Tuples behind the scenes! Assigning values and Returning tuples explained with diagrams!

Let's see how tuples work, both when you return a tuple from a function and when you assign values from tuples to variables on left hand side tuples.

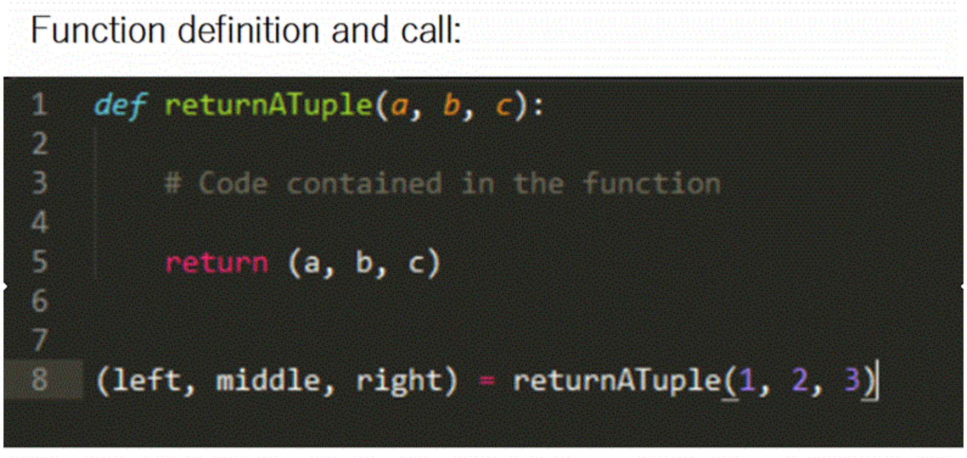
So... Let’s start! :-)

We've studied functions, their structure and how formal and actual parameters work, but let's refresh our knowledge a bit...

Here we have a function definition, it may contain code inside it or not, but the important thing for us right now is that ***it returns a Tuple!!!***

On line 8 we see our function call (when magic starts).

- We can see a tuple on the left hand side containing three variables: `left`, `middle` and `right`. We are declaring them on this line, so they don't have a value yet, until our function returns a tuple with the corresponding values.
- On the right hand side we call our function `returnATuple()` with *actual parameters* `1`, `2`and `3`.

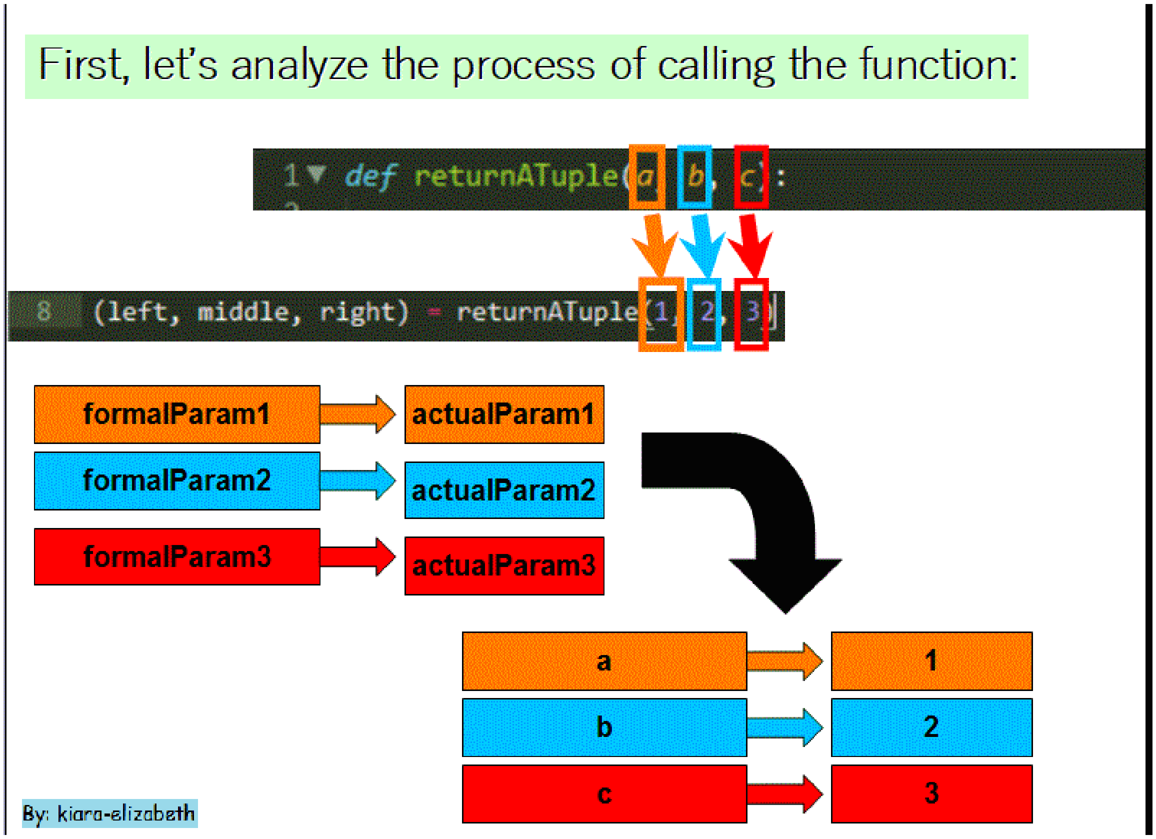


Function definition and call:

```
1 def returnATuple(a, b, c):
2
3     # Code contained in the function
4
5     return (a, b, c)
6
7
8 (left, middle, right) = returnATuple(1, 2, 3)
```

- How will they be assigned to our formal parameters in our function definition?

The first actual parameter will be assigned to the first formal parameter, and so on (as you can see in the diagram)



First, let's analyze the process of calling the function:

```
def returnATuple(a, b, c):
```

```
(left, middle, right) = returnATuple(1, 2, 3)
```

formalParam1 → actualParam1

formalParam2 → actualParam2

formalParam3 → actualParam3

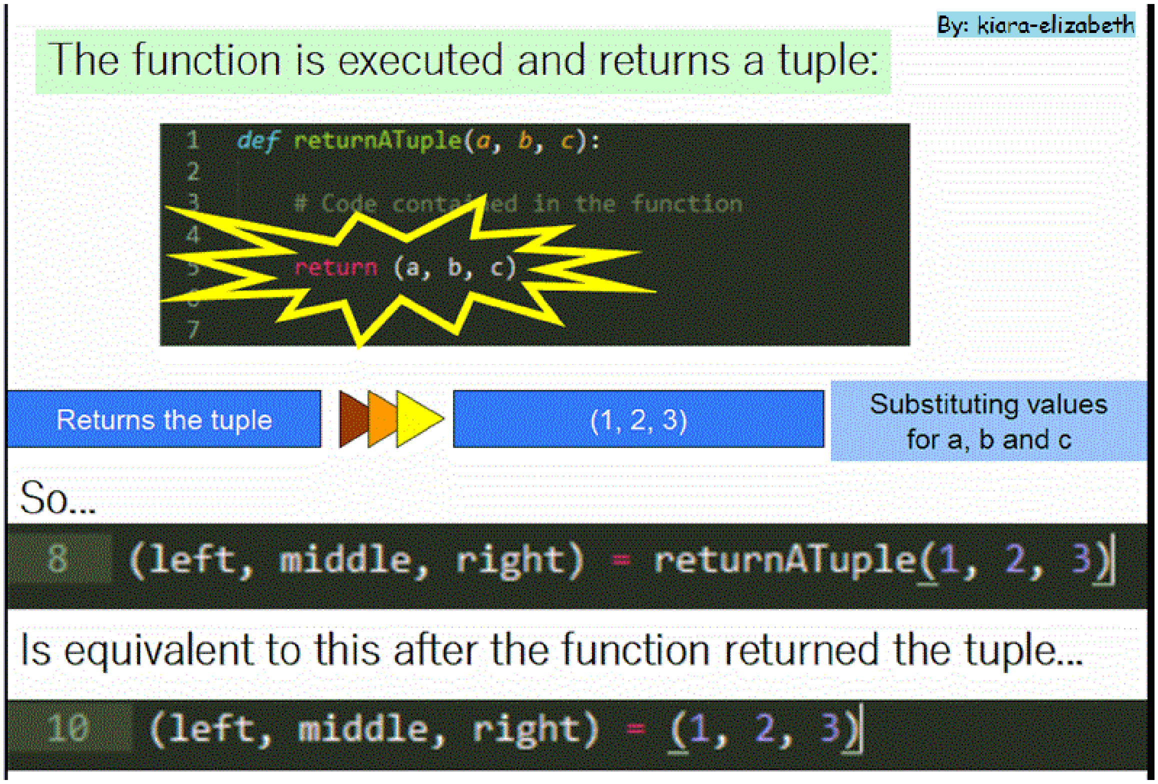
a → 1

b → 2

c → 3

By: kiara-elizabeth

When the function is executed, it will return a tuple (a, b, c). In this case a=1, b=2, c=3 so the tuple returned will be (1, 2, 3) and **you can think of the expression at this point as being replaced by the tuple returned by the function.**



By: kiara-elizabeth

The function is executed and returns a tuple:

```
def returnATuple(a, b, c):

    # Code contained in the function

    return (a, b, c)

```

Returns the tuple → (1, 2, 3)

Substituting values for a, b and c

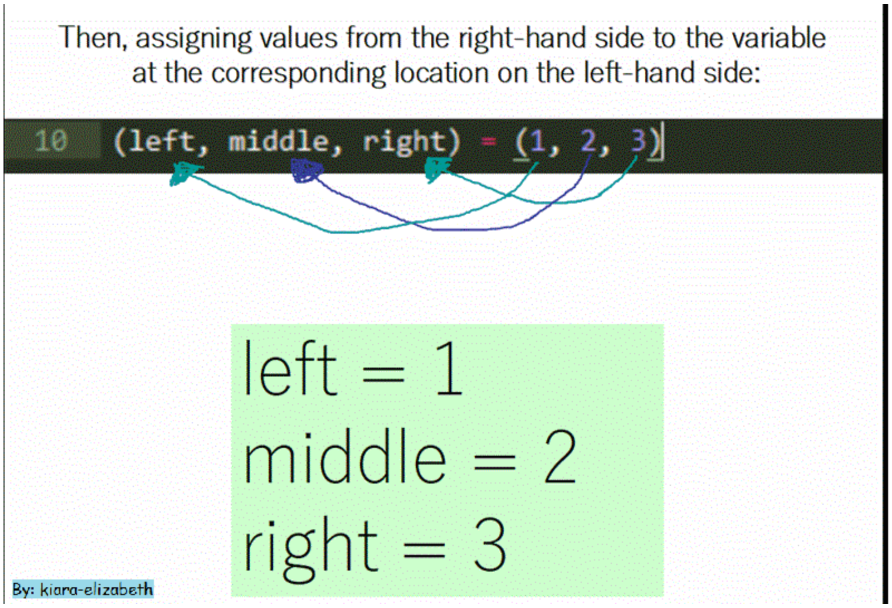
So...

```
(left, middle, right) = returnATuple(1, 2, 3)
```

Is equivalent to this after the function returned the tuple...

```
(left, middle, right) = (1, 2, 3)
```

Now, finally we assign the values to the variables, each value on the right hand side tuple is assigned to the variable located at its corresponding position on the left hand side tuple. The first value is assigned to the first variable and so on.



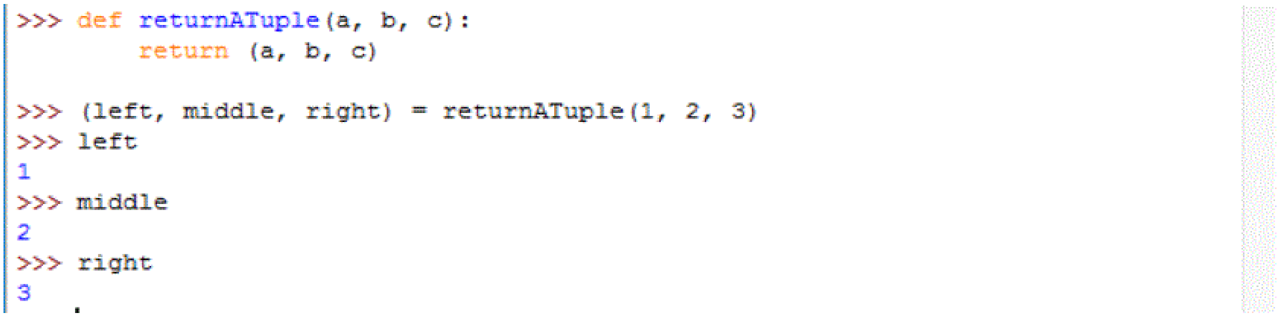
Then, assigning values from the right-hand side to the variable at the corresponding location on the left-hand side:

```
(left, middle, right) = (1, 2, 3)
```

left = 1
middle = 2
right = 3

By: kiara-elizabeth

Here you can see this process executed in the shell:



```
>>> def returnATuple(a, b, c):
        return (a, b, c)

>>> (left, middle, right) = returnATuple(1, 2, 3)
>>> left
1
>>> middle
2
>>> right
3
```

Hope it helps! If you have any questions or doubts, please ask them on the forums!

Estefania (kiara-elizabeth).