

Lists and Tuples Syntax

<code>L = [1, 2, 3, 4, 5]</code>	Lists are created with <code>[]</code>
<code>T = (10, 20, 30, 40, 50)</code>	Tuples are created with <code>()</code>
<code>L[0]</code>	Returns 1st element of <code>L</code> (1)
<code>T[0]</code>	Returns 1st element of <code>T</code> (10)
<code>L[1:4]</code>	Returns 2nd to 4th element of <code>L</code> ([2, 3, 4])
<code>T[1:4]</code>	Returns 2nd to 4th element of <code>T</code> ((20, 30, 40))
<code>L[0:-1:2]</code>	Returns 1st to 2nd last element of <code>L</code> skipping one at a time ([1, 3])
<code>T[0:-1:2]</code>	Returns 1st to 2nd last element of <code>L</code> skipping one at a time ((10, 30))
<code>L[1] = 22</code>	Assigns 22 to 2nd element of <code>L</code> (<code>L == [1, 22, 3, 4, 5]</code>)
<code>T[1] = 22</code>	ERROR: You can't assign anything to tuples
<code>L[0:2] = [11, 22]</code>	Assigns 11 and 22 to 1st and 2nd element of <code>L</code> respectively (<code>L == [11, 22, 3, 4, 5]</code>)

Lists are mutable and Tuples are NOT mutable

Lists - Methods

<code>a = ['a', 'b', 'c']</code>	
<code>b = [1, 3, 2]</code>	
<code>a + b</code>	Returns <code>a</code> concatenated with <code>b</code> (['a', 'b', 'c', 1, 3, 2])
<code>'c' in a</code>	Returns <code>True</code> if 'c' is in the list <code>a</code> and <code>False</code> otherwise (True)
<code>len(a)</code>	Returns the number of elements in <code>a</code> (3)
<code>a.append('d')</code>	Appends 'd' to the end of the list <code>a</code> (<code>a == ['a', 'b', 'c', 'd']</code>)
<code>a.extend(['d', 'e', 'f'])</code>	Appends every element of the iterable to the end of <code>a</code> (<code>a == ['a', 'b', 'c', 'd', 'e', 'f']</code>)
<code>a.insert(1, 'd')</code>	Inserts 'd' to index 1 of <code>a</code> (<code>a == ['a', 'd', 'b', 'c']</code>)
<code>a.pop()</code>	Returns the last element of the list and deletes it from the list. ('c')

Lists - Methods (cont)

<code>a.pop(1)</code>	Returns 2nd element of <code>a</code> and removes it from the list ('b')
<code>a.remove('b')</code>	Removes first occurrence of 'b' in <code>a</code> (<code>a == ['a', 'c']</code>)
<code>a.clear()</code>	Clears the list entirely (<code>a == []</code>)
<code>a.index('b')</code>	Returns the index of the first occurrence of 'b' (1)
<code>a.count('b')</code>	Returns the number of occurrences of 'b' in <code>a</code> (1)
<code>b.sort()</code>	Returns a sorted version of <code>b</code> ([1, 2, 3])
<code>a.reverse()</code>	Reverses the list <code>a</code> (['c', 'b', 'a'])
<code>a.copy()</code>	Returns a copy of <code>a</code>

The `copy()` method returns a list identical to the original, but with a different ID. It means that they are allocated in different places of memory.

Tuple - Methods

<code>t1 = ('a', 'b', 'c')</code>	
<code>t2 = (1, 2, 3)</code>	
<code>t1 + t2</code>	Returns a concatenated version of <code>t1</code> and <code>t2</code>
<code>2 in t2</code>	Returns <code>True</code> if 2 is in <code>t2</code> and <code>False</code> otherwise (True)
<code>len(t1)</code>	Returns the number of elements in <code>t1</code> (3)
<code>t2.count(2)</code>	Returns the number of occurrences of 2 in <code>t2</code> (1)
<code>t2.index(1)</code>	Returns the index of the 1st occurrence of 1 (0)

Lists - Loops 1

```
a = ['one', 'two', 'three']
for i in a:
    print(i)

one
two
three
```

Lists - Loops 2

```
a = ['one', 'two', 'three']
for i in range(len(a)):
    print(f"a[{i}] == {a[i]}")

a[0] == one
a[1] == two
a[2] == three
```

