

Tuples are very similar to lists. However they have one key difference - Immutability

once an element is inside a Tuple, it can not be reassigned.

Tuples are parathesis : (1,2,3)

```
In [1]: t = (1,2,3)

In [2]: t
Out[2]: (1, 2, 3)

In [3]: mylsit = [1,2,3]

In [4]: mylsit
Out[4]: [1, 2, 3]

In [5]: type(t)
Out[5]: tuple

In [6]: type(mylsit)
Out[6]: list

In [7]: t
Out[7]: (1, 2, 3)

In [8]: len(t)
Out[8]: 3

In [9]: t = ('one',2)

In [10]: t[0]
Out[10]: 'one'

In [11]: t[-1]
Out[11]: 2
```

Built in methods for Tuples

1. count, 2. index

```
In [12]: t = ('a','a','b')

In [16]: t.count('a')
Out[16]: 2

In [19]: t.index('a')
Out[19]: 0

In [20]: t.index('b')
Out[20]: 2

In [ ]: # Immutability difference between Lists and Tuples

In [21]: t = ('a','a','b')

In [22]: t
Out[22]: ('a', 'a', 'b')

In [23]: mylist = [1,2,3]

In [24]: mylist
Out[24]: [1, 2, 3]

In [25]: mylist[0] = 'NEW'

In [26]: mylist
Out[26]: ['NEW', 2, 3]

In [27]: t[0] = 'NEW'

-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_118904\2140988817.py in <module>
----> 1 t[0] = 'NEW'

TypeError: 'tuple' object does not support item assignment

Since lists are mutate and tuples are immutate

In [ ]:
```