

Set

A set is a collection which is unordered and unindexed in a way that we cannot be sure in which order the items will appear.

Set Example

```
RYB_color = {"Red", "Yellow", "Blue"}
print(RYB_color)
>>> {'Red', 'Yellow', 'Blue'}
```

The set() Constructor

```
RYB_color = set(("Red", "Yellow", "Blue"))
```

Access Items

Example 1

```
for x in RYB_color:
    print(x)

>>> Red
>>> Yellow
>>> Blue
```

Example 2

```
print("Yellow" in RYB_color)
>>> True
```

Change Items

Since a set is not ordered neither indexed, then we cannot access to any item to change its value.

Add Items

```
RYB_color.add("White")
print(RYB_color)
>>> {"Red", "White", "Yellow", "Blue"}
```

Get the Length of a Set

```
print(len(RYB_color))
>>> 3
```

Delete a set

```
del RYB_color
```

Remove Item

```
RYB_color.remove("Yellow")
print(RYB_color)
>>> {'Red', 'Blue'}
```

If the item doesn't exist then `remove()` will raise an error.

```
RYB_color.discard("Yellow")
print(RYB_color)
>>> {'Red', 'Blue'}
```

If the item doesn't exist then `discard()` will raise an error.

```
c = thisset.pop()
print(c)
print(thisset)
>>> Yellow
>>> {'Red', 'Blue'}
```

Since a set is not ordered neither indexed, then we cannot change access to any item to change its value.

```
RYB_color.clear()
```

`clear()` returns an empty set.

Join Two Sets

```
Second_color = {"Green", "Orange", "Purple"}
Color = RYB_color.union(Second_color)
print(Color)
>>> {'Yellow', 'Orange', 'Red', 'Green', 'Blue', 'Purple'}
```

`union()` joins two sets into a new one and excludes any duplicate items.

```
RYB_color.update(Second_color)
print(RYB_color)
>>> {'Yellow', 'Orange', 'Red', 'Green', 'Blue', 'Purple'}
```

`update()` inserts the items in `Second_color` into `RYB_color` and excludes any duplicate items.



By [Nouha_Thabet](#)
cheatography.com/nouha-thabet/

Published 14th December, 2019.
 Last updated 19th December, 2019.
 Page 1 of 1.

Sponsored by [Readable.com](#)
 Measure your website readability!
<https://readable.com>