

# Python Tuples

If this Python Tutorial saves you  
hours of work, please **whitelist it in**  
**your ad blocker** 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web  
hosting fee and CDN to keep the

website running.

**Summary:** in this tutorial, you'll learn about Python tuples and how to use them effectively.

## Introduction to Python tuples

Sometimes, you want to create a [list](https://www.pythontutorial.net/python-basics/python-list/) of items that cannot be changed throughout the program. Tuples allow you to do that.

A tuple is a list that cannot change. Python refers to a value that cannot change as **immutable** (<https://www.pythontutorial.net/advanced-python/python-mutable-and-immutable/>). So by definition, **a tuple is an immutable list.**

## Defining a tuple

A tuple is like a list except that it uses parentheses `()` instead of square brackets `[]`.

The following example defines a tuple called `rgb`:

```
rgb = ('red', 'green', 'blue')
```

Once you define a tuple, you can access an individual element by its index, just. For example:

```
rgb = ('red', 'green', 'blue')

print(rgb[0])
print(rgb[1])
print(rgb[2])
```

Output:

```
red
green
blue
```

Since a tuple is immutable, you cannot change its elements. The following example attempts to change the first element of the `rgb` tuple to `'yellow'` :

```
rgb = ('red', 'green', 'blue')
rgb[0] = 'yellow'
```

And it results in an error:

```
TypeError: 'tuple' object does not support item assignment
```

## Defining a tuple that has one element

To define a tuple with one element, you need to include a trailing comma after the first element. For example:

```
numbers = (3,)
print(type(numbers))
```

Output:

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

If you exclude the trailing comma, the type of the `numbers` will be `int` , which stands for `integer` (<https://www.pythontutorial.net/advanced-python/python-integers/>) . And its value is 3. Python won't create a tuple that includes the number 3:

```
numbers = (3)
print(type(numbers))
```

Output:

```
<class 'int'>
```

## Assigning a tuple

Even though you can't change a tuple, you can assign a new tuple to a `variable` (<https://www.pythontutorial.net/python-basics/python-variables/>) that `references` (<https://www.pythontutorial.net/advanced-python/python-references/>) a tuple. For example:

```
colors = ('red', 'green', 'blue')
print(colors)

colors = ('Cyan', 'Magenta', 'Yellow', 'black')
print(colors)
```

## Summary

- Tuples are immutable lists.
- Use tuples when you want to define a list that cannot change.