Python Online Free Ramzan Course 2025
Taught by: Shaida Muhammad

Dictionaries

Day 9 - Python Basics

Shaida Muhammad

Thursday, March 13, 2025

ShaidaSherpao@gmail.com

Agenda

1. What are dictionaries?
2. Creating and accessing dictionaries
3. Adding, updating, and deleting items
4. Dictionary methods (keys(), values(), items())
5. Looping through dictionaries
6. Hands-on practice

What are Dictionaries?

- **Definition:** A dictionary is a collection of key-value pairs.
- **Features:**
  - Unordered: Items are not stored in a specific order.
  - Mutable: Items can be added, removed, or changed.
  - Keys must be unique and immutable (e.g., strings, numbers).
- **Example:**

```
student = {"name": "Ali", "age": 17, "grade": "A"}
```

Creating and Accessing Dictionaries

- **Creating a Dictionary:**

```
my_dict = {"key1": "value1", "key2": "value2"}
```

- **Accessing Items:**
  - Use the key to access the value (`dict[key]`).

```
student = {"name": "Ali", "age": 17, "grade": "A"}
print(student["name"])  # Output: Ali
```

- **Using `get()` Method:**
  - Safely access a value without raising an error if the key doesn't exist.

```
print(student.get("age"))  # Output: 17
print(student.get("address", "Not Found")) # Output: Not Found
```

Adding, Updating, and Deleting Items

- **Adding Items:**

```
student["address"] = "Islamabad"
```

- **Updating Items:**

```
student["age"] = 17
```

- **Deleting Items:**
  - Use `del` to remove a key-value pair.

```
del student["grade"]
```

  - Use `pop()` to remove a key-value pair and return the value.

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

  - Use `clear()` to remove all items.

```
student.clear()
```

Dictionary Methods

- **`keys()`**: Returns a list of all keys.

```
print(student.keys())
# Output: dict_keys(['name', 'age', 'grade'])
```

- **`values()`**: Returns a list of all values.

```
print(student.values())
# Output: dict_values(['Ali', 17, 'A'])
```

- **`items()`**: Returns a list of key-value pairs as tuples.

```
print(student.items())
# Output: dict_items([('name', 'Ali'), ('age', 17), ('grade', 'A')])
```

Looping Through Dictionaries

- **Looping Through Keys:**

```
for key in student:
    print(key)
```

- **Looping Through Values:**

```
for value in student.values():
    print(value)
```

- **Looping Through Key-Value Pairs:**

```
for key, value in student.items():
    print(f"{key}: {value}")
```

Hands-On Practice

- **Task 1:** Create a dictionary to store student information (name, age, grade) and print it.

```
student = {"name": "Ali", "age": 17, "grade": "A"}
print(student)
```

- **Task 2:** Add a new key-value pair (e.g., `address`) and print the updated dictionary.

```
student["address"] = "Islamabad"
print(student)
```

- **Task 3:** Update the age and print the updated dictionary.

```
student["age"] = 17
print(student)
```

- **Task 4:** Remove the `grade` key and print the updated dictionary.

```
del student["grade"]
print(student)
```

- **Task 5:** Loop through the dictionary and print each key-value pair.

```
for key, value in student.items():
    print(f"{key}: {value}")
```

Recap

- Dictionaries store key-value pairs.
- Use keys to access, add, update, or delete items.
- Important methods: keys(), values(), items().
- Loop through dictionaries using for loops.

Homework

1. Create a dictionary to store your favorite book’s title, author, and year of publication. Print the dictionary.
2. Add a new key-value pair (e.g., genre) and print the updated dictionary.
3. Update the year and print the updated dictionary.
4. Loop through the dictionary and print each key-value pair.

Q&A

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

Closing

Next class: Sets