

🔗 Dictionaries in Python

A dictionary in Python is a collection of **key-value pairs**. Each key in a dictionary is associated with a value, and you can retrieve or manipulate data using the key. Unlike lists and tuples, dictionaries are **unordered** and **mutable** (changeable).

🔗 1. Creating a Dictionary

You can create a dictionary using curly braces `{}` or the `dict()` function.

🔗 Syntax:

```
my_dict = {  
    "key1": "value1",  
    "key2": "value2",  
    "key3": "value3"  
}
```



🔗 1. Creating a Dictionary

You can create a dictionary using curly braces `{}` or the `dict()` function.

🔗 Syntax:

```
my_dict = {  
    "key1": "value1",  
    "key2": "value2",  
    "key3": "value3"  
}
```



🔗 Example:

Let's create a dictionary of famous cities in Karnataka and their popular dishes.


```
karnataka_food = {  
    "Bengaluru": "Bisi Be  
    "Mysuru": "Mysore Pak  
    "Mangaluru": "Neer Do  
}
```




2. Accessing Dictionary Elements

To access the values stored in a dictionary, you use the key.

Example:

```
print(karnataka_food["Mys 
```

You can also use the `get()` method to access values, which is safer because it doesn't throw an error if the key doesn't exist.

```
print(karnataka_food.get(   
print(karnataka_food.get(  
  

```

3. Adding and Updating Dictionary Elements

You can add new key-value pairs or update existing values in a dictionary.

Adding an Item:

```
karnataka_food["Shivamogg"]  
print(karnataka_food)
```

Updating an Item:


```
karnataka_food["Bengaluru"]
```



4. Removing Elements from a Dictionary

You can remove items from a dictionary using several methods:

- `pop()` : Removes the specified key and returns the associated value.

```
mysuru_food = karnata   
print(mysuru_food) #
```

- `del` : Removes the specified key.

```
del karnataka_food["M" 
```

- `clear()` : Empties the dictionary.

```
karnataka_food.clear( 
```

🔗 5. Dictionary Methods

Here are some common methods available for dictionaries:

- `keys()` : Returns all the keys in the dictionary.

```
print(karnataka_food. 
```


- `values()` : Returns all the values in the dictionary.

```
print(karnataka_food. 
```

- `items()` : Returns key-value pairs as tuples.

```
print(karnataka_food. 
```

- `update()` : Updates the dictionary with another dictionary or iterable.

```
new_dishes = {"Hubbal"   
karnataka_food.update
```

🔗 6. Dictionary Characteristics

- **Unordered:** Dictionary keys are not stored in any particular order.
- **Mutable:** You can change, add, or remove items.
- **Keys Must Be Immutable:** Keys in a dictionary must be of a data type that is immutable, such as a string, number, or tuple.
- **Unique Keys:** A dictionary cannot have duplicate keys. If you try to add a duplicate key, the latest value will overwrite the previous one.

Homework

1. Basic Dictionary Operations:

- Create a dictionary to store information about 5 cities in Karnataka and their famous dishes.
- Add a new city and its dish to the dictionary.
- Update the dish for Bengaluru.
- Remove one city from the dictionary.
- Use the `keys()` method to print all city names in the dictionary.
- Use the `values()` method to print all dishes in the dictionary.

2. **Nested Dictionary Practice**

(Simple for now):

- Create a dictionary to store details of two of your friends, including their names, favorite subject, and favorite food.
- Access and print the favorite food of one friend.