Set

- Used to store multiple items in a single variable.
- Collection which is `unordered`, `unchangeable`, and `unindexed`, `duplicates are not allowed`
- Unordered means that the items in a set do not have a defined order.
- unchangeable, means that we cannot change the items after the set has been created.
- No duplicates, means cannot have two items with the same value.
- Starts with {}

```
st = {"apple", "banana", "cherry"}
```

Methods

add()

Adds an `element` to the set.

```
fruits = {"apple", "banana", "cherry"}
fruits.add("apple")
print(fruits)
```

clear()

`Removes all the elements` from the set

```
fruits = {"apple", "banana", "cherry"}
fruits.clear()
print(fruits)
```

copy()

Returns a `copy` of the set

```
fruits = {"apple", "banana", "cherry"}
newfruits = fruits.copy()
print(newfruits)
```

difference()

Returns a set `containing` the `difference between two or more sets`.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
result = set1.difference(set2)
print(result)
```

difference_update()

`Removes` the items `in this set that are also included in another`, specified set.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
set1.difference_update(set2)
print(set1)
```

discard()

`Remove` the `specified` item.

```
fruits = {"apple", "banana", "cherry"}
fruits.discard("banana")
print(fruits)
```

intersection()

Returns a set, that is the `intersection` of two other sets.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
result = set1.intersection(set2)
print(result)
```

intersection_update()

`Removes` the items in this set that are `not present in other`, specified set(s).

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
set1.intersection_update(set2)
print(set1)
```

isdisjoint()

- Returns whether `two sets have an intersection` or not.

```
set1 = {1, 2, 3}
set2 = {4, 5, 6}
result = set1.isdisjoint(set2)
print(result)
```

issubset()

- Returns whether `another set contains this set` or not.

```
set1 = {1, 2}
set2 = {1, 2, 3, 4}
result = set1.issubset(set2)
print(result)
```

issuperset()

- Returns whether `this set contains another set` or not.

```
set1 = {1, 2, 3, 4}
set2 = {1, 2}
result = set1.issuperset(set2)
print(result)
```

pop()

- `Removes` an `element` from the set.

```
fruits = {"apple", "banana", "cherry"}
removed_item = fruits.pop()
print(removed_item, fruits)
```

remove()

- `Removes` the `specified element`.

```
fruits = {"apple", "banana", "cherry"}
fruits.remove("banana")
print(fruits)
```

symmetric_difference()

- Returns a set with the `symmetric differences` of two sets.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
result = set1.symmetric_difference(set2)
print(result)
```

symmetric_difference_update()

- `Inserts` the `symmetric differences` from this set and another.

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
set1.symmetric_difference_update(set2)
print(set1)
```

union()

- Return a set containing the `union` of sets.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
result = set1.union(set2)
print(result)
```

update()

- `Update` the set with the `union` of this set and others.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
set1.update(set2)
print(set1)
```