Functions That Work with Tuples

---

1. `len()` → Returns the number of items

```
fruits = ("apple", "banana", "mango")
print(len(fruits))
```

Output:

```
3
```

---

2. `count(value)` → Counts how many times a value appears

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

Output:

```
3
```

---

3. `index(value)` → Finds the first position of the value

```
colors = ("red", "blue", "green", "blue")
print(colors.index("blue"))
```

Output:

```
1
```

⚠️ If the value is not found, it gives an error.

---

4. `sum()` → Adds all values (numbers only)

```python
marks = (80, 75, 60)
print(sum(marks))
```

Output:

```
215
```

---

5. `max()` and `min()` → Get the biggest/smallest value

```python
nums = (10, 50, 20, 5)
print(max(nums))   # 50
print(min(nums))   # 5
```

Output:

```
50
5
```

⚠️ Works only with **numbers** or **comparable items** (like all strings).

---

6. `sorted()` → Returns a sorted list version of the tuple

```python
t = (30, 10, 20)
sorted_list = sorted(t)
print(sorted_list)
```

Output:

```
[10, 20, 30]
```

Note: `sorted()` doesn't change the tuple. It returns a **new list**.

What is a Dictionary?

A **dictionary** in Python stores data in **key-value** format.

You use a **key** to access the **value** — just like a real dictionary where you search a word (key) to get its meaning (value).

Example:

```
student = {
    "name": "Anjali",
    "age": 20,
    "course": "Python"
}
```

Here:

- `"name"` is a key → `"Anjali"` is the value

- `"age"` is a key → `20` is the value

- `"course"` is a key → `"Python"` is the value

---

2. Creating a Dictionary

```
info = {"brand": "Nike", "price": 2999}
print(info)
```

Output:

```
{'brand': 'Nike', 'price': 2999}
```

---

3. Accessing Values

```
print(info["brand"])        # Nike
```

Using .get() (avoids error if key doesn't exist):

```
print(info.get("color", "Not found"))
```

Output:

```
Not found
```

---

4. Changing & Adding Items

Change value:

```
info["price"] = 2499
```

Add new key-value pair:

```
info["category"] = "Shoes"
```

5. Removing Items

```
info.pop("price")      # Removes 'price'
del info["brand"]      # Removes 'brand'
info.clear()           # Empties the dictionary
```

6. Looping Through Dictionary

```
student = {"name": "Amit", "age": 19}

for key, value in student.items():
    print(key, "→", value)
```

Output:

```
name → Amit
age → 19
```

7. Useful Dictionary Functions

Function	Description
.keys()	Returns all keys
.values()	Returns all values
.items()	Returns key-value pairs (tuple format)

`.get(key)`	Safer way to access a value
`.update()`	Adds or updates multiple key-values
`.pop(key)`	Removes the key-value pair
`.clear()`	Empties the dictionary

---

1. `get()` – To Access Values

```python
student = {"name": "Anjali", "age": 20}
print(student.get("name"))        # Anjali
print(student.get("email", "N/A"))  # N/A (default)
```

Use this instead of `dict["key"]` to **avoid errors** if the key is missing.

---

2. `keys()` – Returns All Keys

```python
print(student.keys())
```

Output:

```python
dict_keys(['name', 'age'])
```

Use in loops:

```python
for key in student.keys():
    print(key)
```

3. `values()` – Returns All Values

```
print(student.values())
```

Output:

```
dict_values(['Anjali', 20])
```

---

4. `items()` – Returns All Key-Value Pairs

```
for k, v in student.items():
    print(k, "→", v)
```

Output:

```
name → Anjali
age → 20
```

---

5. `update()` – Add or Modify Key-Value Pairs

```
student.update({"course": "Python", "city": "Delhi"})
print(student)
```

Output:

```
{'name': 'Anjali', 'age': 20, 'course': 'Python', 'city': 'Delhi'}
```

---

6. `pop(key)` – Remove Item by Key

```
student.pop("age")
print(student)
```

Output:

```
{'name': 'Anjali', 'course': 'Python', 'city': 'Delhi'}
```

> ⚠️ Gives an error if the key doesn't exist (unless you give a default value).

---

7. `clear()` – Removes All Items

```
student.clear()
print(student)
```

Output:

CopyEdit
```
{}
```

---

8. `copy()` – Makes a Copy of Dictionary

```
original = {"a": 1, "b": 2}
duplicate = original.copy()
print(duplicate)
```

> 🔒 Safe copy — changes to `duplicate` won't affect `original`.

Practice Questions

1. Create a dictionary of a student with name, age, course

2. Print the course using the key

3. Add a new key `"marks"` with value 95

4. Change the student's name

5. Use `.get()` to access `"email"` key — handle the missing case

6. Loop through the dictionary and print all key-value pairs

7. Create a dictionary of 3 countries with their capitals

8. Use `.update()` to add `"city": "Delhi"` and `"college": "XYZ"`

9. Remove `"age"` key from the student dictionary

10. Use `.keys()` and `.values()` to print them separately

1. What is a Set?

A **set** in Python is a collection of **unordered** and **unique** values.

Key Points:

- ✅ **Duplicates are not allowed**

- ✅ **Items are unordered** (no index)

- ✅ **Set is changeable** (you can add/remove items, but can't access using index)

---

2. Creating a Set

```python
fruits = {"apple", "banana", "mango"}
print(fruits)
```

Output:

```
{'apple', 'banana', 'mango'}
```

---

3. Duplicates Are Automatically Removed

```python
nums = {1, 2, 2, 3, 4, 4, 4}
print(nums)
```

Output:

```
{1, 2, 3, 4}
```

---

4. Set Functions and Methods

✅ `add(item)` – **Adds one item**

```python
colors = {"red", "blue"}
colors.add("green")
```

✅ **update([list])** – **Adds multiple items**

```
colors.update(["yellow", "black"])
```

✅ **remove(item)** – **Removes item (error if not found)**

```
colors.remove("red")
```

✅ **discard(item)** – **Removes item (no error if not found)**

```
colors.discard("orange")  # Safe
```

✅ **pop()** – **Removes a random item**

```
colors.pop()
```

✅ **clear()** – **Removes all items**

```
colors.clear()
```

✅ **copy()** – **Copies the set**

```
copy_colors = colors.copy()
```

---

5. Sets are Unordered

```python
items = {"apple", "banana", "mango"}
print(items[0])  # ❌ Error
```

Output:

```
TypeError: 'set' object is not subscriptable
```

Practice Questions

1. Create a set of your 5 favorite movies

2. Add one more movie using `add()`

3. Add 3 more movies using `update()`

4. Remove a movie using `discard()`