6. Tuple

April 21, 2023

1 Introduction

- Tuple is a collection which allows us to put many objects in a single object.
- Same definition as List.
- Difference between Tuple and List is
  - Lists are mutable
  - Tuples are immutable

```
[1]: movies = ( "Tenet", "Ice Age", "Gravity", "Joker", "Spiderman", "Avengers:␣
      ↪Endgame")
```

```
[2]: movies
```

```
[2]: ('Tenet', 'Ice Age', 'Gravity', 'Joker', 'Spiderman', 'Avengers: Endgame')
```

```
[3]: type(movies)
```

```
[3]: tuple
```

```
[4]: len(movies)
```

```
[4]: 6
```

```
[5]: mixed_tuple = ( "BLR", 36.5, "HYD", 42.5, True, 45.2, None )
```

2 Accessing elements of a tuple

```
[6]: movies
```

```
[6]: ('Tenet', 'Ice Age', 'Gravity', 'Joker', 'Spiderman', 'Avengers: Endgame')
```

```
[7]: movies[2]
```

```
[7]: 'Gravity'
```

```
[8]: movies[-1]
```

```
[8]:  'Avengers: Endgame'

[9]:  movies[6]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
/tmp/ipykernel_8050/2057263709.py in <module>
----> 1 movies[6]

IndexError: tuple index out of range
```

3   Slicing

```
[10]:  movies

[10]:  ('Tenet', 'Ice Age', 'Gravity', 'Joker', 'Spiderman', 'Avengers: Endgame')

[11]:  movies[1:4]

[11]:  ('Ice Age', 'Gravity', 'Joker')

[12]:  movies[:3]

[12]:  ('Tenet', 'Ice Age', 'Gravity')

[13]:  movies[4:]

[13]:  ('Spiderman', 'Avengers: Endgame')

[14]:  movies[-4:-2]

[14]:  ('Gravity', 'Joker')
```

4   Modifying a Tuple

```
[15]:  movies

[15]:  ('Tenet', 'Ice Age', 'Gravity', 'Joker', 'Spiderman', 'Avengers: Endgame')

[16]:  movies[4]

[16]:  'Spiderman'

[17]:  movies[4] = "Ant Man"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipykernel_8050/4177106749.py in <module>
----> 1 movies[4] = "Ant Man"

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

5 Methods of Tuple

1. index(elem)
2. count(elem)

5.1 index(elem)

- Return index of the first occurrence of elem

```
[18]: a = (2, 6, 8, 3, 5, 9, 1, 0, 2, 8, 6, 6)
```

```
[19]: len(a)
```

[19]: 12

```
[20]: movies = ( "Tenet", "Ice Age", "Gravity", "Joker", "Spiderman", "Avengers:␣
      ↪Endgame")
```

```
[21]: len(movies)
```

[21]: 6

```
[22]: movies.index("Joker")
```

[22]: 3

```
[23]: movies.index("Joker2")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipykernel_8050/1893580528.py in <module>
----> 1 movies.index("Joker2")

ValueError: tuple.index(x): x not in tuple
```

```
[24]: a.index(8,5)
```

[24]: 9

```
[25]: movies.index("Joker", 2, 5)
```

```
[25]: 3
```

5.2 count(elem)

- Return the number of elem items present in it

```
[26]: movies.count("Joker")
```

```
[26]: 1
```

```
[27]: a
```

```
[27]: (2, 6, 8, 3, 5, 9, 1, 0, 2, 8, 6, 6)
```

```
[28]: a.count(6)
```

```
[28]: 3
```

6 Deleting a Tuple

```
[29]: movies
```

```
[29]: ('Tenet', 'Ice Age', 'Gravity', 'Joker', 'Spiderman', 'Avengers: Endgame')
```

```
[30]: del movies
```

```
[31]: movies
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_8050/878749064.py in <module>
----> 1 movies

NameError: name 'movies' is not defined
```

7 Operators on List

1. Membership
2. Arithmetic

7.1 Membership

- in

- not in

```
[32]: a
```

```
[32]: (2, 6, 8, 3, 5, 9, 1, 0, 2, 8, 6, 6)
```

```
[33]: 5 in a
```

```
[33]: True
```

```
[34]: 4 not in a
```

```
[34]: True
```

7.2 Arithmetic

- +
- *

```
[35]: a
```

```
[35]: (2, 6, 8, 3, 5, 9, 1, 0, 2, 8, 6, 6)
```

```
[36]: b = ("A", "B")
```

```
[37]: a + b
```

```
[37]: (2, 6, 8, 3, 5, 9, 1, 0, 2, 8, 6, 6, 'A', 'B')
```

```
[ ]: b * 5
```

8 Nested Tuples

```
[38]: nestl = ("Monday", "Today", [1, 2, 3])
```

```
[39]: nestl[2]
```

```
[39]: [1, 2, 3]
```

```
[40]: nestl[2][0]
```

```
[40]: 1
```

A matrix can be represented using a nested list

```
[41]: mx = ((1, 2, 3), (4, 5, 6), (7, 8, 9))
```

```
[42]: mx
```

```
[42]: ((1, 2, 3), (4, 5, 6), (7, 8, 9))
```

```
[43]: mx[0][2]
```

```
[43]: 3
```

9 For Loop on Tuple

```
[45]: movies = ( "Tenet", "Ice Age", "Gravity", "Joker", "Spiderman", "Avengers:␣
       ↪Endgame")
```

```
[46]: for i in movies:
          print(i.lower())
          print('.'*10)
```

```
tenet
…
ice age
…
gravity
…
joker
…
spiderman
…
avengers: endgame
…
```