

PYTHON

Dictionary

Definition:

A **dictionary** is an **ordered collection** of data that stores values in a **key-value pair** format. It allows multiple values to be stored in a single variable and is **enclosed within curly braces {}**.

Defining a Dictionary

Creating a dictionary

```
info = {  
    "Name": "Hello",  
    "Age": 10  
}
```

Here,

- "Name" and "Age" are **keys**
 - "Hello" and 10 are **values**
-

Accessing Values

```
print(info["Name"]) # Accessing value using the key
```

```
print(info.get("Age")) # Using get() method to retrieve a value
```

- `info["Name"]` → Fetches the value "Hello"
 - `info.get("Age")` → Fetches the value 10
 - `info.get("Hello")` → Returns None (since "Hello" is not a key in the dictionary)
-

Fetching Keys, Values, and Items

```
print(info.keys()) # Returns a list of all keys
```

```
print(info.values()) # Returns a list of all values
```

```
print(info.items()) # Returns a list of key-value pairs (as tuples)
```

- `info.keys()` → Returns `dict_keys(['Name', 'Age'])`
 - `info.values()` → Returns `dict_values(['Hello', 10])`
 - `info.items()` → Returns `dict_items([('Name', 'Hello'), ('Age', 10)])`
-

Iterating Through a Dictionary

Method 1: Iterating Over Keys

```
for key in info.keys():
```

```
    print(key)    # Prints the key
```

```
    print(info[key]) # Prints the corresponding value
```

- `keys()` method is used to fetch keys
- `info[key]` retrieves the corresponding value

Method 2: Iterating Over Key-Value Pairs

for key, value in info.items():

 print(key) # Prints the key

 print(value) # Prints the value

- items() method returns key-value pairs as tuples
- The for loop unpacks them into key and value