

Dictionaries in Python

Presentation by Abhisek Sarkar
as20ms091@iiserkol.ac.in

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

What is Python Dictionary?

A Python dictionary is a data structure that stores the value in key: value pairs. Values in a dictionary can be of any data type and can be duplicated, whereas keys can't be repeated and must be immutable.

Why do we use Dictionary in Python?

The data is stored in key:value pairs in dictionaries, which makes it easier to find values.

Example:

```
{1: 'Football', 2: 'Cricket', 3: 'Basketball'}
```

```
# create dictionary using { }  
d1 = {1: 'Football', 2: 'Cricket', 3:  
      'Basketball'}  
print(d1)
```

```
# create dictionary using dict() constructor  
d2 = dict(a = "Football", b = "Cricket", c =  
          "Basketball")  
print(d2)
```

Let's construct a Dictionary

```
# Output
```

```
{1: 'Football', 2: 'Cricket', 3:  
'Basketball'}
```

```
{'a': 'Football', 'b':  
'Cricket', 'c': 'Basketball'}
```

Output:

- From Python 3.7 Version onward, Python dictionary are Ordered.
- Dictionary keys are case sensitive: the same name but different cases of Key will be treated distinctly.
- Keys must be immutable: This means keys can be strings, numbers, or tuples but not lists.
- Keys must be unique: Duplicate keys are not allowed and any duplicate key will overwrite the previous value.
- Operations like search, insert, delete can be performed in Constant Time.

A few points:

```
d = { "name": "Abhisek", 1: "Python" }
```

```
# Access using key
```

```
print(d["name"]) #Output: Abhisek
```

```
# Access using get()
```

```
print(d.get("name")) # Output: Abhisek
```

Accessing Dictionary Items

```
d = {1: 'Chess', 2: 'Ludo', 3: 'Carrom'}
```

```
# Adding a new key-value pair
```

```
d[4] = 'poker'
```

```
# Updating an existing value
```

```
d[2] = "Bridge"
```

```
print(d)
```

Output:

```
{1: 'Chess', 2: 'Bridge', 3: 'Carrom', 4: 'poker'}
```

Adding and Updating Dictionary Items

We can remove items from dictionary using the following methods:

- **del**: Removes an item by key.
- **pop()**: Removes an item by key and returns its value.
- **clear()**: Empties the dictionary.
- **popitem()**: Removes and returns the last key-value pair.

```
d = {1: 'Cards', 2: 'Chess', 3: 'Ludo', 'age':22}

# Using del to remove an item
del d["age"]
print(d)           #output: {1: 'Cards', 2: 'Chess', 3: 'Ludo'}

# Using pop() to remove an item and return the value
val = d.pop(1)
print(val)         #output: Cards

# Using popitem to removes and returns
# the last key-value pair.
key, val = d.popitem()
print(f"Key: {key}, Value: {val}") #output: Key: 3, Value: Ludo

# Clear all items from the dictionary
d.clear()
print(d)           #output: {}
```

Example Code: