Tuples

Day 8 - Python Basics

Shaida Muhammad

Wednesday, March 12, 2025

ShaidaSherpao@gmail.com

Agenda

1 What are tuples?

2 Creating and accessing tuples

3 Immutability of tuples

4 Comparing tuples with lists

5 Hands-on practice

What are Tuples?

- **Creating a Tuple:**

```
my_tuple = (1, 2, 3, "apple", True)
```

- **Accessing Items:**
  - Use indexing (`tuple[index]`).
  - Indexing starts from `0`.

```
fruits = ("apple", "banana", "cherry")
print(fruits[0])  # Output: apple
print(fruits[1])  # Output: banana
```

- **Negative Indexing:**

```
print(fruits[-1])  # Output: cherry
```

- **Slicing: Extracting a part of the tuple**

```
numbers = (1, 2, 3, 4, 5)
print(numbers[1:3])  # Output: (2, 3)
print(numbers[:3])   # Output: (1, 2, 3)
print(numbers[2:])   # Output: (3, 4, 5)
```

Creating and Accessing Tuples

- **Creating a Tuple:**

```
my_tuple = (1, 2, 3, "apple", True)
```

- **Accessing Items:**
  - Use indexing (`tuple[index]`).
  - Indexing starts from `0`.

```
fruits = ("apple", "banana", "cherry")
print(fruits[0])  # Output: apple
print(fruits[1])  # Output: banana
```

- **Negative Indexing:**

```
print(fruits[-1])  # Output: cherry
```

Immutability of Tuples

- **Definition:** Tuples are immutable, meaning their contents cannot be changed after creation.
- **Example:**

```
fruits = ("apple", "banana", "cherry")
fruits[0] = "mango"  # This will raise an error
```

- **Why use tuples?**
  - Protect data from being modified.
  - Faster than lists for fixed data.

Important Tuple Methods

- **count():** Returns the number of times a value appears in the tuple.

```
numbers = (1, 2, 3, 2, 4, 2)
print(numbers.count(2))  # Output: 3
```

- **index():** Returns the index of the first occurrence of a value.

```
fruits = ("apple", "banana", "cherry")
print(fruits.index("banana"))  # Output: 1
```

For Loop with Tuples

- **Simple For Loop:**

```
fruits = ("apple", "banana", "cherry")

for fruit in fruits:
    print(fruit)
```

- **Use enumerate() to track loop iteration.**

```
fruits = ("apple", "banana", "cherry")

for index, fruit in enumerate(fruits):
    print(f"Index {index}: {fruit}")
```

Comparing Tuples with Lists

- **Similarities:**
  - Both are ordered collections.
  - Both can contain different data types.
- **Differences:**
  - Lists are mutable; tuples are immutable.
  - Lists use `[ ]`; tuples use `( )`.
- **Example:**

```
my_list = [1, 2, 3]
my_tuple = (1, 2, 3)
```

Hands-On Practice

- **Task 1:** Create a tuple of your favorite fruits and print each fruit.

```
fruits = ("apple", "banana", "cherry")
for fruit in fruits:
    print(fruit)
```

- **Task 2:** Access the second item in the tuple and print it.

```
print(fruits[1])  # Output: banana
```

- **Task 3:** Try to change an item in the tuple and observe the error.

```
fruits[0] = "mango"  # This will raise an error
```

Task 4: Compare a tuple and a list by creating both and printing them.

```
my_list = [1, 2, 3]
my_tuple = (1, 2, 3)
print(my_list)
print(my_tuple)
```

Recap

- Tuples are ordered, immutable collections of items.
- Use indexing to access items.
- Tuples are faster and safer for fixed data compared to lists.

Homework

1. Create a tuple of your favorite movies and print each movie.
2. Access the third movie in the tuple and print it.
3. Compare a tuple and a list by creating both and printing them.

Q&A

- Do you have any questions?
- Share your thoughts.

Closing

Next class: Dictionaries