untitled6

September 10, 2024

1 Python Tuples

- Python tuples are a type of data structure that is very similar to lists. The main difference between the two is that tuples are immutable, meaning they cannot be changed once they are created. This makes them ideal for storing data that should not be modified.
- Tuples are written with round brackets.

Example.

[1]:

```
mytuple = ("watermelon", "apple", "orange", "banana")
print(mytuple)
print(type(mytuple))
```

```
('watermelon', 'apple', 'orange', 'banana')
<class 'tuple'>
```

[3]:

```
mytuple = ("orange", 5, 5.9, 2+3j)
print(mytuple)
```

```
('orange', 5, 5.9, (2+3j))
```

[4]:

```
mytuple = ("watermelon",)
print(mytuple)
```

```
('watermelon',)
```

2 Immutability

once a tuple is created you can't make any changes in the tuple.

[5]:

```
mytuple = ("watermelon", "apple", "orange", "banana")
mytuple[1] = "dog"
print(mytuple)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-01444f69552f> in <cell line: 2>()
      1 mytuple = ("watermelon", "apple", "orange", "banana")
----> 2 mytuple[1] = "dog"
```

```
      3 print(mytuple)

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

3 Tuple Operations

1.Concatination

combine two tuples

[10]:
```
tuple1 = ("watermelon", "apple", "orange", "banana")
tuple2 = (1, 2, 3, 4)
tuple3 =  tuple1+tuple2
print(tuple3)
```

```
('watermelon', 'apple', 'orange', 'banana', 1, 2, 3, 4)
```

2.Repetition

repeating the tuple

[11]:
```
tuple1 = ("watermelon", "apple", "orange", "banana")
tuple2 = tuple1 * 3
print(tuple2)
```

```
('watermelon', 'apple', 'orange', 'banana', 'watermelon', 'apple', 'orange',
'banana', 'watermelon', 'apple', 'orange', 'banana')
```

4 Tuples Methods

1.index():

returns the index of the item.

[6]:
```
mytuple = ("watermelon", "apple", "orange", "banana")
item = mytuple.index("orange")
print(item)
```

```
2
```

2.Count():

returns the count of the item

[9]:
```
mytuple = ("orange", "watermelon", "apple", "orange")
item = mytuple.count("orange")
print(item)
```

```
2
```

3.min():

returns the minimum element in a tuple

[12]:

```python
mytuple = ("watermelon", "apple", "orange", "banana")
item = min(mytuple)
print(item)
```

```
apple
```

4.len():

returns the length of the tuple

[13]:

```python
mytuple = ("watermelon", "apple", "orange", "banana")
item = len(mytuple)
print(item)
```

```
4
```

5.max():

returns the maximum of element in a tuple.

[14]:

```python
mytuple = ("watermelon", "apple", "orange", "banana")
item = max(mytuple)
print(item)
```

```
watermelon
```

6.del():

delete the entire tuple.

[19]:

```python
mytuple = ("watermelon", "apple", "orange", "banana")
del mytuple
print(mytuple)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-19-a104a35608f6> in <cell line: 3>()
      1 mytuple = ("watermelon", "apple", "orange", "banana")
      2 del mytuple
----> 3 print(mytuple)

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