Python Set Methods:

Sure, here are some of the most commonly used methods for sets in Python:

1. `add()`: Adds an element to the set. If the element already exists in the set, it won't be added again.

```
my_set = {1, 2, 3}
my_set.add(4)
print(my_set) # Output: {1, 2, 3, 4}
```

2. `remove()`: Removes an element from the set. If the element doesn't exist in the set, a `KeyError` will be raised.

```
my_set = {1, 2, 3}
my_set.remove(2)
print(my_set) # Output: {1, 3}
```

3. `discard()`: Removes an element from the set if it exists. If the element doesn't exist in the set, no error will be raised.

```
my_set = {1, 2, 3}
my_set.discard(2)
print(my_set) # Output: {1, 3}
```

4. `pop()`: Removes and returns an arbitrary element from the set. If the set is empty, a `KeyError` will be raised.

```
my_set = {1, 2, 3}
elem = my_set.pop()
print(my_set) # Output: {2, 3}
print(elem) # Output: 1
```

5. `clear()`: Removes all elements from the set.

```

```
my_set = {1, 2, 3}
my_set.clear()
print(my_set) # Output: set()
```

6. `union()`: Returns a new set that contains all the elements from the current set and another set.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
union_set = set1.union(set2)
print(union_set) # Output: {1, 2, 3, 4, 5}
```

7. `intersection()`: Returns a new set that contains only the elements that are in both the current set and another set.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
intersection_set = set1.intersection(set2)
print(intersection_set) # Output: {3}
```

8. `difference()`: Returns a new set that contains only the elements that are in the current set but not in another set.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
difference_set = set1.difference(set2)
print(difference_set) # Output: {1, 2}
```

9. `symmetric_difference()`: Returns a new set that contains only the elements that are in either the current set or another set, but not in both.

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
symmetric_difference_set = set1.symmetric_difference(set2)
print(symmetric_difference_set) # Output: {1, 2, 4, 5}
```

These are just some of the most commonly used methods for sets in Python. There are several other methods available, as well as set operations like subset and superset checks, that you can explore in the Python documentation.