

Introduction

Python has very simple syntax. You don't need to import any libraries to get started.

To print any statement, you need to use the `print("Statement")` command. Semi-colons at the end of statement are not required in python.

```
In [1]: print("Hello World")
```

Hello World

Variables and Type

In python, you don't need to explicitly define the variable type. You just have to assign the value and python implicitly defines the type. You can use the `type(Variable_Name)` to know the type of variable.

A variable can hold integer, float, string, boolean etc values.

Variable name rules

Must be alpha-numeric i.e. can only contain alphabets and numbers. The only special character allowed is `_`(underscore)

Must always start with an alphabet

It is case sensitive i.e. "glass" and "Glass" are considered two separate variables

Operators, Comments

Operators: `+`(addition), `-`(subtraction), `*`(multiplication), `/`(division), `%`(modulus), `**`(exponent), `//`(integer division)

Single line comments: `#Statements`

```
In [2]: integer_variable = 10
float_variable = 12.3
string_variable = "Harry Potter"
boolean_variable = False
print("Type of",integer_variable,"is ",type(integer_variable))
print("Type of",float_variable,"is ",type(float_variable))
print("Type of",string_variable,"is ",type(string_variable))
print("Type of",boolean_variable,"is ",type(boolean_variable))

#This is an example of a single line comment
print("\nExample of Operators: ")
x = 10
y = 3

print("x + y= ",x+y)
print("x - y= ",x-y)
print("x * y= ",x*y)
```

```
print("x / y= ",x/y)
print("x % y= ",x%y)
print("x ** y= ",x**y)
print("x // y= ",x//y)
```

```
Type of 10 is <class 'int'>
Type of 12.3 is <class 'float'>
Type of Harry Potter is <class 'str'>
Type of False is <class 'bool'>
```

Example of Operators:

```
x + y= 13
x - y= 7
x * y= 30
x / y= 3.3333333333333335
x % y= 1
x ** y= 1000
x // y= 3
```

Conditional operators

== Equal

!= Not Equal

>= Greater than or Equal

<= Less than or Equal

> Greater than

< Less than

These operators are used to compare two quantities and returns the value True or False.

Using 'and' & 'or'

When you want to check two or more conditional statements, 'and' and 'or' keywords are to be used.

'and' is equivalent to &&

'or' is equivalent to ||

Rules are as below:

Given two conditions: A, B

A and B returns True when both the statements are true

A or B returns True when either A or B is true

```
In [3]: print("Checking equality of two variables:")
x = 2
y = 3
z = 2
print("x == y:",x==y)
print("x != y:",x!=y)
print("x <= y:",x<=y)
print("x >= y:",x>=y)
print("x < y:",x<y)
print("x > y:",x>y)
```

```
print("\nUing and & or")
A = x == y
B = x == z

print("A and B:", A and B)
print("A or B:", A or B)
```

Checking equality of two variables:

```
x == y: False
x != y: True
x <= y: True
x >= y: False
x < y: True
x > y: False
```

Uing and & or

A and B: False

A or B: True

Rules of Precedence

Following are the rules of precedence in the decreasing order:

() Parantheses

** Exponent

*, /, %, // Multiplication, Division, Modulo, Integer Division

+, - Addition, Subtraction

Operators with the same level of precedence eg + and - are evaluated as per the rules of associativity

All the operators except the exponent(**) have associativity from left to right.