Strings

Strings in Python are enclosed by a pair of single quotes (‘ … ’). Strings are immutable. This means they cannot be altered when assigned or when a string variable is created. Strings can be indexed like a list as well as sliced to create new lists.

```
my_string = 'Schatz'
my_string[0]      # get first index of string
'Output': 'S'
my_string[1:4]    # slice the string from the 2nd to the 5th element
                    (indexed from 0)
'Output': 'cha'
len(my_string)    # get the length of the string
'Output': 6
my_string[-1]     # get last element of the string
'Output': 'z'
```

We can operate on string values with the boolean operators.

```
't' in my_string
'Output': True
't' not in my_string
'Output': False
't' is my_string
'Output': False
't' is not my_string
'Output': True
't' == my_string
'Output': False
't' != my_string
'Output': True
```

We can concatenate two strings to create a new string using the overloaded operator +.

```
a = 'I'
b = 'Love'
c = 'You'
```

```
a + b + c
'Output': 'ILoveYou'

# let's add some space
a + ' ' + b +  ' ' + c
```

Arithmetic and Boolean Operations

This section introduces operators for programming arithmetic and logical constructs.

Arithmetic Operations

In Python, we can operate on data using familiar algebra operations such as addition +, subtraction -, multiplication *, division /, and exponentiation **.

```
2 + 2     # addition
'Output': 4
5 - 3     # subtraction
'Output': 2
4 * 4     # multiplication
'Output': 16
10 / 2    # division
'Output': 5.0
2**4 / (5 + 3)    # use brackets to enforce precedence
'Output': 2.0
```

Boolean Operations

Boolean operations evaluate to True or False. Boolean operators include the comparison and logical operators. The comparison operators include less than or equal to <=, less than <, greater than or equal to >=, greater than >, not equal to !=, and equal to ==.

```
2 < 5
'Output': True
2 <= 5
'Output': True
```