



# SET in Python – Unit 5

By: Bhavika Vaghela  
Asst.Prof.  
PICA

# Set in Python

- Set is an unordered collection of items. Every element is unique (no duplicates) and must be immutable.
- the set itself is mutable (we can add or remove items).
- Use to perform mathematical operation like union, intersection, symmetric difference.

- Syntax : { } # set use curly braces

```
set1 = {1,2,3}
```

```
set2 = {1,"helo",3,11.45} //mixed data
```

**try this**

```
set3 = {1,3,5,'h',1,'h'} ????
```

```
my_set = {1, 2, [3, 4]} ???? set cannot have mutable items.
```

## Set cont..

- but we can make set from a list

```
>>> set([1,2,3,2])
```

```
{1, 2, 3}
```

- **Empty set** : like list we can not create the empty set like : `a = { }`

```
type(a) //out put : <class 'dict'>
```

- But for this we have to use `set()` function Like :

```
b=set()
```

```
type(b) // output : <class 'set'>
```

## Set cont..

### Changing the set :

- indexing have no meaning. We cannot access or change an element of set using indexing or slicing.
- We can add single elements using the method **add()**.
- Multiple elements can be added using **update()** method.
- The **update()** method can take tuples, lists, strings or other sets as its argument.
- In all cases, duplicates are avoided.
- A particular item can be removed from set using methods like **discard()** and **remove()**.
- while using **discard()** if the item does not exist in the set, it remains unchanged. But **remove()** will raise an error in such condition.

## Set cont..

```
>>> my_set = {1,3}
>>> my_set[0]
...
TypeError: 'set' object does not support indexing
>>> my_set.add(2)
>>> my_set
{1, 2, 3}
>>> my_set.update([2,3,4])
>>> my_set
{1, 2, 3, 4}
>>> my_set.update([4,5], {1,6,8})
>>> my_set
{1, 2, 3, 4, 5, 6, 8}
```

## Set cont..

```
>>> my_set = {1, 3, 4, 5, 6}
>>> my_set.discard(4)
>>> my_set
{1, 3, 5, 6}
>>> my_set.remove(6)
>>> my_set
{1, 3, 5}
>>> my_set.discard(2)
>>> my_set
{1, 3, 5}
>>> my_set.remove(2)
...
KeyError: 2
```

## Set cont..

- we can remove and return an item using the **pop()** method.
- Set being unordered, there is no way of determining which item will be popped. It is completely **arbitrary**. We can also remove all items from a set using **clear()**.

```
>>> my_set = set("HelloWorld")
>>> my_set.pop()
'r'
>>> my_set.pop()
'W'
>>> my_set
{'d', 'e', 'H', 'o', 'l'}
>>> my_set.clear()
>>> my_set
set()
```

# Set Operation

- set operations like union, intersection, difference and symmetric difference.
- We can do this with operators or methods.

**Union** : Union is performed using | operator. Same can be accomplished using the method union().

```
a = {1,2,3}
```

```
b = {2,5,7}
```

```
a | b    // output : set([1, 2, 3, 5, 7])
```

```
a.union(b) // output : set([1, 2, 3, 5, 7])
```

```
b.union(a)
```

```
c = a|b
```

```
print c    //output : set([1, 2, 3, 5, 7])
```



## Set operation cont..

**Intersection** : elements that are common in both sets.  
Intersection is performed using & operator.

- Same can be accomplished using the method **intersection()**.

**a & b**

**set([2])**

**>>> a.intersection(b)**

**set([2])**

**>>> b.intersection(a)**

**set([2])**

## Set operation cont...

**Set difference** :  $(A - B)$  is a set of elements that are only in A but not in B. Similarly,  $B - A$  is a set of element in B but not in A.

- Difference is performed using - operator. Same can be accomplished using the method **difference()**.

**a-b**

```
set([1, 3])
```

```
>>> b-a
```

```
set([5, 7])
```

```
a.difference(b)
```

```
set([1, 3])
```

## Set operation cont...

**Symmetric Difference** : A and B is a set of element in both A and B except those common in both.

- Symmetric difference is performed using ^ operator. Same can be accomplished using the method **symmetric\_difference()**.

**a^b**

```
set([1, 3, 5, 7])
```

```
>>> b^a
```

```
set([1, 3, 5, 7])
```

```
>>> a.symmetric_difference(b)
```

```
set([1, 3, 5, 7])
```

For more method :

<http://www.programiz.com/python-programming/set>

# Python Frozen set

- its elements cannot be changed once assigned.
- Frozen sets are immutable sets. Sets being mutable are un-hashable, so they can't be used as dictionary keys.
- On the other hand, frozen sets are hashable and can be used as keys to a dictionary.
- created using the function **frozenset()**.
- This datatype supports methods like `copy()`, `difference()`, `intersection()`, `isdisjoint()`, `issubset()`, `issuperset()`, `symmetric_difference()` and `union()`.
- Being immutable it does not have method that add or remove elements.

## Frozen set cont...

```
set1 = frozenset([1,2,3,4])
```

```
set2 = frozenset(['a',1,2,4,'b'])
```

```
set1|set2
```

```
Output : frozenset(['a', 1, 2, 3, 4, 'b'])
```

```
set1.isdisjoint(set2)
```

```
Output : False
```

```
set1.difference(set2)
```

```
Output : frozenset([3])
```

```
Try this : set1.add('j') ????
```

## Develop below program for set

- Write a program for set which show the use of membership operator.
- Iterate the set using for loop as well as while loop.
- Program to Illustrate Different Set Operations like union, intersection, difference, symmetric difference.
- Write a program for set which show the use of `copy()`, `issubset()`, `issuperset()`, `all()` and `any()`.
- WAP to perform all set operation using user defined function. and if possible create menu driven program where user can enter his/her choice to perform set operation.
- WAP which show the use of set methods `add()`, `update()`, `discard()`, `remove()`, `pop()` and `clear()`.

- WAP to create frozen set and try to implement all above listed methods on it. and also perform union and intersect operations.
- Write a program for set which shows the use of `copy()`, `issubset()`, `issuperset()`, `all()` and `any()`.
- WAP to iterate the set using for loop also try using while loop.
- WAP to find missing and additional values in two sets.
- WAP to find maximum and minimum from set.
- <https://www.youtube.com/watch?v=xEb8kberFzo> (list comprehension)
- <https://snakify.org/en/lessons/sets/>