



Tuple

Agenda

1

Tuple

2

Accessing tuple items

3

Slicing

4

Updating tuple

5

for loop with tuple

6

in keyword with tuple

Tuple



Tuple

- Tuple is a collection of **immutable** elements.
- It maintains the insertion order and supports index based access.
- It can contain elements of different types.
- Elements are enclosed in parentheses ().
- If there is only one element it must be terminated with a comma.

Examples:

```
t_1 = (1, '@', 'wipro', 5.5)
```

```
t_2 = (100, 200, 300)
```

```
t_3 = ('john',)
```

Accessing tuple items



Accessing tuple items

- Items in a tuple can be accessed with both positive and negative indexes.
- Positive index starts from 0.
- Negative index starts from -1 from the last item.

Program:

```
# 0      1      2
tup = (100, 200, 300)
# -3     -2     -1
print(tup[1]) #accessing second element
print(tup[-1]) #accessing last element
```

Output:

200

300

Slicing



Slicing

- We can extract a set of items from a tuple by mentioning the start and end indexes.
- While specifying this start and end range, the returned value will be a new tuple with the specified items.

Program:

```
      # 0      1      2      3
tup = ('car', 'bus', 'ship', 'train')
print(tup[1:3])
```

Output:

```
('bus', 'ship')
```

Ending index is
excluded.

Slicing

Negative slicing:

Program:

```
      #  -4      -3      -2      -1
tup = ('car', 'bus', 'ship', 'train')
print(tup[-4:-2])
```

Output:

```
('car', 'bus')
```

Updating tuple



Updating tuple

Tuples are immutable, but we can update the items in this way:

1. Convert the tuple into a list using `list()` constructor.
2. Update the list.
3. Convert the list back to tuple using `tuple()` constructor.

Program:

```
tup = ('car', 'bus', 'ship')  
li = list(tup)  
li[0] = 'flight'  
tup = tuple(li)  
print(tup)
```

Output:

```
('flight', 'bus', 'ship')
```

for loop with tuple



for loop with tuple

We can loop through the tuple items by using a for loop:

Program:

```
tup = ('car', 'bus', 'ship', 'train')  
  
for item in tup:  
    print(item)
```

Output:

```
car  
bus  
ship  
train
```

in keyword with tuple



in keyword with tuple

To check if a specified item is present in a tuple we can use the in keyword:

Program:

```
tup = ('car', 'bus', 'ship', 'train')  
  
if 'flight' in tup:  
    print("Yes present")  
else:  
    print("Not present")
```

Output:

Not present



Thank you