

Arithmetic Operators

+ Addition	<code>print(3 + 5) # 8</code>
- Subtraction	<code>print(8 - 6) # 2</code>
* Multiplication	<code>print(1 + 2 + 3 * 3) # 12</code>
/ Division	<code>print(8 / 4) # 2</code>
% Mod (the remainder after dividing)	<code>print(9 % 2) # 1</code>
** Exponentiation	<code>print(3 ** 2) # 9</code>
// Divides and rounds down to the nearest integer	<code>print(16 // 3) # 5</code>

Logical Operators

and Evaluates if all provided statements are True	<code>5 < 3 and 5 == 5 # False</code>
or Evaluates if at least one of many statements is True	<code>5 < 3 or 5 == 5 # True</code>
not Flips the Bool Value	<code>not 5 < 3 # True</code>

Data Structures

```
months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']

print(mon ths[0]) # January
print(mon ths[1]) # February
print(mon ths[7]) # August
print(mon ths [-1]) # December
print(mon ths [25]) # IndexError: list index out of range
```

Assignment Operators

<code>x = 3 y= 4 z=5</code>	<code>x, y, z = 3, 4, 5</code>
+= Addition Assignment	<code>a += 1 # a = a + 1</code>
-= Subtraction Assignment	<code>a -= 3 # a = a - 3</code>
*= Multiplication Assignment	<code>a = 4 # a = a4</code>
/= Division Assignment	<code>a /= 3 # a = a / 3</code>
%= Remainder Assignment	<code>a %= 10 # a = a % 10</code>
**= Exponent Assignment	<code>a **= 10 # a = a ** 10</code>

Strings

```
my_string = 'this is a string!'

my_string2 = "this is also a string!!!"

this_string = 'Simon's skateboard is in the garage.'

print(first_word + ' ' + second_word) >> Hello There

print(len(first_word)) >> 5

print(first_word * 5) >> HelloHelloHelloHelloHello

index into strings >> first_word[0] >> H, first_word[1] >> e
```

Comparison Operators

== Is Equal To	<code>3 == 5 # False</code>
!= Not Equal To	<code>3 != 5 # True</code>
> Greater Than	<code>3 > 5 # False</code>
< Less Than	<code>3 < 5 # True</code>
>= Greater Than or Equal To	<code>3 >= 5 # False</code>
<= Less Than or Equal To	<code>3 <= 5 # True</code>

Strings methods

len() , returns the length of an object, like a string	<code>print(len("ababa") / len("ab")) >> 2.5</code>
type() , check the data type of any variable	<code>print(type('Hello')) >> str</code>
<code>'sebastian thrun'.islower()</code>	True
<code>'Sebastian Thrun'.count('a')</code>	2
<code>'Sebastian Thrun'.find('a')</code>	3
<code>'Sebastian Thrun'.rfind('a')</code>	7
<code>animal = "dog" action = "bite" print("Does your {} {}?" .format(animal, action))</code>	Does your dog bite?
<code>"The cow jumped over the moon.".split()</code>	['The', 'cow', 'jumped', 'over', 'the', 'moon.']
<code>"The cow jumped over the moon.".split(', 3')</code>	['The', 'cow', 'jumped', 'over the moon.']
<code>"The cow jumped over the moon.".split('.')</code>	['The cow jumped over the moon', '']
<code>"The cow jumped over the moon.".split(- None, 3)</code>	['The', 'cow', 'jumped', 'over the moon.']