



How to Use Tuples in Python

Now we're going to see how to use **tuples** in Python. Earlier we got a little introduction to them. Basically tuple allows you to take the return and assign it into components. So if you have a function that's returning three different values, you can assign each of those values into a tuple.

Here are some sample data for this lesson:

```
customer1 = "John Doe"
pickupLocation1 = "350 5th Ave"
package1 = customer1 + ":" + pickupLocation1

customer2 = "Jane Doe"
pickupLocation2 = "100 7th Ave"
package2 = customer2 + ":" + pickupLocation2

customer3 = "Joe Daniels"
pickupLocation3 = "11 1st Ave"
package = customer3 + ":" + pickupLocation3

packages2 = {customer1:pickupLocation1,customer2:pickupLocation2,customer3:pickupLocation3}
```

For example, we can now split by semi colon:

```
a,b = package1.split(":", 1)
print(a)
#John Doe
print(b)
#350 5th Ave
```

`a` will contain part of the string to the left of `:` and `b` - the right part.

What's more, you can store the result in a list:

```
tuplePackage = package1.split(":", 1)
print(tuplePackage[0])
```

I'm just putting everything into one variable.

Tuple Pair

Tuples are also used in `for` loops. The syntax should already be familiar to you:

```
for (k,v) in packages2.items():
    print(k + " address=" + v)
#output
#John Doe address=350 5th Ave
#Jane Doe address=100 7th Ave
#Joe Daniels address=11 1st Ave
```

Creating A Tuple

```
numbers = (1,)
print(numbers[0])
# 1
numbers = ("1st street", "2nd street", "3rd street", "4th street")
print(numbers[1:4])
```

Note that tuples can be referenced just like lists. What's more, you can print out a bunch of items by saying `numbers[1:4]`.

Can You Reassign Tuples Values?

```
test = (1,2,3,4,5)
print(test[1:3])

#reassignment error
#test[0] = 22
test = (22, 33, 44, 55, 66)
print(test[0])
```

You can't assign or reassign the tuple value - you will get an error. What you can do, however, is reassign variable's contents completely just like showed in this example.