Python sets

- A `set` is an unordered, mutable collection of unique items
- Items in Python set must be immutable (for the same reason keys in a dictionary must be immutable)
- Create a set with: `my_set = set([1, 2, 3])`
- Or create a set with: `my_set = {5, 8, "str", 49.2}`
- We can test for existence in a set and perform set operations

Set examples

```
# create and update using add method
myclasses = set()
myclasses.add("math")
myclasses.add("chemistry")
myclasses.add("literature")

# create using {} notation
yourclasses = {"physics", "gym", "math"}
```

```
myclasses
```

```
yourclasses
```

Test for membership with the `in` operator:

```
"gym" in myclasses
```

```
"gym" in yourclasses
```

Compute set intersections:

```
myclasses & yourclasses
```

Compute set union:

```
myclasses | yourclasses # union
```

Find unique items in a list

Let's create a list with non-unique elements:

```
basket = ['apple', 'orange', 'apple', 'pear', 'orange', 'banana']
```

From this list, we create a `set`:

```
fruit = set(basket)
print(fruit)
```

Set methods

For more information and examples see the `set` documentation in the official Python Tutorial and Library Reference.

Also see `help(set)`:

```
add(...)
    Add an element to a set.

    This has no effect if the element is already present.

clear(...)
    Remove all elements from this set.

copy(...)
    Return a shallow copy of a set.

difference(...)
    Return the difference of two or more sets as a new set.

    (i.e. all elements that are in this set but not the others.)

difference_update(...)
    Remove all elements of another set from this set.

discard(...)
    Remove an element from a set if it is a member.

    If the element is not a member, do nothing.

intersection(...)
    Return the intersection of two sets as a new set.

    (i.e. all elements that are in both sets.)

intersection_update(...)
    Update a set with the intersection of itself and another.

isdisjoint(...)
    Return True if two sets have a null intersection.

issubset(...)
    Report whether another set contains this set.

issuperset(...)
    Report whether this set contains another set.

pop(...)
    Remove and return an arbitrary set element.
    Raises KeyError if the set is empty.

remove(...)
    Remove an element from a set; it must be a member.

    If the element is not a member, raise a KeyError.

symmetric_difference(...)
    Return the symmetric difference of two sets as a new set.

    (i.e. all elements that are in exactly one of the sets.)
```

```
symmetric_difference_update(...)
    Update a set with the symmetric difference of itself and another.

union(...)
    Return the union of sets as a new set.

    (i.e. all elements that are in either set.)

update(...)
    Update a set with the union of itself and others.
```