

Data Structure - Tuple

Objectives:

- Tuple Basics
- Tuple Packing and Unpacking

1. How to create a tuple?

A Tuple is a collection of Python objects which is **immutable**, i.e. not modifiable after creation.

Tuple is created with listed of items surrounded by parentheses "()", and separated by comma ",".

- To create an empty tuple, simple use ()
- To create a single-item tuple, need to add **common** "," behind the element. E.g. `tup = (3,)`

Exercise:

- Create a tuple `t` with values 1, 2, 3, 4
- Print it and its type

```
In [1]: 1 t = (1,2,3,4)
        2 print(type(t))

<class 'tuple'>
```

Parentheses is Optional

In fact, parentheses is optional unless it is to create an empty tuple.

```
In [2]: 1 t = 1, 2, 3, 4
        2 print(type(t))

<class 'tuple'>
```

2. Tuple Unpacking

Function with Multiple Returning Values

Function can only return a single value. Multiple values can still be returned as a Tuple.

Exercise:

Define a function `minmax()` which fulfills following conditions.

- It accepts a list as input
- It returns both min and max values of the list

```
In [3]: 1 def minmax(s):  
2         return min(s), max(s)  
3  
4 x, y = minmax([1,2,3,4,5])  
5 print(x, y)
```

1 5

Tuple Unpacking

Tuple can be easily unpacked into multiple values.

- During unpacking, number of variable needs to match number of items in tuple

Exercise:

How to swap two values `x = 10` and `y = 20` in a single statement?

```
In [4]: 1 x = 10  
2 y = 20  
3 x, y = y, x  
4 print(x, y)
```

20 10

Rest Operator *

You can use `*` to hold any number of unpacked values.

```
In [5]: 1 t = (1,2,3,4,5)  
2 a, *b = t  
3 print(a, b)
```

1 [2, 3, 4, 5]