Chapter 04 Lists and Tuples

Python Notes by Wahab

Lists

Definition

- A list is a mutable **(can changed after created)**, ordered collection of items.
- Lists can contain items of different data types.

Creating Lists

```python
# Creating a list with different data types
my_list = [1, 2.5, "Hello", True]
```

Accessing Elements

```python
# Accessing the first element
first_element = my_list[0]

# Accessing elements from the end
last_element = my_list[-1]
```

List Methods

append()

- **Description**: Adds an element to the end of the list.
- **Syntax**: `list.append(element)`

extend()

- **Description**: Extends the list by appending elements from another iterable.
- **Syntax**: `list.extend(iterable)`

insert()

- **Description**: Inserts an element at a specific position.
- **Syntax**: `list.insert(index, element)`

remove()

- **Description**: Removes the first occurrence of a specified value.
- **Syntax**: `list.remove(element)`

pop()

- **Description**: Removes and returns an element at a given position (default is the last item).
- **Syntax**: `list.pop([index])`

sort()

- **Description**: Sorts the list in ascending order (or descending if specified).
- **Syntax**: `list.sort(key=None, reverse=False)`

reverse()

- **Description**: Reverses the order of the list in place.
- **Syntax**: `list.reverse()`

copy()

- **Description**: Returns a shallow copy of the list.
- **Syntax**: `list.copy()`

List Slicing

A method to extract a portion of a list by specifying a `start`, `stop`, and `step` index, creating a new list from the specified range of elements.

Syntax

- **`list[start:stop:step]`**

Example

```
    my_list = [0, 1, 2, 3, 4, 5]
    sub_list = my_list[1:4]
    # Result: [1, 2, 3]
```

#Tuples

Definition

- **Tuple**: An immutable **(cannot changed once created)**, ordered collection of items, which can contain elements of different data types.

Creating Tuples

- **Syntax**: `tuple = (item1, item2, item3, ...)`

```
my_tuple = (1, 2.5, "Hello", True)
```

Accessing Elements

- **Syntax**: `tuple[index]`

```
    first_element = my_tuple[0]
    last_element = my_tuple[-1]
```

Tuple Methods

- `count(value)`: Returns the number of occurrences of a value.

```
    count_value = my_tuple.count(1)
```

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

```
   index_value = my_tuple.index("Hello")
```

Tuple Packing and Unpacking

- **Packing**: Assigning multiple values to a tuple.

```
    packed_tuple = 1, 2.5, "Hello"
```

- **Unpacking**: Assigning values from a tuple to variables.

```
    a, b, c = packed_tuple
```

Differences from Lists

- **Mutability**: Tuples are immutable; lists are mutable.
- **Syntax**: Tuples use parentheses `()`, while lists use square brackets `[]`.
- **Performance**: Tuples are generally faster and require less memory than lists.