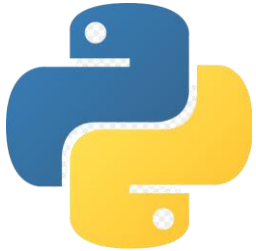


Lecture-6

Python Dictionaries and Sets

Content

- Create Dictionary
- Dictionary Operations
- Dictionary methods keys and values
- Create Set
- Working with Set

Dictionary

- Dictionaries are Python's most powerful data collection
- Non-sequence collections
- Unordered collection which stores key–value pairs
- Dictionary contains 2 things i.e - Key, value
- Key is unique & immutable (once declared can't changed)
- Key, value both contains different types of data

Create Dictionary

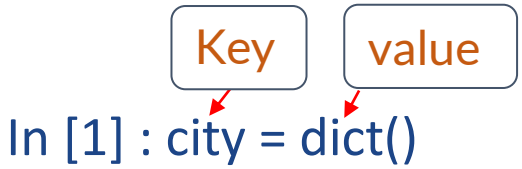
- Create a dictionary using dict() or {}

```
[1]   1 city = dict()
      2 city
```

```
{}
```

Create Dictionary

- Create a dictionary using dict()
- Add a key with corresponding value




```
[2]   1 city['Dhanmondi'] = 1205
```

```
[3]   1 city['Cantonment'] = 1206
```

```
[4]   1 city
```

```
{'Cantonment': 1206, 'Dhanmondi': 1205}
```

Iterate a Dictionary

- Use a for loop to iterate
- dictionary_name.**items( )** used to print key-value pair

```
[5]   1 print(city)
      2 for area, code in city.items():
      3   print(f'{area}’s area code is {code}')
```

```
{'Dhanmondi': 1205, 'Cantonment': 1206}
Dhanmondi’s area code is 1205
Cantonment’s area code is 1206
```

Dictionary Operations

- Access value with a key
- Update value with a key

```
[7]   1 # Accessing a value
      2 city["Cantonment"]

      1206
```

```
[8]   1 # updating/insert both key-value respectively
      2 city.update({'Mohammadpur':1207})
      3 # print
      4 city

      {'Cantonment': 1206, 'Dhanmondi': 1205, 'Mohammadpur': 1207}
```

Dictionary Operations

- Add new key-value pair
- Remove a key-value pair

```
[9]   1 # Update a value
      2 city['Cantonment'] = 1202
      3 # print to check
      4 city

      {'Cantonment': 1202, 'Dhanmondi':
```

```
[14]  1 # remove a key-value pair
      2 del city['Cantonment']
      3 # print
      4 city

      {'Dhanmondi': 1205, 'Mohammadpur': 1208, 'Mohammadpur Housing': 1207}
```

Set

- Set is unordered collections
- Non-sequence collections
- Contains unique value

Creating a Set

```
[2]   1  colors = set()
      2  colors = {'red','green','blue','yellow','red'}
      3  colors
```

```
{'blue', 'green', 'red', 'yellow'}
```

Set is unique. You can see ‘red’ occurred in set 2 times, but has displayed only one time

Set Operations

- Checking Whether a Value Is in a Set
- ‘red’ inside colors
- ‘purple’ is not

```
[3]   1   'red' in colors
```

```
True
```

```
[4]   1   'purple' in colors
```

```
False
```

Set Operations

- Determine length of a Set
- 5 elements of output is 5

```
[6]   1  colors = {'red','green','blue','yellow','orange'}
      2  len(colors)
```

```
5
```

`.upper()` used for uppercase all charachters/letters

```
[22]  1 for color in colors:
      2   print(color.upper(), end=' ')
```

```
BLUE RED YELLOW ORANGE GREEN
```

Set Methods

- Add element into a Set. **add( )** method use for this

```
[7]   1  colors = {'red','green','blue','yellow','orange'}
      2  colors.add('pink')
      3  colors
```

```
{'blue', 'green', 'orange', 'pink', 'red', 'yellow'}
```

- Remove element from a Set. **remove( )** method use for this

```
[8]   1  colors = {'red','green','blue','yellow','orange'}
      2  colors.remove('yellow')
      3  colors
```

```
{'blue', 'green', 'orange', 'red'}
```

Set Methods

- Delete all values from a Set
- .clear( ) method use for this
- Return an empty set

```
[9]  1  colors = {'red','green','blue','yellow','orange'}
     2  colors.clear()
     3  colors
```

```
set()
```

Set Mathematical Operations

- A Set is equal to another set or not

```
[32]  1 {1, 3, 5} == {3, 5, 1}
```

```
True
```

```
[33]  1 {1, 3, 5} != {3, 5, 1}
```

```
False
```

Set Mathematical Operations

- Subset and Superset

```
[34]  1 {1, 2}.issubset({3, 5, 1})
```

```
False
```

```
[35]  1 {1, 3, 5}.issuperset({3, 5, 1})
```

```
True
```

```
[36]  1 {1, 3, 5}.issuperset({3, 2})
```

```
False
```

Set Mathematical Operations

- Mathematical union operations
- | or union used for that

```
[38]  1 {1, 3, 5} | {2, 3, 4}
```

```
{1, 2, 3, 4, 5}
```

```
[39]  1 {1, 3, 5}.union([20, 20, 3, 40, 40])
```

```
{1, 3, 5, 20, 40}
```

Set Mathematical Operations

- Mathematical intersection operations
- **&** or intersection used for that

```
[40]  1 {1, 3, 5} & {2, 3, 4}
```

```
{3}
```

```
[41]  1 {1, 3, 5}.intersection([1, 2, 2, 3, 3, 4, 4])
```

```
{1, 3}
```

Set Mathematical Operations

- Mathematical difference operations
- - or difference used for that

```
[42]  1 {1, 3, 5} - {2, 3, 4}
```

```
{1, 5}
```

```
[43]  1 {1, 3, 5, 7}.difference([2, 2, 3, 3, 4, 4])
```

```
{1, 5, 7}
```

Thank You