CITY
UNIVERSITY OF LONDON
EST 1894

STEM Digital Academy

School of Science & Technology

www.city.ac.uk

Tuples

Programming and Algorithms

Lecture by

Dr Daniil Osudin

```
n = 3
for i in range(1,n+1):
    print("Hello World!")
```

```
Hello World!
Hello World!
Hello World!
```

What will we Cover?

- Introduction to the tuple data structure
- Storing data in tuples

CITY
UNIVERSITY OF LONDON
EST 1894

What is the Purpose of Tuples?

- Like lists, tuples are ordered sequence or collection of data items
  - Data items can be of any data type
- Unlike lists, tuples are immutable, so their values cannot be modified

Examples:

Recording the values of electricity meter (date, number of kWh)

Tuple Data Type

- Tuples are created using parenthesis (), as opposed to lists which are defined using square brackets [] around the elements
- **Examples:**

```
# my_tuple1 contains 3 elements of type int
my_tuple1 = (1, 2, 3)
# my_tuple2 contains 3 elements of type int, string and  float
my_tuple2 = (4, "apples", 4.5)
```

Accessing Elements in Tuples

- Like in lists, elements are indexed:
  - 0 is the index of the first element position
  - 1 is the index of the second element position
  - Tuple length - 1 (or just -1) is the index of the last element position
- Tuple elements can be accessed by using the tuple name followed by the index of the element inside the [ ]
  - `my_tuple2[0]` returns `4`
  - `my_tuple2[1]` returns `"apples"`
  - `my_tuple2[-1]` returns `4.5`

Examples I

Printing the tuple and its elements

```
tuple1 = (1, 2, 3)
print(tuple1)
```

```
(1, 2, 3)
```

```
tuple2 = (1, 2, 3)
print(tuple2[-1])
```

```
3
```

```
tuple3 = (1, 2, 3)
print(tuple3[0])
```

```
1
```

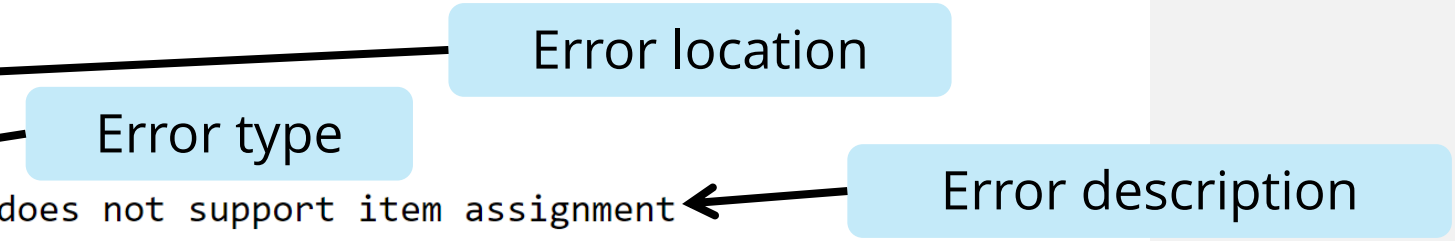
Common Errors in Python

Tuples are **immutable**: once created, position and value of an element cannot change

```
tuple4 = (1, 2, 3)
tuple4[1] += 1
print(tuple4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_6560\1369268017.py in <module>
      1 tuple4 = (1, 2, 3)
----> 2 tuple4[1] += 1
      3 print(tuple4)

TypeError: 'tuple' object does not support item assignment
```



Tuple Properties

- If a tuple has only one value, this is indicated using a comma after the value

```
my_tuple3 = (1,)
```

- A tuple can have elements of different types, including another tuple

```
my_tuple4 = (1, "apples", [1, 2], (3, 4))
```

Examples III

Single element tuple

```
tuple5 = ("a",)
print(tuple5)
```

```
('a',)
```

Lists as elements

```
tuple7 = ("a", ["b", "c"])
print(tuple7)
print(tuple7[1])
```

```
('a', ['b', 'c'])
['b', 'c']
```

Tuples as elements

```
tuple6 = (1, (2, 3))
print(tuple6)
print(tuple6[1])
```

```
(1, (2, 3))
(2, 3)
```

Accessing an element of a tuple element

```
tuple8 = (("a", 2), 5)
print(tuple8[0])
print(tuple8[0][0])
```

```
('a', 2)
a
```

CITY UNIVERSITY OF LONDON EST 1894

Try It Yourself

Enter and run the following statements in the python environment:

```
tuple1 = (5, 7)
sum = tuple1[0] + tuple1[1]
print(sum)
```

```
tuple2 = ("apple", 3.5)
if tuple2[0] == "apple":
 print(tuple2[1])
```

```
tuple3 = ((1, 2, 3), 4)
print(tuple3[1])
print(tuple3[0][1])
```

```
tuple4 = ([9, 3],)
print(tuple4[0])
print(tuple4[-1])
```

CITY
UNIVERSITY OF LONDON
EST 1894