Python Operators

Python4All

- https://github.com/bachinaram/python4all

Python operator groups

Arithmetic operators

Assignment operators

Comparison operators

Logical operators

Identity operators

Ternary operators

Membership operators

Bitwise operators – We are not learning this (&, | ,^,~,<<,>>)

Arithmetic operator

Name	Operator	e.g.,
Addition	+	10+20=30 "hello"+"world"='helloworld' 10.25+0.123=10.373 True+True=2, by default True is 1, False is 0 [1,2,3,4]+["hello",0.12,5]=[1,2,3,4,"hello",0.12,5]
Subtraction	-	10-20=-10 10.25+0.123=10.127 True-True=0, by default True is 1 False is 0
multiplication	*	10*20=200 10.25+0.123=1.26075 True*True=1 "Hello"*2=HelloHello
Division	/	20/10=2 10.2/4.56=2.23684210526 True/5=0.2
Floor Division	//	7//2=3 (round to floored whole number) 4.5/0.2=22.5 True//5=0
Modulus	%	6%4=2 (It is a reminder)
Exponentiation	**	10**4=10000

Assignment Operators

Operator	Example	Equivalent Syntax
=	Var1=10	Var1=10
+=	Var1+=10	Var1=var1+10
-=	Var1-=10	Var1=var1-10
*=	Var1*=10	Var1=var1*10
/=	Var1/=10	Var1=var1/10
//=	Var1//=10	Var1=var1//10
%=	Var1%=10	Var1=var1%10
**=	Var1**=10	Var1=var1**10
..		
..		

Comparision Operator

Name	Operator	Usage
Equal	==	10==20 False; "hello"=="hello True; [1,2,3]==[4,5] False
Not Equal	!=	10!=20 True;"hello!="hello" False;[1,2,3]!=[4,5] True
Less than	<	20<30 True; 20.25<13.25 False;'H'<'h' True
Greater than	>	20>30 False;2.25>13.25 True;'Hello'>'hello' False
Less than equal to	<=	20<=30 True;10.25<=11.25 True;
Greater than equal to	>=	20<=30 False;10.25>=10.25 True;

Logical operators

Name	Operator	Example
Logical AND	and	True and True – True True and False – False False and False – False 10 and 20 – 20 20 and 5 – 5 10>5 and 20<40 - True
Logical OR	or	True or True – True True or False – True False or False – False 10 or 20 – 10 20 or 5 – 20 10>5 and 20<40 - True
Logical NOT	not	not(True or True) – False not(True and True) – False

Identity operators

- is
  - Compares if both operands refer same memory address
  - By default, number from –5 to 256 always refers to fix memory
  - Check using id() - object identity address. E.g., id(10) returns object address
- is not
  - Compares if not both operands refers same memory address
- is vs ==
  - Both are not same
  - "is checks for memory address for both operands"
  - "== check for data in memory address is same or not"
- is not vs !=
  - Both are not same

Ternary operator

- Python doesn’t have ternary operators like(?:)
- Instead, we can create the situation using below if else
  - Syntax – a If condition else b
  - First condition evaluates if condition is True then a returned else then b is returned
  - e.g1, 10 if True else 20 – 10
  - e.g2., x=10 if False else 20 – returns x=20

Membership Operators

- "in - True if value/variable is found in the sequence(tuple,list,dictionary,string,set)
  - 2 in [1,2,3] - True
  - 'c' in "hello" - false
  - 'name' in {'name':'ram', 'age':25}; checks in keys - True
- "not in - True if value/variable is found in the sequence
  - 2 not in [1,2,3] - False
  - 'c' not in "hello" - True
  - 'name' not in {'name':'ram', 'age':25}; checks in keys - False