tupless

April 21, 2024

tuple is one of the most common datatype in python tuple is immutable datatype —> cannot be changed after creation

```python
[ ]: "1- define a list "
     t = ()
     myt = tuple()
```

```python
[1]: "tuple can hold different data from different datatypes"
     myt = (34,4, "iti", 344.32, "ahmed", True, ['kubernates', 'GCP', "ansible"],␣
      ↪"iti")
```

```python
[3]: """ each element is assigned to an index --> index start from 0 """
     print(myt[4])
     print(myt[6][2])
```

```
ahmed
ansible
```

```python
[4]: "get len of tuple"
     print(len(myt))
```

```
8
```

```python
[5]: "count element occurence"
     print(myt.count("iti"))
```

```
2
```

```python
[6]: """get index of element """
     print(myt.index("iti"))
```

```
2
```

```python
[7]: """slicing the tuple"""
     print(myt[3:5])
```

```
(344.32, 'ahmed')
```

```python
[8]: print(myt[::-1]) # new tuple
```

```
('iti', ['kubernates', 'GCP', 'ansible'], True, 'ahmed', 344.32, 'iti', 4, 34)
```

```
[ ]: """ loop over the tuple """
```

```
[10]: for element in myt:
          print(f"element = {element}")
```

```
element = 34
element = 4
element = iti
element = 344.32
element = ahmed
element = True
element = ['kubernates', 'GCP', 'ansible']
element = iti
```

```
[11]: """check element in tuple or not """
      print("iti" in myt)
```

```
True
```

```
[12]: """---> tuple concat"""
      t1 = ("python", "mongo", "javascript", "sql")
      t2= ("jenkins", "GCP", "Ansible", "Kubernates", "Docker")
```

```
[13]: lst_of_courses = t1  + t2 # NEW tuple
      print(lst_of_courses)
```

```
('python', 'mongo', 'javascript', 'sql', 'jenkins', 'GCP', 'Ansible',
'Kubernates', 'Docker')
```

Casting string to a tuple.

```
[14]: name = 'noha'
      name = tuple(name)
      print(name)
```

```
('n', 'o', 'h', 'a')
```

min-max

```
[15]: print(min((3,454,23,323,12,21)))
      print(max((3,454,23,323,12,21)))
```

```
3
454
```

for loop and index

```
[16]: index  = 0
      for item in myt:
          print(f"{item},{myt.index(item)}, {index}")
          index += 1
```

```
34,0, 0
4,1, 1
iti,2, 2
344.32,3, 3
ahmed,4, 4
True,5, 5
['kubernates', 'GCP', 'ansible'],6, 6
iti,2, 7
```

enum

```
[17]: myt_enum = enumerate(myt)
      print(myt_enum)
      # print(tuple(myt_enum))
```

```
<enumerate object at 0x72a477ed37e0>
```

use it with loops

```
[18]: for i , value in myt_enum:
          print(f"i={i}, item = {value}")
```

```
i=0, item = 34
i=1, item = 4
i=2, item = iti
i=3, item = 344.32
i=4, item = ahmed
i=5, item = True
i=6, item = ['kubernates', 'GCP', 'ansible']
i=7, item = iti
```

[ ]: