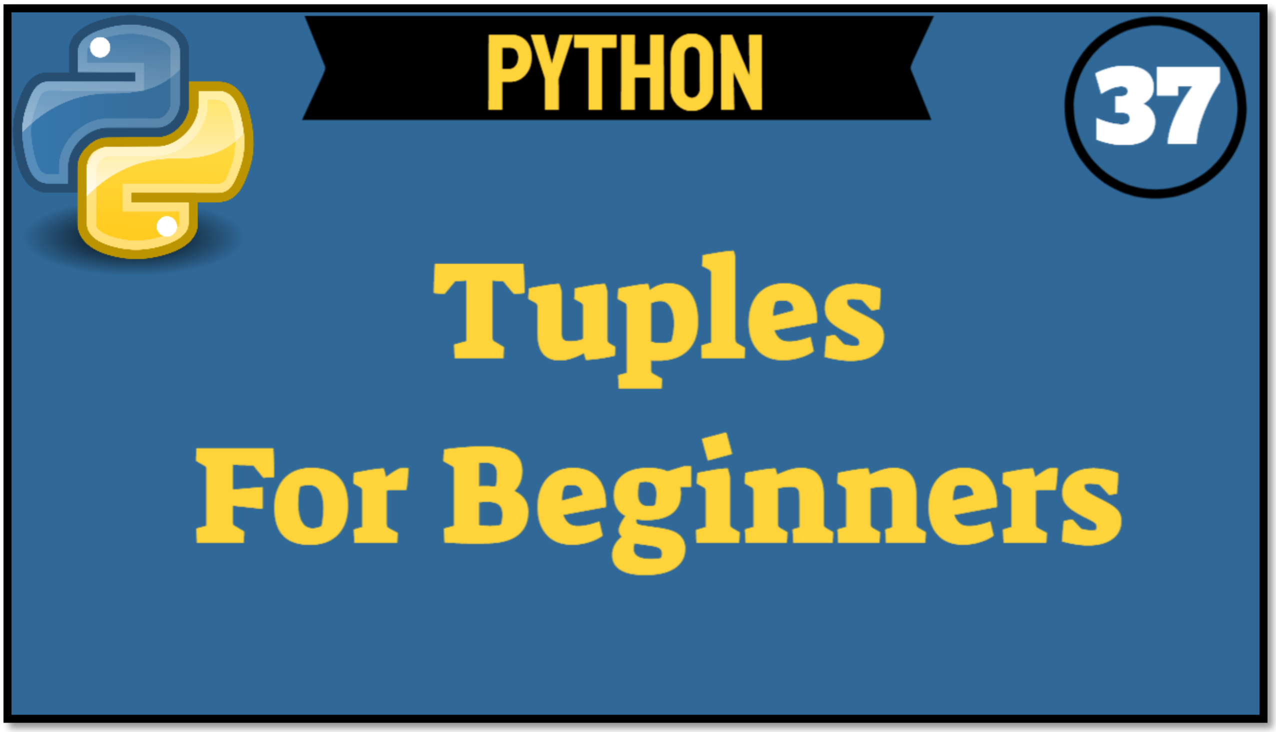
Rex Jones II

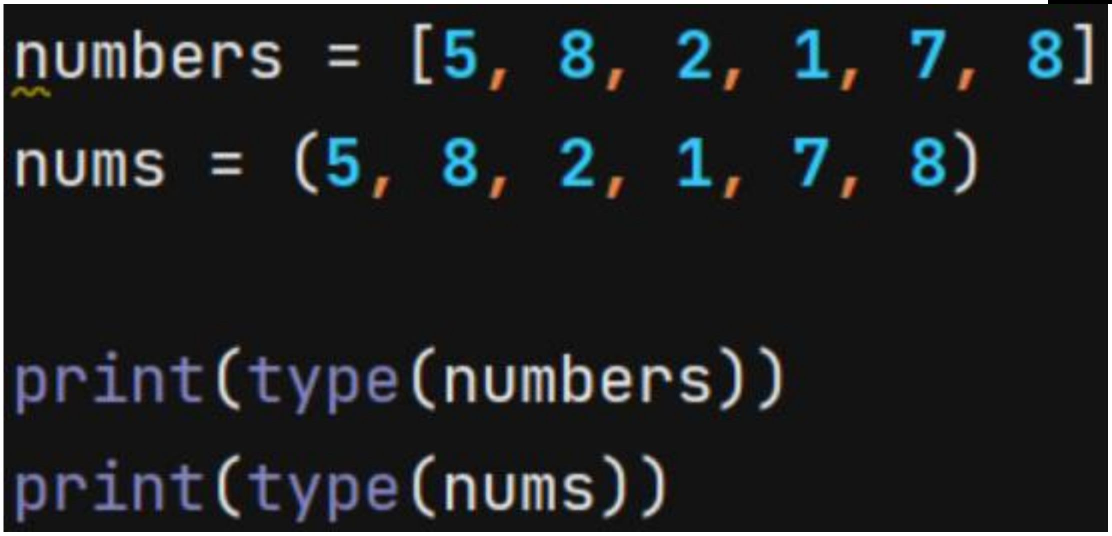
Python Tuples



Python Video = https://youtu.be/7CNiEnmwnOE

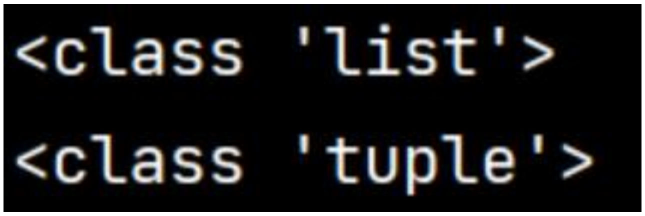
In this session, we are going to look at tuples. A tuple data type is similar to the list data type. However, there are 2 differences between a tuple and a list. First difference is the tuple data type requires parentheses while a list requires square brackets. Second, a tuple is immutable and a list is mutable. Therefore, we cannot change, add, or remove values from the tuple data type. If you are interested in more content, feel free to read the transcript on GitHub or download the source code. Also, like this video and subscribe to my channel. Plus follow me on Twitter, connect with me on LinkedIn and Facebook.

In the IDE, let me show the difference between a tuple and a list. Copy the list data numbers and I'm going to paste them. Paste one more time for the tuple data type and change numbers to nums. Next, change the square brackets [] to opening and closing parentheses (). That's it. We can print() the type() by writing then pass in numbers. Copy and paste then change numbers to nums.



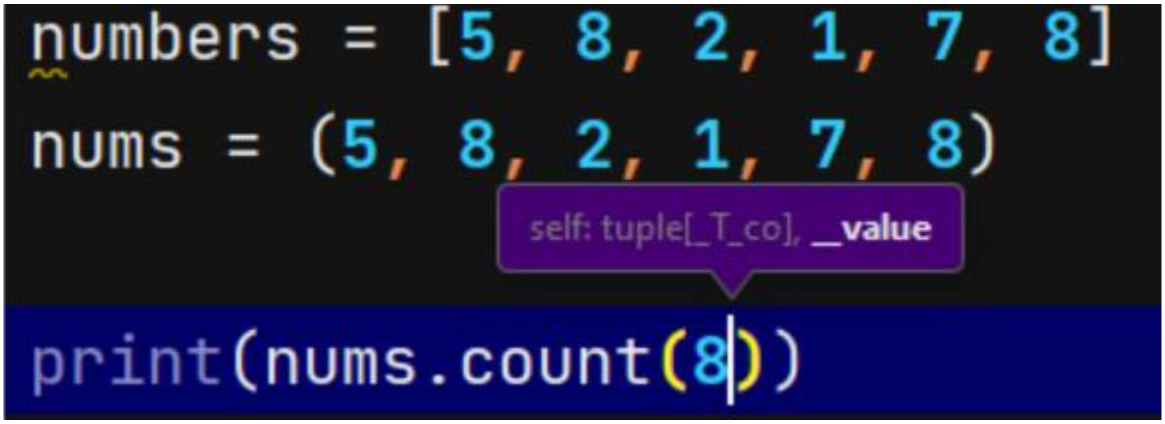
```
numbers = [5, 8, 2, 1, 7, 8]
nums = (5, 8, 2, 1, 7, 8)


print(type(numbers))
print(type(nums))
```

When I run, the console shows list and tuple.



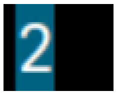
```
<class 'list'>
<class 'tuple'>
```

When it comes to methods, the tuple data type only has 2 methods. I'm going to print(nums.) and we see count and index. However, the list data type has many methods print(numbers.) that includes count and index. There are more methods for the list data type because they involve mutating better known as changing a value. I'm going to print(nums.count): count let's us know how many times a value shows up in the tuple. Pass in (8) then run.



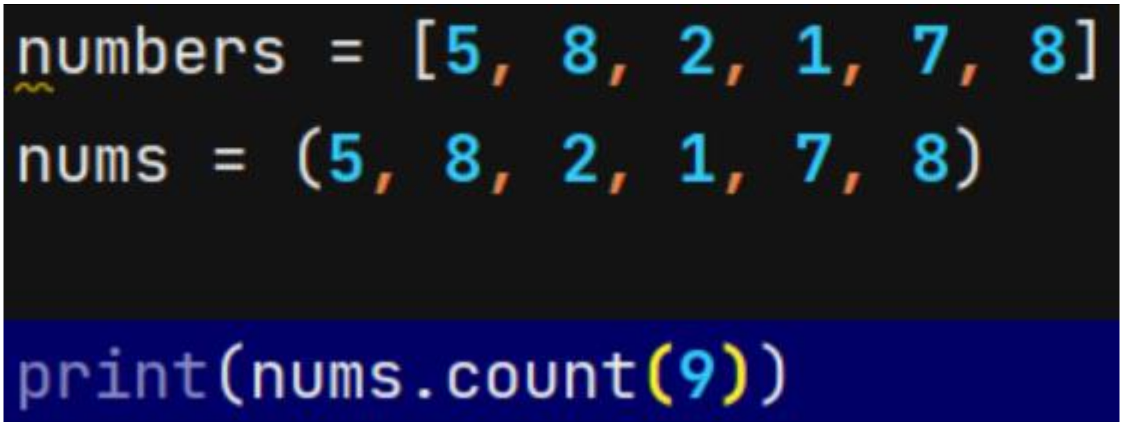
```
numbers = [5, 8, 2, 1, 7, 8]
nums = (5, 8, 2, 1, 7, 8)


print(nums.count(8))
```

As expected, the console returns 2 because there are two number 8's in the tuple.



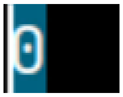
```
2
```

How about if change 8 to a number like 9?



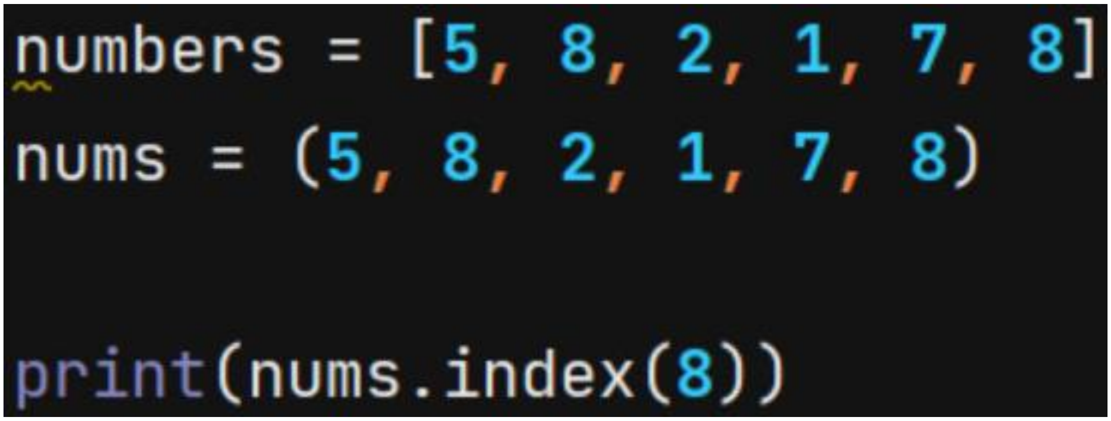
```
numbers = [5, 8, 2, 1, 7, 8]
nums = (5, 8, 2, 1, 7, 8)

print(nums.count(9))
```

The console returns 0 because there are no number 9's in the tuple.



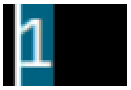
```
0
```

How about number 5? We see 1 because there's only one 5 in the tuple. The index method let's us know the position of a value so change count to index then pass in number 8.



```
numbers = [5, 8, 2, 1, 7, 8]
nums = (5, 8, 2, 1, 7, 8)


print(nums.index(8))
```

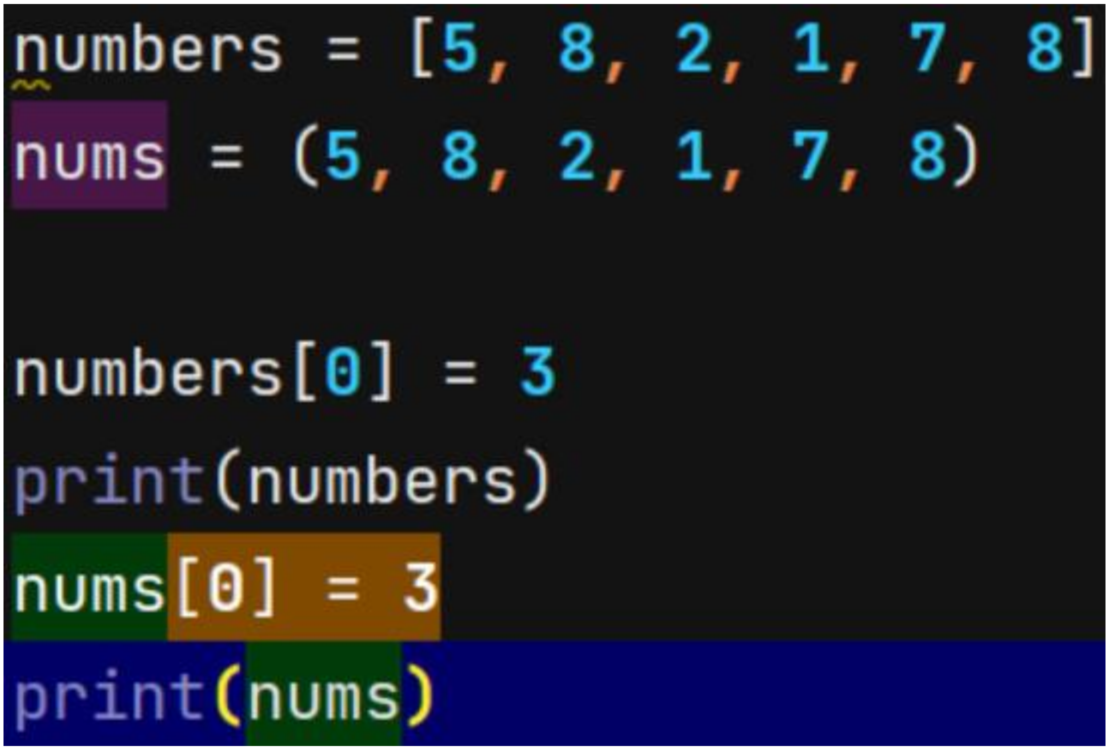
Since there are two number 8's, Python will only return the first 8 in position 1. Run and we see 1 in the console.



```
1
```

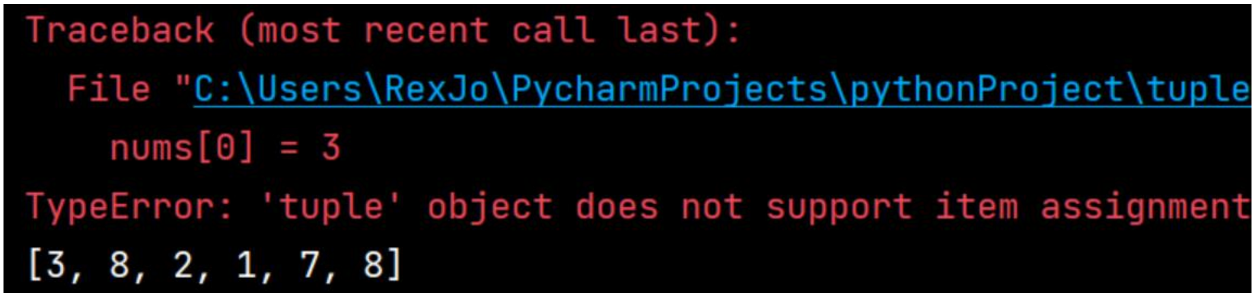
The tuple data type only has 2 methods because they are immutable (cannot be modified).

If we try to modify a tuple then Python will return a TypeError. For example, if I write nums[0] = 3. This is a command to try and change the 1st item from 5 to 3. We already see the brown background. Hover and it says "Tuples don't support item assignment". I'm going to write for the list It's no problem if we assign a value to the 1st item for a list data type by writing numbers[0] = 3. Also print(numbers) list and I'm going to also print(nums) tuple.



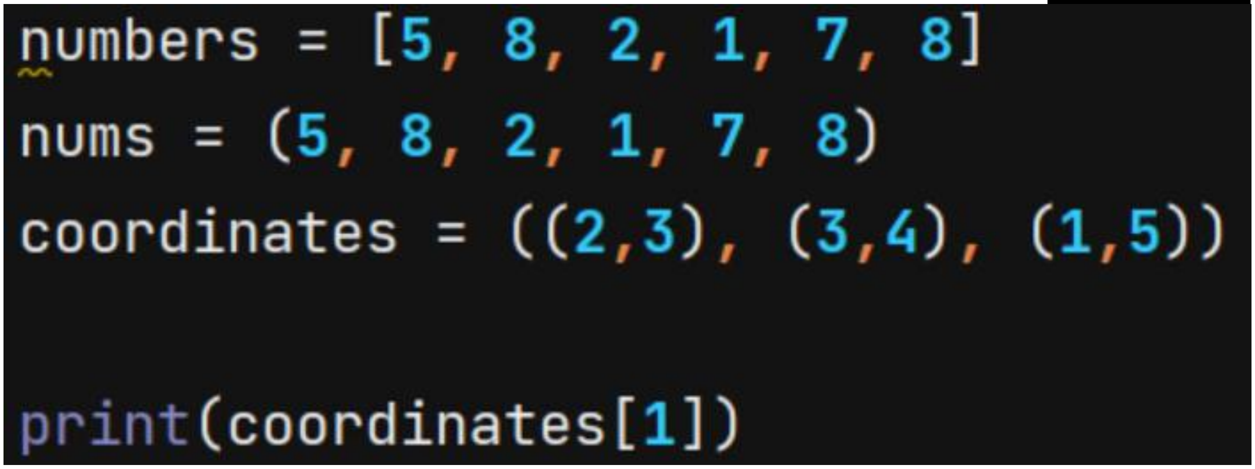
```
numbers = [5, 8, 2, 1, 7, 8]
nums = (5, 8, 2, 1, 7, 8)

numbers[0] = 3
print(numbers)
nums[0] = 3
print(nums)
```

Run and the console returns TypeError: 'tuple' object does not support item assignment but shows the updated list that includes 3 as the 1st number. That's because it's no problem assigning a different value to a list data type.



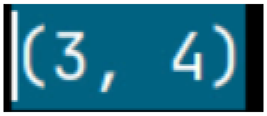
```
Traceback (most recent call last):
  File "C:\Users\RexJo\PycharmProjects\pythonProject\tuple
    nums[0] = 3
TypeError: 'tuple' object does not support item assignment
[3, 8, 2, 1, 7, 8]
```

Most engineers use a list data type because there are more use cases to change, add, or remove a value. However, the tuple data type maintains protection when we do not want a value or set of values to change. For example, if we had a program that involved coordinates. Coordinates are a set of numbers that identifies a point on a line. We should not change a coordinate because it locates a point. I'll write coordinates = ((2, 3), (3, 4), (1, 5)). Two and Three is one coordinate. Three – Four and One – Five are the other coordinates. Print the 2nd coordinate which is (3, 4) by writing print(coordinates[1]).



```
numbers = [5, 8, 2, 1, 7, 8]
nums = (5, 8, 2, 1, 7, 8)
coordinates = ((2,3), (3,4), (1,5))

print(coordinates[1])
```

The index position always starts at 0. Run and the console returns (3, 4). This is a use case for a tuple.



```
(3, 4)
```

Contact Info

- ✓ Email Rex.Jones@Test4Success.org
- ✓ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✓ Facebook https://facebook.com/JonesRexII
- ✓ Twitter https://twitter.com/RexJonesII
- ✓ GitHub https://github.com/RexJonesII/Free-Videos
- ✓ LinkedIn https://www.linkedin.com/in/rexjones34/