Python Arithmetic Operators

Addition	`9 + 2`	`>> 11`
Subtraction	`9 - 2`	`>> 7`
Multiplication	`9 * 2`	`>> 18`
Division	`9 / 2`	`>> 4.5`
Modulus	`9 % 2`	`>> 1`
Exponentiation	`3 ** 2`	`>> 81`
Floor division	`9 // 2`	`>> 4`

Python Assignment Operators

Operator	Example	Same As
=	`x = 2`	`x = 2`
+=	`x += 2`	`x = x + 2`
-=	`x -= 2`	`x = x - 2`
*=	`x *= 2`	`x = x * 2`
/=	`x /= 2`	`x = x / 2`
%=	`x %= 2`	`x = x % 2`
//=	`x //= 2`	`x = x // 2`
**=	`x **= 2`	`x = x ** 2`

Python Comparison Operators

Equal	`x == y`
Not equal	`x != y`
Greater than	`x > y`
Less than	`x < y`
Greater than or equal to	`x >= y`
Less than or equal to	`x <= y`

Boolean Values

In programming you often need to know if an expression is `True` or `False`.
You can evaluate any expression in Python, and get the answer.

```
print(5 < 8)        >>> True
print(5 > 8)        >>> False
```

Python Logical Operators

and	Returns True if both statements are true

```
x < 5 and x < 10
```

or	Returns True if one of the statements is true

```
x < 5 or x < 4
```

not	Reverse the result, returns False if the result is true

```
not(x < 5 and x < 10)
```

Python Identity Operators

is	Returns true if both variables are the same object

```
x is y
```

is not	Returns true if both variables are not the same object

```
x is not y
```

Python Membership Operators

in	Returns True if a sequence with the specified value is present in the object

```
x in y
```

not in	Returns True if a sequence with the specified value is not present in the object

```
x not in y
```

Python Bitwise Operators

&	AND	Sets each bit to 1 if both bits are 1
|	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off