Python basics: Sets

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

No indexing

A set is an unordered collection of distinct hashable objects

No duplicates

Don’t worry about this for now ☺

Numbers and strings and other things!

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

For example...

```
my_set = {1, 2, 3, 4, 5}

my_empty_set = set()

print(my_set)
print(my_empty_set)
```

Note that {} does not create an empty set, it creates an empty dictionary

```
{1, 2, 3, 4, 5}
set()
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

We can use it to remove duplicates in a list

```
my_list = [1, 2, 3, 4, 5, 5, 5, 5, 5]

my_set = set(my_list)

print(my_set)
```

```
{1, 2, 3, 4, 5}
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Or check for items in one but not in another

```
my_list = [1, 2, 3, 4, 5, 5, 5, 5, 5]
my_second_list = [4, 5, 5, 5, 6, 7, 8]

set_difference = set(my_list).difference(set(my_second_list))

print(set_difference)
```

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

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Or items common to both

```
my_list = [1, 2, 3, 4, 5, 5, 5, 5, 5]
my_second_list = [4, 5, 5, 5, 6, 7, 8]

set_intersection = set(my_list).intersection(set(my_second_list))

print(set_intersection)
```

```
{4, 5}
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Or get distinct items from both

```python
my_list = [1, 2, 3, 4, 5, 5, 5, 5, 5]
my_second_list = [4, 5, 5, 5, 6, 7, 8]

set_union = set(my_list).union(set(my_second_list))

print(set_union)
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

Alternate syntax

```python
set_union = set(my_list) | set(my_second_list)
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Or find items in one or the other but not both

```
my_list = [1, 2, 3, 4, 5, 5, 5, 5, 5]
my_second_list = [4, 5, 5, 5, 6, 7, 8]

set_symmetric_difference = (set(my_list)
                            .symmetric_difference(set(my_second_list)))

print(set_symmetric_difference)
```

```
{1, 2, 3, 6, 7, 8}
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

We can check for subsets

```
my_set = {1, 2, 3, 4, 5}
my_subset = {1, 2, 3}

is_subset = my_subset.issubset(my_set)

print(is_subset)
```

```
True
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Or for supersets

```
my_set = {1, 2, 3, 4, 5}
my_subset = {1, 2, 3}

is_superset = my_set.issuperset(my_subset)

print(is_superset)
```

```
True
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

We can test for simple membership

```
my_set = {1, 2, 3, 4, 5}
member = 1

is_member = member in my_set

print(is_member)
```

```
True
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Or for non-membership

```
my_set = {1, 2, 3, 4, 5}
non_member = 6

is_not_member = non_member not in my_set

print(is_not_member)
```

```
True
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Return the length of the set

```
my_set = {1, 2, 3, 4, 5}

set_length = len(my_set)

print(set_length)
```

```
5
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

And much more!

Set Method	Explanation
.add(x)	Adds element x to the set.
.remove(x)	Removes element x from the set.
.discard(x)	Removes element x from the set if it exists; does nothing if x is not in the set.
.pop()	Removes and returns an arbitrary element from the set. Raises an error if the set is empty.
.clear()	Removes all elements from the set.
.copy()	Returns a shallow copy of the set.
.union(other_set)	Returns a new set containing all unique elements from both sets.
.intersection(other_set)	Returns a new set containing common elements between both sets.
.difference(other_set)	Returns a new set containing elements that are in the set but not in other_set.
.symmetric_difference(other_set)	Returns a new set containing elements that are unique to each set.
.issubset(other_set)	Checks if the set is a subset of other_set. Returns True or False.
.issuperset(other_set)	Checks if the set is a superset of other_set. Returns True or False.
len(set)	Returns the number of elements in the set.

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata

Bonus! Set comprehensions!

```
# Return the distinct consonants from a sentence
{c
 for c in 'really long sentence with lots of letters'
 if c not in 'aeiou' and c not in ' '}
```

```
{'c', 'f', 'g', 'h', 'l', 'n', 'r', 's', 't', 'w', 'y'}
```

Owen Price
https://www.flexyourdata.com
https://www.youtube.com/@flexyourdata