Pierian-Data resetting for 2020 updates

2 contributors

266 lines (266 sloc) | 6.59 KB

Tuples

In Python tuples are very similar to lists, however, unlike lists they are *immutable* meaning they can not be changed. You would use tuples to present things that shouldn't be changed, such as days of the week, or dates on a calendar.

In this section, we will get a brief overview of the following:

```
1.) Constructing Tuples
2.) Basic Tuple Methods
3.) Immutability
4.) When to Use Tuples
```

You'll have an intuition of how to use tuples based on what you've learned about lists. We can treat them very similarly with the major distinction being that tuples are immutable.

Constructing Tuples

The construction of a tuples use () with elements separated by commas. For example:

In [1]:
```
# Create a tuple
t = (1,2,3)
```

In [2]:
```
# Check len just like a list
len(t)
```

Out[2]:
```
3
```

In [3]:
```
# Can also mix object types
t = ('one',2)

# Show
t
```

Out[3]:
```
('one', 2)
```

In [4]:
```
# Use indexing just like we did in lists
t[0]
```

Out[4]:
```
'one'
```

In [5]:
```
# Slicing just like a list
t[-1]
```

Out[5]:
```
2
```

Basic Tuple Methods

Tuples have built-in methods, but not as many as lists do. Let's look at two of them:

In [6]:
```
# Use .index to enter a value and return the
t.index('one')
```

Out[6]:
```
0
```

In [7]:
```
# Use .count to count the number of times a
t.count('one')
```

Out[7]:
```
1
```

Immutability

It can't be stressed enough that tuples are immutable. To drive that point home:

In [8]:
```
t[0]= 'change'
```

```
---------------------------------------------
------------------------------
TypeError                                 Tr
aceback (most recent call last)
 in ()
----> 1 t[0]= 'change'

TypeError: 'tuple' object does not support i
tem assignment
```

Because of this immutability, tuples can't grow. Once a tuple is made we can not add to it.

In [9]:
```
t.append('nope')
```

```
---------------------------------------------
------------------------------
AttributeError                            Tr
```