Module 2 : CSS 103 Relational Operators, Logical Operators, Precedence of Operators, Bitwise Operators

Notes

Relational Operators, Logical operators, Precedence of Operators, Bitwise Operators

Operators are the constructs which can manipulate the value of operands.

Consider the expression 4 + 5 = 9. Here, 4 and 5 are called operands and + is called operator.

Types of Operator

Python language supports the following types of operators.

- Arithmetic Operators
- Comparison (Relational) Operators
- Logical Operators
- Bitwise Operators

Python Arithmetic Operators

Operator 0		Description
Addition	+	Adds values on either side of the operator.
Subtraction	-	Subtracts right hand operand from left hand operand.
Multiplication	*	Multiplies values on either side of the operator
Division	/	Divides left hand operand by right hand operand
Modulus	%	Divides left hand operand by right hand operand and
		returns remainder
Exponent	**	Performs exponential (power) calculation on operators

Python Relational Operators

These operators compare the values on either sides of them and decide the relation among them. They are also called Relational operators.

Operator Description

- == If the values of two operands are equal, then the condition becomes true.
- != If values of two operands are not equal, then condition becomes true.
- <>> If values of two operands are not equal, then condition becomes true.
- > If the value of left operand is greater than the value of right operand, then condition becomes true.
- < If the value of left operand is less than the value of right operand, then condition becomes true.
- >= If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.
- <= If the value of left operand is less than or equal to the value of right operand, then condition becomes true.

Python Logical Operators

Operator	Description
and Logical AND	If both the operands are true then condition becomes true.
or Logical OR	If any of the two operands are non-zero then condition
	becomes true.
not Logical NOT	Used to reverse the logical state of its operand.

Python Bitwise Operators

Bitwise operator works on bit	ts and performs bit by bit operation.			
Operator	Description			
& Binary AND	Operator copies a bit to the result if it exists in both			
operands				
Binary OR	It copies a bit if it exists in either operand.			
^ Binary XOR	It copies the bit if it is set in one operand but not			
	both.			
~ Binary 1s Complement	It is unary and has the effect of 'flipping' bits.			
<< Binary Left Shift	The left operands value is moved left by the number of			
bits specified by the right operand.				
>> Binary Right Shift	The left operands value is moved right by the number			
of bits specified by the right operand.				

Python Precedence of Operators The operator precedence in Python in descending order.

<u>Operators</u>	<u>Meaning</u>
0	Parentheses
**	Exponent
+x, -x, ~x	Unary plus, Unary minus, Bitwise NOT
*, /, //, %	Multiplication, Division, Floor division, Modulus
+, -	Addition, Subtraction
<<,>>>	Bitwise shift operators
&	Bitwise AND
۸	Bitwise XOR
	Bitwise OR
==, !=, >, >=, <, <=, is, is not, in, not in	Comparisons, Identity, Membership operators
not	Logical NOT
and	Logical AND
or	Logical OR