

Tuple in Python

- A Tuple is a collection of Python objects separated by commas.
- A tuple in Python is similar to a list. The difference between the two is that we cannot change the elements of a tuple once it is assigned whereas, in a list, elements can be changed i.e. a tuple is immutable unlike lists which are mutable.

Creating a Tuple

- A tuple is created by placing all the items (elements) inside parentheses (), separated by commas. The parentheses are optional, however, it is a good practice to use them.
- A tuple can have any number of items and they may be of different types (integer, float, list, string, etc.).

Creating a Tuple

- A tuple can also be created without using parentheses. This is known as **tuple packing**.
- **Packing and Unpacking a Tuple** : In Python there is a very powerful tuple assignment feature that assigns **right hand side of values into left hand side**. In other way it is called **unpacking of a tuple of values into a variable**. In **packing**, we put values into a new tuple while in **unpacking** we extract those values into a single variable.
- In **unpacking of tuple** number of variables on left hand side should be equal to number of values in given tuple.

Creating a tuple with one element

- Having one element within parentheses is not enough. **We will need a trailing comma** to indicate that it is, in fact, a tuple.

Access Tuple Elements

- We can use the **index operator []** to access an item in a tuple where the index starts from 0.
- So, a tuple having 6 elements will have indices from 0 to 5.
- Python allows **negative indexing** for its sequences.
- The index of -1 refers to the last item, -2 to the second last item and so on.

Slicing

- We can access a range of items in a tuple by using the slicing operator - colon ":".

```
my_tuple = ('p','r','o','g','r','a','m','i','z')
```

```
# elements 2nd to 4th
```

```
#Output: ('r', 'o', 'g')
```

```
print(my_tuple[1:4])
```

Slicing

- Slicing can be best visualized by considering the index to be between the elements as shown below. So if we want to access a range, we need the index that will slice the portion from the tuple.

P	R	O	G	R	A	M	I	Z	
0	1	2	3	4	5	6	7	8	9
-9	-8	-7	-6	-5	-4	-3	-2	-1	

Changing a Tuple

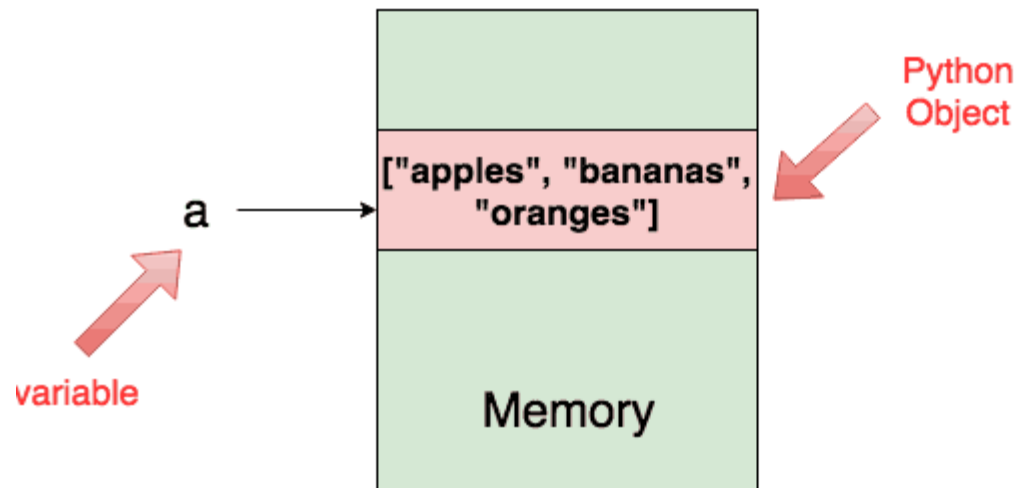
- Unlike lists, tuples are immutable.
- This means that elements of a tuple cannot be changed once it has been assigned. But, if the element is itself a mutable datatype like list, its nested items can be changed.
- We can also assign a tuple to different values (reassignment).

Mutable vs Immutable i.e. (List vs Tuple)

- a variable is nothing but a reference to the actual python object in memory.
- The variable itself is not the object.

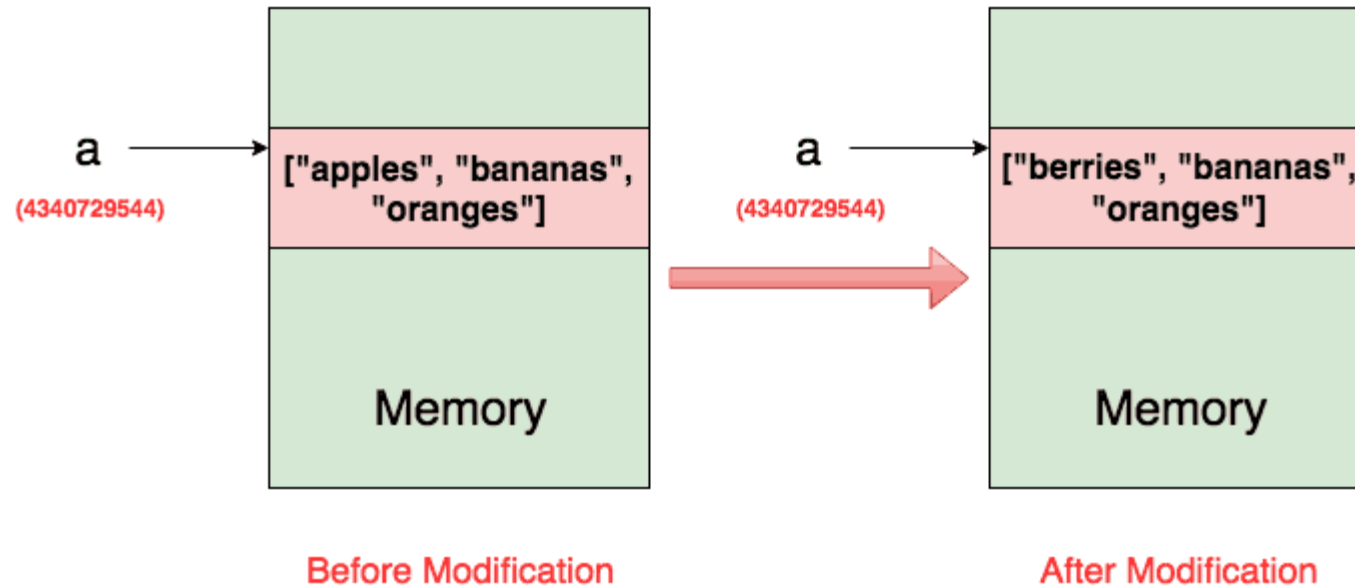
Variable and Python object

- `a = ["apples", "bananas", "oranges"]`

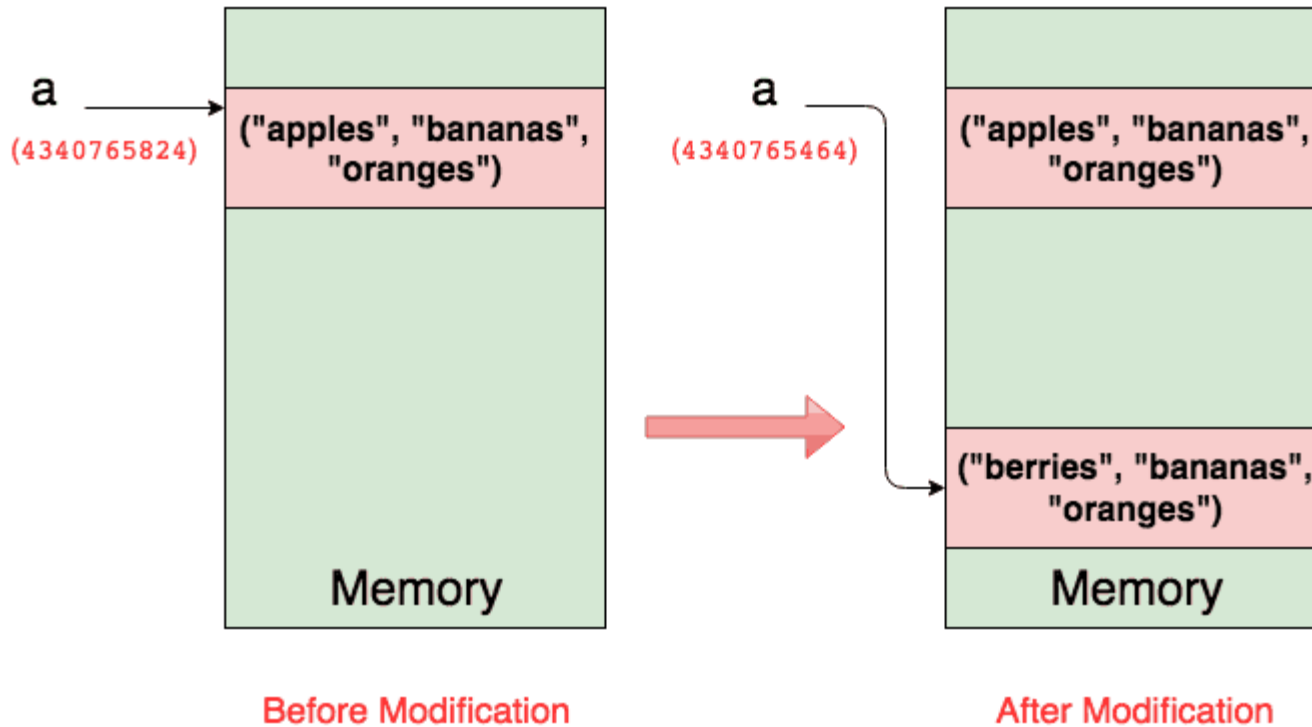


- When you do this, a python object of type list is created in the memory and the variable `a` refers to this object by holding its location in memory.

In case of List



In Case of Tuple



- Variables don't have types instead they are just labels in Python. It is the value which gets associated with a type. Hence, the same variable, the label can refer values of different Python data types.