## Python Operators

Operators are used to perform operations on variables and values.

Python divides the operators in the following groups:

* Arithmetic operators
* Assignment operators
* Comparison operators
* Logical operators
* Identity operators
* Membership operators
* Bitwise operators

**Python Arithmetic Operators**

Arithmetic operators are used with numeric values to perform common mathematical operations:

|  |  |  |  |
| --- | --- | --- | --- |
| **Addition** | **Subtraction** | **Multiplication** | **Exponentiation** |
| **+** | **-** | **\*** | **\*\*** |
| x = 5  y = 3  print(x + y) | x = 5  y = 3  print(x - y) | x = 5  y = 3  print(x \* y) | x = 2  y = 5  print(x \*\* y)  #2\*2\*2\*2\*2 |
| **Division** | **Floor division** | | **Modulus** |
| **/** | **//** | | **%** |
| x = 12  y = 3  print(x / y) | x = 15  y = 2  print(x // y)  #division with round number | | x = 5  y = 2  print(x % y) |

## Python Assignment Operators

Assignment operators are used to assign values to variables:

|  |  |  |  |
| --- | --- | --- | --- |
| **Assignment Operator** **=** | **Addition A/O** **+=** | **Subtraction A/O** **-=** | **Multiplication A/O** **\*=** |
|  | **x = x + 3** | **x = x - 3** | **x = x \* 3** |
| x = 5  print(x) | x = 5  x += 3  print(x) | x = 5  x -= 3  print(x) | x = 5  x \*= 3  print(x) |
| **Division A/O** **/=** | **Modulus A/O** **%=** | **Floor Division A/O** **//=** | **Exponentiation****A/O** **\*\*=** |
| |  |  | | --- | --- | |  | **x = x / 3** | | **x = x % 3** | **x = x // 3** | |  |  | | --- | --- | |  | **x = x \*\* 3** | |
| x = 5  x /= 3  print(x) | x = 5  x%=3  print(x) | x = 5  x//=3  print(x) | x = 5  x \*\*= 3  print(x) |
| **Bitwise AND AO** **&=** | **Bitwise OR A/O** **|=** | **Bitwise XOR**  **^=** |  |
| **x = x & 3** | **x = x | 3** | **x = x ^ 3** |  |
| x = 5  x &= 3  print(x) | x = 5  x |= 3  print(x) | x = 5  x ^= 3  print(x) |  |
| 1 0 1 – 5  0 1 1 – 3  0 0 1 - 1 | 1 0 1 – 5  0 1 1 – 3  1 1 1 - 7 | 1 0 1 – 5  0 1 1 – 3  1 1 0 - 6 |  |
| **Bitwise Right Shift Assignment Operator** **>>=** | **Bitwise Left Shift Assignment Operator** **<<=** |  |  |
| x= x >> 3 | x = x << 3 |  |  |
| x = 5  x >>= 3  print(x) | x = 5  x <<= 3  print(x) |  |  |
| **4 2 1**  1 0 1  0 0 0 1 0 1 | **32 16 8 4 2 1**  1 0 1  1 0 1 0 0 0 |  |  |

**Python Comparison Operators**

Comparison operators are used to compare two values.

|  |  |  |
| --- | --- | --- |
| **Equal Operator**  **==** | **Not equal Operator**  **!=** | **Greater than Operator**  **>** |
| x = 5  y = 3  print(x == y) | x = 5  y = 3  print(x != y) | x = 5  y = 3  print(x > y) |
| returns False because 5 is not equal to 3 | returns True because 5 is not equal to 3 | returns True because 5 is greater than 3 |
| **Less than Operator**  **<** | **Greater than or equal to Operator**  **>=** | **Less than or equal to**  **Operator**  **<=** |
| x = 5  y = 3  print(x < y) | x = 5  y = 3  print(x >= y) | x = 5  y = 3  print(x <= y) |
| returns False because 5 is not less than 3 | returns True because five is greater, or equal, to 3 | returns False because 5 is neither less than or equal to 3 |

## Python Logical Operators

Logical operators are used to combine conditional statements:

|  |  |  |
| --- | --- | --- |
| **and** | **or** | **not** |
| x = 5  print(x > 3 and x < 10) | x = 5  print(x > 3 or x < 4) | x = 5  print(not(x > 3 and x < 10)) |
| returns True because 5 is greater than 3 AND 5 is less than 10 | returns True because one of the conditions are true (5 is greater than 3, but 5 is not less than 4) | returns False because not is used to reverse the result |

## Python Identity Operators

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location:

|  |  |
| --- | --- |
| **is** | **is not** |
| x = ["apple", "banana"]  y = ["apple", "banana"]  z = x  print(x is z) # return true  print(x is y) # return false  print(x == y) #return true | x = ["apple", "banana"]  y = ["apple", "banana"]  z = x  print(x is not z) #false  print(x is not y) #true  print(x != y) #false |
| Returns True if both variables are the same object | |  |  | | --- | --- | |  | Returns True if both variables are not the same object | |

## Python Membership Operators

Membership operators are used to test if a sequence is presented in an object:

|  |  |
| --- | --- |
| **in** | **not in** |
| x = ["apple", "banana"]  print("banana" in x) #return true | x = ["apple", "banana"]  print("pineapple" not in x) #return true |
| |  |  | | --- | --- | |  | Returns True if a sequence with the specified value is present in the object | | |  |  | | --- | --- | |  | Returns True if a sequence with the specified value is not present in the object | |