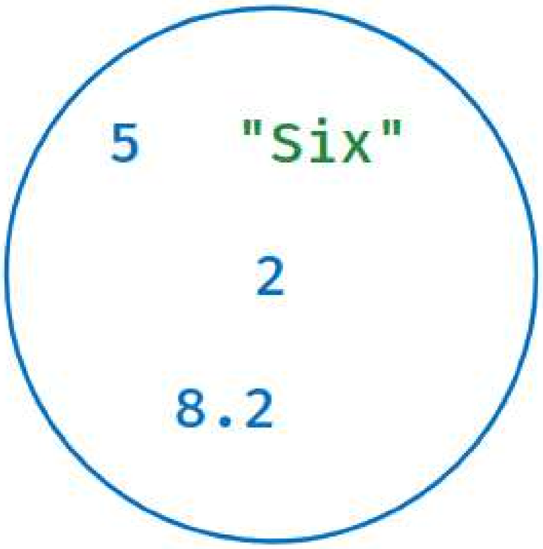
Cheat Sheet

Sets

Unordered collection of items.

Every set element is

- Unique (no duplicates)
- Must be immutable



Creating a Set

- Created by enclosing elements within *{curly}* brackets.
- Each item is separated by a comma.

Code

PYTHON

```python
a = 2
set_a = {5, "Six", a, 8.2}
print(type(set_a))
print(set_a)
```

Output

```
<class 'set'>
{8.2, 2, 'Six', 5}
```

Need not be in the same order as defined.

No Duplicate Items

Sets contain unique elements

Code

PYTHON

```python
set_a = {"a", "b", "c", "a"}
print(set_a)
```

Output

```
{'b', 'a', 'c'}
```

Immutable Items

Set items must be immutable.
As List is mutable, Set cannot have list as an item.

Code

PYTHON

```python
set_a = {"a", ["c", "a"]}
print(set_a)
```

Output

```
TypeError: unhashable type: 'list'
```

Creating Empty Set

We use

`set()` to create an empty set.

Code

PYTHON

```python
set_a = set()
print(type(set_a))
print(set_a)
```

Output

```
<class 'set'>
set()
```

Converting to Set

`set(sequence)` takes any sequence as argument and converts to set, avoiding duplicates

List to Set

Code

PYTHON

```python
set_a = set([1,2,1])
print(type(set_a))
print(set_a)
```

Output

```
<class 'set'>
```

```
{1, 2}
```

String to Set

Code

PYTHON

```python
set_a = set("apple")
print(set_a)
```

Output

```
{'l', 'p', 'e', 'a'}
```

Tuple to Set

Code

PYTHON

```python
set_a = set((1, 2, 1))
print(set_a)
```

Output

```
{1, 2}
```

Accessing Items

As sets are unordered, we cannot access or change an item of a set using

- Indexing
- Slicing

Code

PYTHON

```python
set_a = {1, 2, 3}
print(set_a[1])
print(set_a[1:3])
```

Output

```
TypeError: 'set' object is not subscriptable
```

Adding Items

`set.add(value)` adds the item to the set, if the item is not present already.

Code

PYTHON

```python
set_a = {1, 3, 6, 2, 9}
set_a.add(7)
print(set_a)
```

Output

```
{1, 2, 3, 6, 7, 9}
```

Adding Multiple Items

`set.update(sequence)` adds multiple items to the set, and duplicates are avoided.

Code

PYTHON

```python
set_a = {1, 3, 9}
```

```
set_a.update([2, 3])
print(set_a)
```

Output

```
{2, 1, 3, 9}
```

Removing Specific Item

`set.discard(value)` takes a single value and removes if present.

Code

PYTHON

```
set_a = {1, 3, 9}
set_a.discard(3)
print(set_a)
```

Output

```
{1, 9}
```

`set_a.remove(value)` takes a value and remove if it present or raise an error.

Code

PYTHON

```
set_a = {1, 3, 9}
set_a.remove(5)
print(set_a)
```

Output

```
    KeyError: 5
```

Operations on Sets

You can perform the following operations on Sets

- clear()
- len()
- Iterating
- Membership Check

Submit Feedback