

Tuple

Tuple

Just as strings are a sequence of characters, tuples are a sequence of objects. This makes their use far more general.

You can set a tuple by separating the objects using commas. For example:

```
[1]: t = 1, 2, 3, 'a', 'b', 'c'

print(t)
```

```
(1, 2, 3, 'a', 'b', 'c')
```

This is called tuple packing.

You can also put brackets around the objects, which is useful if you need to instance a tuple and use it in the same line (for example as a function argument):

```
[2]: print(('a', 1, 'b', 2, 'c', 3))
```

```
('a', 1, 'b', 2, 'c', 3)
```

Like strings, tuples can be indexed and sliced:

```
[3]: print('Index 3:', t[3])
      print('Slice from index 3:', t[3:])
```

Index 3: a

Slice from index 3: ('a', 'b', 'c')

Tuples are also immutable (like strings):

```
[5]: t[2] = 5
```

[illegible]

```
<ipython-input-5-5255d5d095a8> in <module>
----> 1 t[2] = 5
```

TypeError: 'tuple' object does not support item assignment

You can unpack a tuple into multiple variables, just like you can pack multiple values into a tuple:

```
[7]: t = (1, 2, 3)
      print('t is ', t)

      x, y, z = t
      print('x is', x)
      print('y is', y)
      print('z is', z)
```

```
t is (1, 2, 3)
x is 1
y is 2
z is 3
```