Buitin Tuple functions

In this notebook, we will cover many of the same built-in functions that we have seen earlier.

1. `len()` : Used to find the number of elements in a sequence
2. `tuple()` : Used to create a tuple
3. `min()` : Note that the min function can be applied to both numeric as well as alpha numeric values as well as to lists.

   `However, the min() function requires that all elements be of the same type.`
4. `max()` : Note that the max function can be applied to both numeric as well as alpha numeric values as well as to lists.

   `However, the max() function requires that all elements be of the same type.`
5. `sum()` : Used to sum up the values in a tuple. Works only with numeric values.
6. membership operators. `in` and `not in`

You can find the number of elements in a tuple, using the `len()` function. You can traverse a tuple using the 'for' construct together with the `range()` function.

In [2]:

```
my_new_tuple2 = ('a', 'b', 'c', 'd')
for i in range(len(my_new_tuple2)):
    print(my_new_tuple2[i])
print()
```

```
a
b
c
d
```

Tuple traversal can also be done in the following manner and like with lists is the preferred approach.

In [3]:

```
my_new_tuple2 = ('a', 'b', 'c', 'd')
for j in my_new_tuple2:
    print(j)
```

```
a
b
c
d
```

Tuples can be created from strings by using the `tuple()` function.

In [4]:

```
my_new_tuple = tuple("Creating a list from a string")
print(my_new_tuple)
my_new_tuple2 = tuple("2345678")
print(my_new_tuple2)
```

```
('C', 'r', 'e', 'a', 't', 'i', 'n', 'g', ' ', 'a', ' ', 'l', 'i', 's', 't', ' ', 'f', 'r', 'o', 'm
', ' ', 'a', ' ', 's', 't', 'r', 'i', 'n', 'g')
('2', '3', '4', '5', '6', '7', '8')
```

To find the element in a list with the highest value, use the `max()` function. Note that the max function can be applied to both numeric as well as alpha numeric values as well as to lists. However, the `max()` function requires that all elements be of the same type.

In [5]:

```
my_new_tuple = tuple(range(0, 120, 10))
print(max(my_new_tuple))
```

```
110
```

To find the element in a list with the smallest value, use the `min()` function. Note that the min function can be applied to both numeric as well as alpha numeric values. as well as to lists. However, the `min()` function requires that all elements be of the same type.

In [6]:

```
my_new_tuple = tuple(range(0, 120, 10))
print(min(my_new_tuple))
```

```
0
```

To find the total of the elements in a list, use the `sum()` function. The sum function cannot be applied to tuples containing non-numeric values.

In [7]:

```
my_new_tuple = tuple(range(0, 120, 10))
print(sum(my_new_tuple))
```

```
660
```

The `sum()` function accepts an optional start paramter. If provided, the start value is added to the total.

In [8]:

```
my_new_tuple = tuple(range(0, 120, 10))
print(sum(my_new_tuple, 15))
```

```
675
```

The `in` and `not in` operators, referred to as membership operators are used to specify whether or not an element is in a tuple.

In [9]:

```
my_tuple = (10, 20, 30, 40, 50 )
print(10 in my_tuple)
print(54 in my_tuple)

print(10 not in my_tuple)
print(54 not in my_tuple)
```

```
True
False
False
True
```