($>=$): Checks greaterthan or equal to condition with numbers and return True or false.

Eg: $a>=b$ =True

($<=$): Checks smallerthan or equal to condition with numbers and return True or false.

Eg: $a<=b$ =False

Assignment Operator: Operator that assign a value to a variable is called assignment operators.

If $a=1$

($=$): Assign value from right side to the Left side variable.

Eg: $c=9+4$

($+=$): Adds right side operand to left and assign result to left operand.

Eg: $a+6$ $a=1+6$ $a=7$

($-=$): Subtract right side operand to left and assign result to left operand.

Eg: $a -= 6$ $a = 7 - 6$ $a = 1$

(\ast =): Multiply right side operand with left and assign result to left operand.

Eg: $a \ast = 7$ $a = 1 \ast 7$ $a = 7$

($/$ =): Divide right side operand with left and assign result to left operand.

Eg: $a /= 7$ $a = 7 / 7$ $a = 1$

($\%$ =): Divide right side operand with left operand and assign remainder as result to left operand.

Eg: $a \% = 5$ $a = 1 \% 5$ $a = 4$

($\ast \ast$ =): perform power of right side operand with left operand and assign result to left operand.

Eg: $a \ast \ast = 3$ $a = 4 \ast \ast 3$ $a = 64$

($//$ =): Perform floor division and assign value to the left operand.

Eg: $a //= 9$ $a = 64 // 9$ $a = 7$

Bitwise Operators: Operators which convert the values into binary format and perform Bitwise operations and give result in binary format.

Binary AND(&): Convert values to binary and perform AND which is if both the both the binary values are 1 the output will be 1 otherwise 0 in all cases.

Eg: $x = 34, y = 23$ $x = 0010 \ 0010$

$y = 0001 \ 0111$

$x \& y = 0000 \ 0010 = 2$

Binary OR(|): Convert values to binary and perform OR which is if any the binary values are 1 the output will be 1 otherwise 0 in all cases.

Eg: $x = 34, y = 23$ $x = 0010 \ 0010$

$y = 0001 \ 0111$

$x | y = 0011 \ 0111 = 55$

Binary XOR(^): Convert values to binary and perform XOR which is if both the values of binary format are same then result is 0 and 1 in othercase.

Eg: $x=34, y=23$ $x = 0010\ 0010$

$y = 0001\ 0111$

$x \wedge y = 0011\ 0101 = 53$

Binary Ones complement(\sim): Convert values to binary and perform Ones complement which is converting 1's to 0's and 0's into 1's.

Eg: $x=34, y=23$ $x = 0010\ 0010$

$\sim x = 1101\ 1101 = -34$

Binary Leftshift(\ll): Left side operand value will be moved towards left taking the condition in right operand.

Eg: $x=34$ $x = 0010\ 0010$

$x \ll 2 = 1000\ 1000 = 136$

Binary RightShift(\gg): Left side operand value will be moved towards right taking the condition in right operand.

Eg: $x=34$ $x = 0010\ 0010$

$x \gg 1 = 0100\ 0100 = 68$

Logical Operators: Operators that are used to perform logical operation such as AND , OR and NOT are called logical operators and give output as Boolean.

Logical AND(and): This will give true only if both the conditions are true otherwise false.

Eg: $a=8<6$ and $6>5$ = False

(F) (T)

Logical OR(or): This will give True if one condition is true otherwise false.

Eg: $a=8<6$ or $6>5$ = True

(F) (T)

Logical NOT(not): This will convert True into False and viceversa.

Eg: $a=\text{not}(5<6)$ = False

Membership Operators: Membership operators are used to test whether the condition is True or False in a Sequence.

In: This operator evaluates True only if the specified variable is found in specified sequence and give false otherwise.

Eg: a="python" list=["datascience","game","python"]

if (a in list):

print ("python")

else: = Python

print ("datascience")

notin: This operator evaluates True only if the specified variable is found in specified sequence and give false otherwise.

Eg: a="python" list=["datascience","game","python"]

if (a notin list):

print ("python")

else: ="datascience"

print ("datascience")

Identity Operators: These operators are used to compare the memory location of the objects in the system.

is: This operator evaluates True only if the variable on both side of the operator point to the same object and false otherwise.

Eg: a=15 b=15

If (a is b):

Print ("a,b are same")

Else: a,b are same

Print ("a,b different")

Is not: This operator evaluates True only if the variable on both side of the operator not point to the same object and false otherwise.

Eg: a=15 b=15

If (a is not b):

Print ("a,b are same")

Else: a,b are different

Print ("a,b different")