Tuples in Python

Python Tuple is a collection of objects separated by commas. In some ways, a tuple is similar to a Python list in terms of indexing, nested objects, and repetition but the main difference between both is Python tuple is immutable, unlike the Python list which is mutable.

Create a Python Tuple

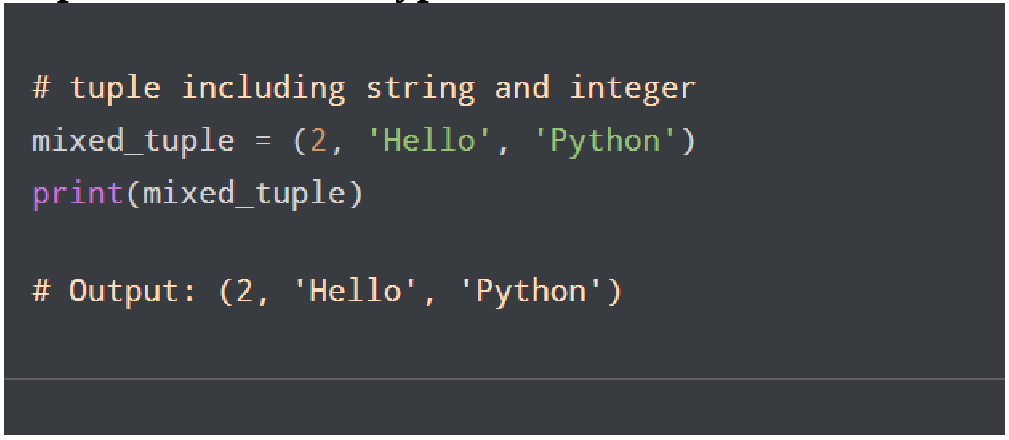
We create a tuple by placing items inside parentheses (). For example,

```
numbers = (1, 2, -5)
print(numbers)

# Output: (1, 2, -5)
```

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

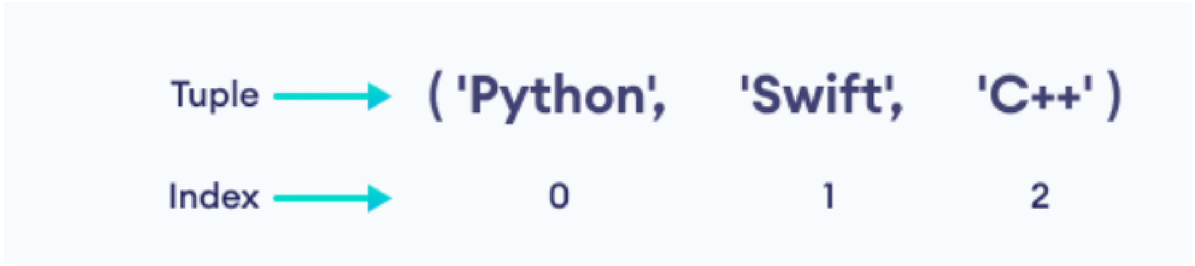
Tuples are:

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

10000 C{}DERS Right path for a Bright Career.

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.



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.

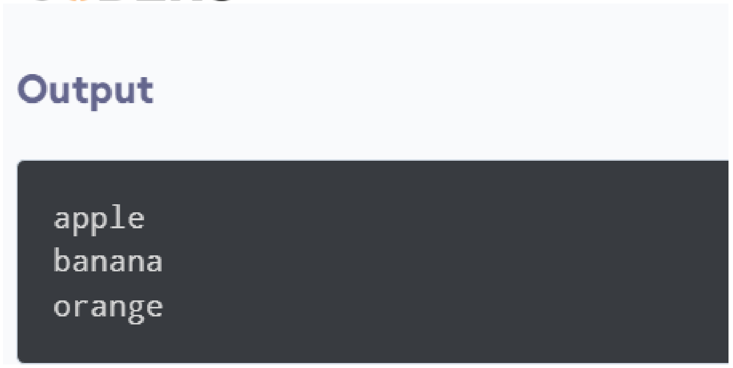
If we try to modify a tuple, we will get an error. For example,

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

# trying to modify a tuple
cars[0] = 'Nissan'    # error

print(cars)
```

Iterate Through a Tuple

We use the 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

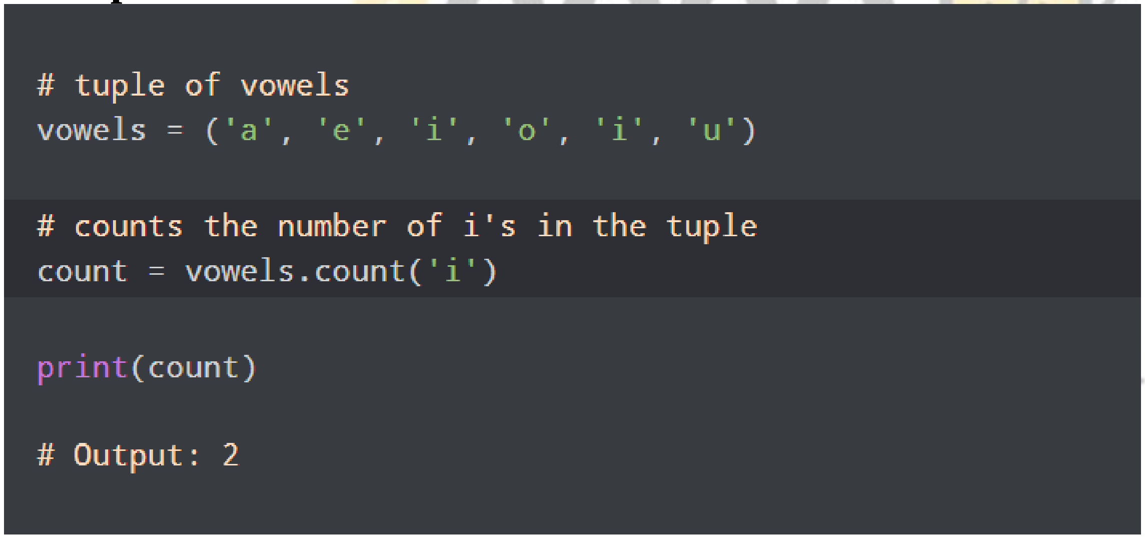
```
apple
banana
orange
```

Python Tuple Methods

Python Tuple count()

The count() method returns the number of times the specified element appears in the tuple.

Example



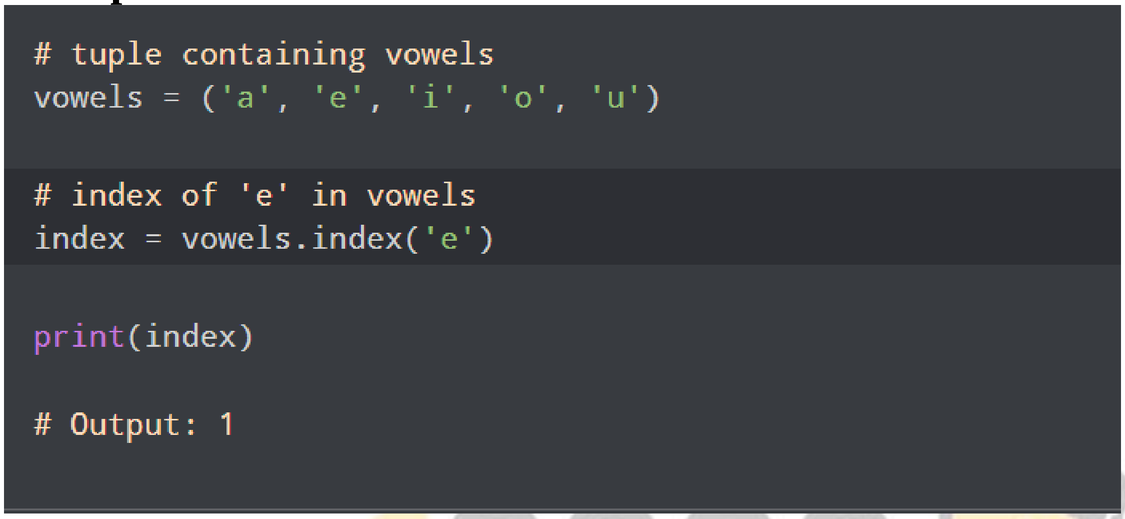
```python
# tuple of vowels
vowels = ('a', 'e', 'i', 'o', 'i', 'u')

# counts the number of i's in the tuple
count = vowels.count('i')

print(count)

# Output: 2
```

+91 6305693431, 9154641231, 8790732332
MIG 214, 2ND FLOOR, KPHB KUKATPALLY, HYDERABAD, 500072 | SRA EDUTECH PVT LTD

Python Tuple index()

The index() method returns the index of the specified element in the tuple.

Example



```
# tuple containing vowels
vowels = ('a', 'e', 'i', 'o', 'u')

# index of 'e' in vowels
index = vowels.index('e')

print(index)

# Output: 1
```

10000 C{}DERS Right path for a Bright Career.

Nested Tuples in Python

Nested tuples are tuples that contain other tuples as their elements. They can be thought of as multi-dimensional tuples, where each element of the main tuple can itself be a tuple. This structure allows you to represent complex data hierarchies, such as matrices, grids, or any other nested data format.

Characteristics of Nested Tuples:

1. Immutability: Like regular tuples, nested tuples are immutable, meaning that once a tuple is created, its elements (including those in nested tuples) cannot be changed.

2. Accessing Elements: You can access elements in a nested tuple using multiple indices. The first index accesses the main tuple, and the subsequent indices access elements within the nested tuple.

3. Fixed Structure: Nested tuples have a fixed structure, so you cannot change their size or content after they are created. However, you can create new tuples based on existing ones.

Example of a Nested Tuple:

```python
# A nested tuple representing a 3x3 matrix
nested_tuple = (
    (1, 2, 3),
    (4, 5, 6),
    (7, 8, 9)
)
```

In this example, nested_tuple is a tuple that contains three inner tuples, each representing a row of a 3x3 matrix.

Accessing Elements in Nested Tuples:

To access elements in a nested tuple, you use multiple levels of indexing. For example:

```python
# Access the element at the first row, first column
print(nested_tuple[0][0])  # Output: 1

# Access the element at the second row, second column
print(nested_tuple[1][1])  # Output: 5
```

Right path for a Bright Career.