

3.2.6 Data Structures: Tuples

Storing and accessing data in lists can be performed in different ways depending on what data you want to store or retrieve. Now Tom and Seth think you can move on to learning how to manipulate and use tuples. Like lists, tuples can be used to store data; however, you can only retrieve data from tuples. Tom is going to walk you through how to retrieve data from tuples.

Tuples are similar to lists in Python, with a major exception: once you create a tuple, it cannot be changed. This means tuples are **immutable**: we can't add or remove items from them.

IMPORTANT

You can create an empty tuple with the following syntax:

```
my_tuple = ( )
```

Alternatively, you can use the built-in `tuple()` method:

```
my_tuple = tuple()
```

Here's a tuple that contains a list of counties:

```
counties_tuple = ("Arapahoe", "Denver", "Jefferson")
```

Like lists, we can find the length of tuples as well as find items in the tuple by index.

To get the length of a tuple, use the same format for getting the length of a list: `len(counties_tuple)`.

```
>>> len(counties_tuple)
3
```

To get an item from a tuple, we can apply indexing and slicing, using square brackets after the tuple, just like with lists.

In the Python interpreter, type `counties_tuple[1]` and press Enter. The output is `Denver`.

```
>>> counties_tuple[1]
'Denver'
```

 [Retake](#)

NOTE

For more information, see the [documentation on built-in Python functions like tuple\(\)](https://docs.python.org/3.7/library/functions.html) [_\(https://docs.python.org/3.7/library/functions.html\)_](https://docs.python.org/3.7/library/functions.html).