Python - Lists and Tuples

Shahriar's
Intelligence
Academy

Tuples

Definition:

A **tuple** is an **immutable sequence in Python** used to **store multiple items in a single variable**. Items in a tuple are ordered and can be of different data types (e.g., integers, strings, floats).

Characteristics:

- Immutable and ordered
- Allows Duplicates
- Defined with Parentheses

Example:

```python
# Creating a tuple
fruits = ("apple", "banana", "cherry")
print(fruits)  # Output: ('apple', 'banana', 'cherry')
```

Tuple Concatenation

Key Features:

- Tuple concatenation is the process of **combining two or more tuples** to create a new tuple.
- This is achieved using the **+ operator**.

Example:

```
# Tuple concatenation
tuple1 = (1, 2, 3)
tuple2 = (4, 5, 6)
result = tuple1 + tuple2
print(result)  # Output: (1, 2, 3, 4, 5, 6)
```

Tuple Slicing

Definition:

Tuple slicing **extracts a portion of a tuple** using a specific range of indices.

Syntax:

tuple[start **:** end **:** step**]**

start: Starting index (inclusive)
end: Ending index (exclusive)
step: Interval between indices

Example:

```
# Slicing a tuple
numbers = (10, 20, 30, 40, 50, 60)
slice_result = numbers[1:4]  # Extracts elements from index 1 to 3
print(slice_result)  # Output: (20, 30, 40)
```

Nested Tuple Slicing

Definition:

- A nested tuple is a tuple that **contains other tuples** as its elements.

Example:

```python
# Nested tuple
nested = (("apple", "banana"), (10, 20, 30), ("x", "y", "z"))
print(nested[1])  # Output: (10, 20, 30)
```

```python
# Slicing a nested tuple
nested = (("apple", "banana"), (10, 20, 30), ("x", "y", "z"))
slice_result = nested[1][0:2]  # Access first two elements of the second tuple
print(slice_result)  # Output: (10, 20)
```

List

Definition:

A **list** is a **mutable sequence in Python** used to **store multiple items** in a single variable. Like tuples, lists can store elements of different data types.

Characteristics:

- Mutable and ordered
- Allows Duplicates
- Defined with Square brackets

Example:

```python
# Creating a list
numbers = [1, 2, 3, 4]
print(numbers)  # Output: [1, 2, 3, 4]
```

List Forward Slicing

Key Features:

- Extracts a portion of a list by traversing from **left to right**.
- Positions start counting from the beginning **with index value of 0**.

Example:

```
# Forward slicing
colors = ["red", "green", "blue", "yellow"]
forward_slice = colors[1:3]  # Extracts elements from index 1 to 2
print(forward_slice)  # Output: ['green', 'blue']
```

List Backward Slicing

Key Features:

- Extracts a portion of a list by traversing from **right to left** (using negative indices).
- Positions start counting from the last **with index value of -1.**

Example:

```
# Backward slicing
backward_slice = colors[-3:-1]  # Extracts second and third elements from the end
print(backward_slice)  # Output: ['green', 'blue']
```

List Striding

Key Features:

- Allows extracting a range of elements from a list **skipping specific values**.
- The syntax: **list[**start **:** end **:** step**]**

Example:

```
# Range slicing
numbers = [10, 20, 30, 40, 50, 60]
range_slice = numbers[0:6:2]  # Extract every second element from the list
print(range_slice)  # Output: [10, 30, 50]
```

Nested List Slicing

Definition:

- A nested list **contains other lists** as elements.

Example:

```
# Nested list
nested = [["apple", "banana"], [1, 2, 3], ["x", "y", "z"]]
print(nested[2])  # Output: ['x', 'y', 'z']
```

```
# Slicing a nested list
nested = [["apple", "banana"], [1, 2, 3], ["x", "y", "z"]]
slice_result = nested[1][0:2]  # Extract first two elements from the second list
print(slice_result)  # Output: [1, 2]
```

List Item Add/Remove

Key Features:

- Use methods like **append() or insert()** to **add elements** to a list.
- Use methods like **remove() or pop()** to **delete elements** from a list.

Example:

```
# Adding and removing values in a list
numbers = [10, 20, 30]
numbers.append(40)  # Add 40 to the end
print(numbers)  # Output: [10, 20, 30, 40]
```

```
numbers.remove(20)  # Remove 20
print(numbers)  # Output: [10, 30, 40]
```

Thank You!

Shahriar's
Intelligence
Academy