We create a tuple by placing items inside Parentheses ().

numbers = (1, 2, -5)

print(numbers)

# Output: (1, 2, -5)

Create a Tuple Using tuple() Constructor

tuple\_constructor = tuple(('Jack', 'Maria', 'David'))

print(tuple\_constructor)

# Output: ('Jack', 'Maria', 'David')

Different Types of Python Tuples

Here are the different types of tuples we can create in Python.

**Empty Tuple**

# create an empty tuple

empty\_tuple = ()

print(empty\_tuple)

# Output: ()

**Tuple of different data types**

# tuple of string types

names = ('James', 'Jack', 'Eva')

print (names)

# tuple of float types

float\_values = (1.2, 3.4, 2.1)

print(float\_values)

**Tuple of mixed data types**

# tuple including string and integer

mixed\_tuple = (2, 'Hello', 'Python')

print(mixed\_tuple)

# Output: (2, 'Hello', 'Python')

**Tuple Characteristics**

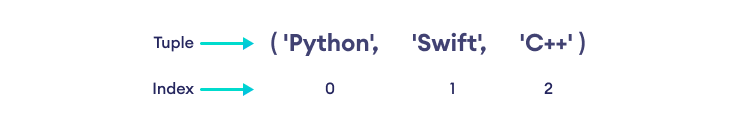
Tuples are:

* **Ordered** - They maintain the order of elements.
* **Immutable** - They cannot be changed after creation.
* **Allow duplicates** - They can contain duplicate values.

## Access Tuple Items

Each item in a tuple is associated with a number, known as a **index**.

The index always starts from **0**, meaning the first item of a tuple is at index **0**, the second item is at index **1,** and so on.

Index of Tuple Item: Access Items Using Index

We use index numbers to access tuple items. For example,

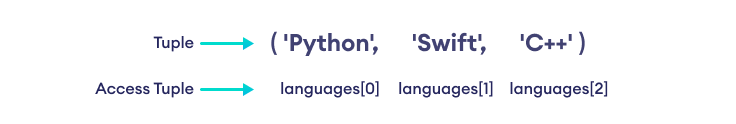
languages = ('Python', 'Swift', 'C++')

# access the first item

print(languages[0]) # Python

# access the third item

print(languages[2]) # C++



## Tuple Cannot be modified

Python tuples are immutable (unchangeable). We cannot add, change, or delete items of a tuple.

If we try to modify a tuple, we will get an error. For example,

cars = ('BMW', 'Tesla', 'Ford', 'Toyota')

# trying to modify a tuple

cars[0] = 'Nissan' # error

print(cars)

## Python Tuple Length

We use the [len()](https://www.programiz.com/python-programming/methods/built-in/len) function to find the number of items present in a tuple. For example,

cars = ('BMW', 'Tesla', 'Ford', 'Toyota')

print('Total Items:', len(cars))

# Output: Total Items: 4

## Iterate Through a Tuple

We use the [for loop](https://www.programiz.com/python-programming/for-loop) to iterate over the items of a tuple. For example,

fruits = ('apple','banana','orange')

# iterate through the tuple

for fruit in fruits:

print(fruit)

**Output**

apple

banana

orange

Check if an Item Exists in the Tuple

We use the in keyword to check if an item exists in the tuple. For example,

colors = ('red', 'orange', 'blue')

print('yellow' in colors) # False

print('red' in colors) # True

Delete Tuples

We cannot delete individual items of a tuple. However, we can delete the tuple itself using the [del](https://www.programiz.com/python-programming/del) statement.

# Python Tuple count()

The count() method returns the number of times the specified element appears in the [tuple](https://www.programiz.com/python-programming/tuple).

## count() Syntax

The syntax of the count() method is:

vowels.count('i')

**count() Parameter**

The count() method takes a single parameter:

* 'i' - an element to count in the tuple

### Example

# tuple of vowels

vowels = ('a', 'e', 'i', 'o', 'i', 'u')

# counts the number of i's in the tuple

count = vowels.count('i')

print(count)

# Output: 2

## Example 1: Python Tuple count()

# tuple of numbers

numbers = (1, 3, 4, 1, 6 ,1 )

# counts the number of 1's in the tuple

count = numbers.count(1)

print('The count of 1 is:', count)

# counts the number of 7's in the tuple

count = numbers.count(7)

print('The count of 7 is:', count)

**Output**

The count of 1 is: 3

The count of 7 is: 0

## Example 2: count() to Count List and Tuple Elements Inside Tuple

# tuple containing list and tuples

random = ('a', ('a', 'b'), ('a', 'b'), [3, 4])

# count element ('a', 'b')

count = random.count(('a', 'b'))

print("The count of tuple ('a', 'b') is:", count)

# count element [3, 4]

count = random.count([3, 4])

print("The count of list [3, 4] is:", count)

**Output**

The count of tuple ('a', 'b') is: 2

The count of list [3, 4] is: 1

# Python Tuple index()

The index() method returns the index of the specified element in the [tuple](https://www.programiz.com/python-programming/tuple).

### Example

# tuple containing vowels

vowels = ('a', 'e', 'i', 'o', 'u')

# index of 'e' in vowels

index = vowels.index('e')

print(index)

# Output: 1

## index() Syntax

The syntax of the index() method is:

tuple.index(element, start\_index, end\_index)

Here, the index() scans the element in the tuple from start\_index to end\_index.

## index() Parameter

The index() method can take one to three parameters:

* element - the item to scan
* start\_index (optional) - start scanning the element from the start\_index
* end\_index (optional) - stop scanning the element at the end\_index

## index() Return Value

The index() method returns:

* the index of the given element in the tuple
* ValueError [exception](https://www.programiz.com/python-programming/exceptions) if the element is not found in the tuple

**Note:** The index() method only returns the first occurrence of the matching element.

## Example 1: Python Tuple index()

# tuple containing vowels

vowels = ('a', 'e', 'i', 'o', 'i', 'u')

# index of 'e' in vowels

index = vowels.index('e')

print('Index of e:', index)

# index of the first 'i' is returned

index = vowels.index('i')

print('Index of i:', index)

**Output**

Index of e: 1

Index of i: 2

## Example 2: index() throws an error if the specified element is absent in the Tuple

# tuple containing numbers

numbers = (0, 2, 4, 6, 8, 10)

# throws error since 3 is absent in the tuple

index = numbers.index(3)

print('Index of 3:', index)

**Output**

ValueError: tuple.index(x): x not in tuple

## Example 3: index() With Start and End Parameters

# alphabets tuple

alphabets = ('a', 'e', 'i', 'o', 'g', 'l', 'i', 'u')

# returns the index of first 'i' in alphabets

index = alphabets.index('i')

print('Index of i in alphabets:', index)

# scans 'i' from index 4 to 7 and returns its index

index = alphabets.index('i', 4, 7)

print('Index of i in alphabets from index 4 to 7:', index)

**Output**

Index of i in alphabets: 2

Index of i in alphabets from index 4 to 7: 6