Tuples

Python Tuples

- Tuple

  A tuple is a collection which is ordered and unchangeable. In Python tuples are written with round brackets.

- Example
- Create a Tuple:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

```
('apple', 'banana', 'cherry')
```

- **Access Tuple Items**

  You can access tuple items by referring to the index number, inside square brackets:

- Example
- Print the second item in the tuple:

  thistuple = ("apple", "banana", "cherry")

  print(thistuple[1])

```
thistuple = ("apple", "banana", "cherry")
print(thistuple[1])
```

```
banana
```

- **Negative Indexing**

Negative indexing means beginning from the end, -1 refers to the last item, -2 refers to the second last item etc.

- Example
- Print the last item of the tuple:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple[-1])
```

```
thistuple = ("apple", "banana", "cherry")
print(thistuple[-1])
```

```
cherry
```

- **Range of Indexes**

  You can specify a range of indexes by specifying where to start and where to end the range.

  When specifying a range, the return value will be a new tuple with the specified items.

- Example
- Return the third, fourth, and fifth item:

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[2:5])
```

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[2:5])

#This will return the items from position 2 to 5.
```

```
('cherry', 'orange', 'kiwi')
```

- **Range of Negative Indexes**

  Specify negative indexes if you want to start the search from the end of the tuple:

  - Example
  - This example returns the items from index -4 (included) to index -1 (excluded)

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[-4:-1])
```

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[-4:-1])

#Negative indexing means starting from the end of the tuple.
```

```
('orange', 'kiwi', 'melon')
```

- **Loop Through a Tuple**
  You can loop through the tuple items by using a for loop.
  Example
  Iterate through the items and print the values:

  thistuple = ("apple", "banana", "cherry")
  for x in thistuple:
    print(x)

```
thistuple = ("apple", "banana", "cherry")
for x in thistuple:
  print(x)
```

```
apple
banana
cherry
```

- **Check if Item Exists**

To determine if a specified item is present in a tuple use the in keyword:

- Example
- Check if "apple" is present in the tuple:

```
thistuple = ("apple", "banana", "cherry")
if "apple" in thistuple:
 print("Yes, 'apple' is in the fruits tuple")
```

- **Tuple Length**

To determine how many items a tuple has, use the len() method:

- Example
- Print the number of items in the tuple:

```
thistuple = ("apple", "banana", "cherry")
print(len(thistuple))
```

```
thistuple = ("apple", "banana", "cherry")
print(len(thistuple))
```

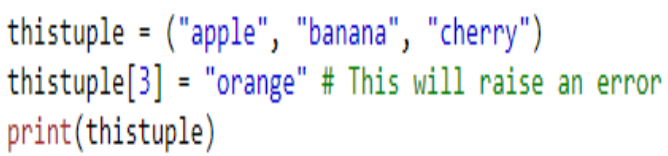
```
3
```

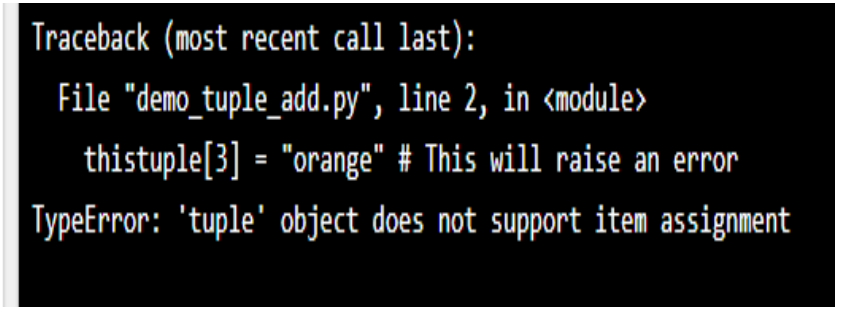
- Add Items

Once a tuple is created, you cannot add items to it. Tuples are unchangeable.

- Example
- You cannot add items to a tuple:

```
thistuple = ("apple", "banana", "cherry")
thistuple[3] = "orange" # This will raise an error
print(thistuple)
```



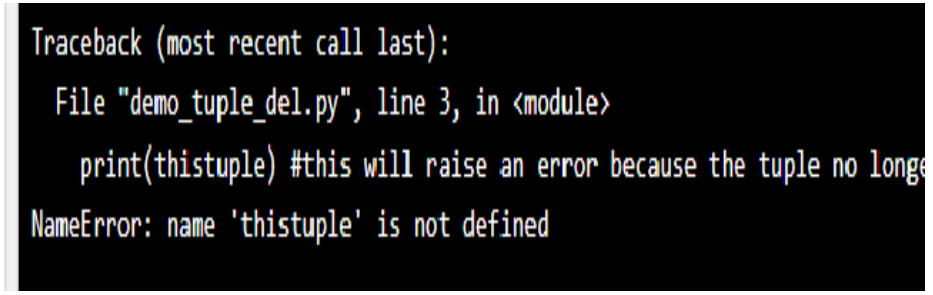


- **Remove Items**
  - Note: You cannot remove items in a tuple.

  Tuples are unchangeable, so you cannot remove items from it, but you can delete the tuple completely:

- Example
- The del keyword can delete the tuple completely:

```
thistuple = ("apple", "banana", "cherry")
del thistuple
print(thistuple) #this will raise an error because the tuple no longer exists
```



- **Join Two Tuples**

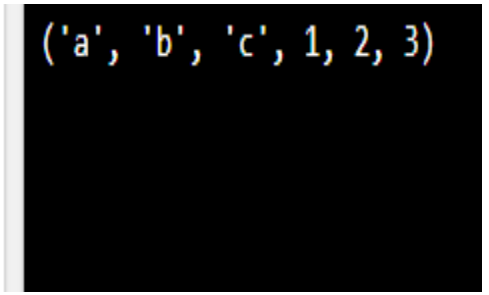
To join two or more tuples you can use the + operator:

- Example
- Join two tuples:

```
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)
tuple3 = tuple1 + tuple2
print(tuple3)
```

```
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)

tuple3 = tuple1 + tuple2
print(tuple3)
```



- **Python Tuple count() Method**
- Example
- Return the number of times the value 5 appears in the tuple:

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.count(5)
print(x)
```

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = thistuple.count(5)

print(x)
```

- **Python Tuple index() Method**
- Example
- Search for the first occurrence of the value 8, and return its position:

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.index(8)
print(x)
```

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = thistuple.index(8)

print(x)
```

Thank You