Python operators

Author : Srishti Sawla
LinkedIn URL : www.linkedin.com/in/srishtisawla

1. Arithmetic operators

Operator	Name	Description	Syntax	Example
+	Addition	Performs addition	c = a **+** b	a = 5, b = 5 then c = 10
-	Subtraction	Performs subtraction	c = a **–** b	a = 5, b = 3 then c = 2
*	Multiplication	Performs multiplication	c = a ***** b	a = 5, b = 5 then c = 25
/	Division	Performs division	c = a **/** b	a = 10, b = 5 then c = 2
%	Modulus	Performs division but returns the remainder	c = a **%** b	a = 15, b = 2 then c = 1
//	Floor Division	Performs division but returns the quotient in which the digits after the decimal points are removed	c = a **//** b	a = 15, b = 2 then c = 7
**	Exponent	Performs multiplication to power raised	c = a ****** b	a = 2, b = 4 then c = 16

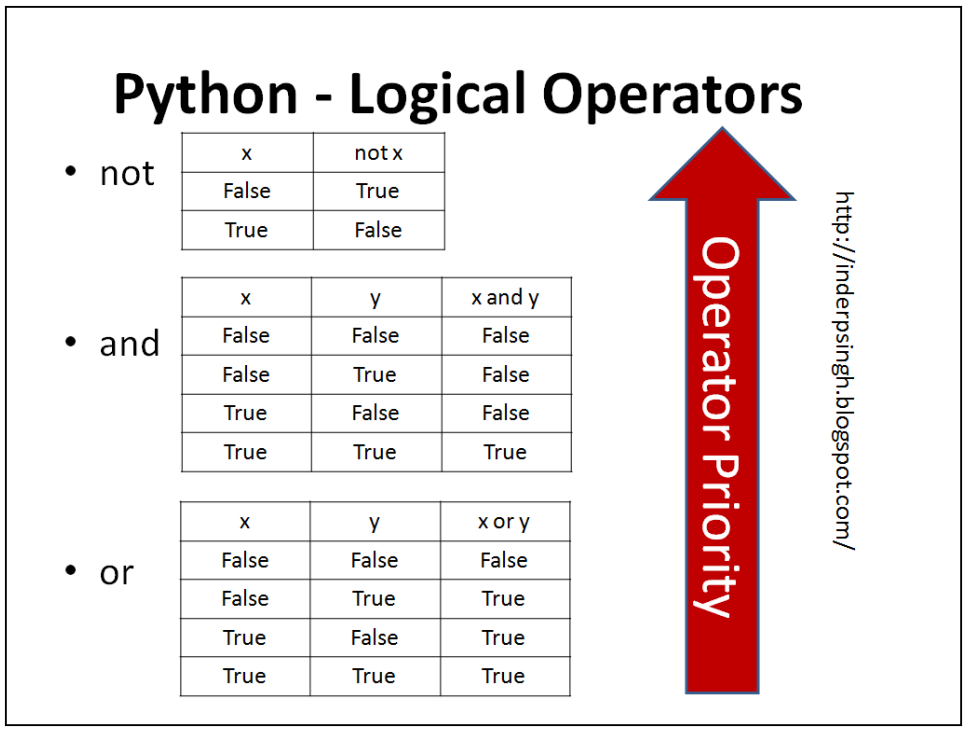
2. Assignment Operators

Operator	Name	Description	Syntax	Example
=	Equals	Assigns the value or operand from the right to the variable on the left	c = a **+** b	a = 15, b = 5 then c = 20
+=	Add AND	Adds the **right operand** to the **left operand** and the result is assigned to the **left operand**.	c += 5	a = 5, b = 15 then True
==	Equal to	Compares the operands and then returns **True** if both the operands are equal or else **False**	a **==** b	a = 5, b = 5 then True
!=	Not equal to	Compares the operands and then returns **True** if both the operands are **not equal** or else **False**	a **!=** b	a = 10, b = 5 then True
>=	Greater than or equal to	Compares the operands and then returns **True** if the left operand is greater than or equal to the right or else **False**	a **>=** b	a = 15, b = 2 then True
<=	Lesser than or equal to	Compares the operands and then returns **True** if the left operand is lesser than or equal to the right or else **False**	a **<=** b	a = 2, b = 15 then True

3. Comparison operators

Operator	Name	Description	Syntax	Example
>	Greater than	Compares the operands and then returns **True** if the left operand is greater than the right or else **False**.	a > b	a = 15, b = 5 then True
<	Lesser than	Compares the operands and then returns **True** if the left operand is lesser than the right or else **False**	a < b	a = 5, b = 15 then True
==	Equal to	Compares the operands and then returns **True** if both the operands are equal or else **False**	a == b	a = 5, b = 5 then True
!=	Not equal to	Compares the operands and then returns **True** if both the operands are **not equal** or else **False**	a != b	a = 10, b = 5 then True
>=	Greater than or equal to	Compares the operands and then returns **True** if the left operand is greater than or equal to the right or else **False**	a >= b	a = 15, b = 2 then True
<=	Lesser than or equal to	Compares the operands and then returns **True** if the left operand is lesser than or equal to the right or else **False**	a <= b	a = 2, b = 15 then True

Logical Operators



4. Identity operators

Operator	Name	Example
is	TRUE, if both the variable points to the same object, with same memory locations.	x is y
is not	TRUE, if both the variable points to different objects.	x is not y

5. Membership operators

Operator	Name	Example
in	TRUE, if variable is in the list, string, dictionary, etc.	x in y
not in	TRU, if variable is not in the list, string, dictionary, etc.	x not in y

7.Bitwise operators

Operator Name	Operations	Result
Bitwise OR (|)	x | y	bitwise or of x and y
Bitwise XOR (^)	x ^ y	bitwise exclusive or of x and y
Bitwise AND (&)	x & y	bitwise and of x and y
Bitwise Left-shift (<<)	x << n	x shifted left by n bits
Bitwise Right-shift (>>)	x >> n	x shifted right by n bits
Bitwise ones Complement (~)	~x	the bits of x inverted

Python built In functions

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

Coding problems solved

1. Write a python program to check if an element exists in the list.
2. Write a python program to swap two numbers without using temp variable.
3. Write a python program to check if a substring is present in the string.
4. Write a python program to print nth letter of a word.
5. Write a python program to check given number is even or odd.
6. Write a python program to check if a given number is palindrome or not.