

Python Sets Notes

1. Introduction to Sets

- A set is an unordered collection of items. Every element in a set is unique, meaning no duplicates.
- Sets are mutable, but the elements in a set must be immutable (i.e., they cannot be changed).

Example:

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

2. Creating a Set

- Sets can be created by using curly braces `{}` or the `set()` function.

Example:

```
# Using curly braces
my_set = {1, 2, 3}

# Using the set() function
my_set = set([1, 2, 3])
```

3. Adding Elements

- Use `add()` to add a single element to a set.
- Use `update()` to add multiple elements.

Example:

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

my_set.update([4, 5])
print(my_set) # Output: {1, 2, 3, 4, 5}
```

4. Removing Elements

- Use `remove()` to remove an element. It raises a `KeyError` if the element is not found.
- Use `discard()` to remove an element without raising an error.
- Use `pop()` to remove and return a random element.

Example:

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

my_set.discard(5) # No error even though 5 is not in the set

popped_element = my_set.pop()
print(f"Popped element: {popped_element}")
```

5. Set Operations

- Sets support mathematical operations like union, intersection, difference, and symmetric difference.

Example:

```
set_a = {1, 2, 3}
set_b = {3, 4, 5}

# Union
print(set_a | set_b) # Output: {1, 2, 3, 4, 5}

# Intersection
print(set_a & set_b) # Output: {3}

# Difference
print(set_a - set_b) # Output: {1, 2}

# Symmetric Difference
print(set_a ^ set_b) # Output: {1, 2, 4, 5}
```

6. Set Methods

- `union()`: Returns the union of sets.
- `intersection()`: Returns the intersection of sets.
- `difference()`: Returns the difference between sets.
- `symmetric_difference()`: Returns elements that are in either of the sets but not in both.

Example:

```
set_a = {1, 2, 3}
set_b = {3, 4, 5}

print(set_a.union(set_b)) # Output: {1, 2, 3, 4, 5}
print(set_a.intersection(set_b)) # Output: {3}
```

```
print(set_a.difference(set_b)) # Output: {1, 2}
print(set_a.symmetric_difference(set_b)) # Output: {1, 2, 4, 5}
```

7. Checking Subsets and Supersets

- `issubset()`: Checks if a set is a subset of another set.
- `issuperset()`: Checks if a set is a superset of another set.

Example:

```
set_a = {1, 2, 3}
set_b = {1, 2}

print(set_b.issubset(set_a)) # Output: True
print(set_a.issuperset(set_b)) # Output: True
```

8. Copying Sets

- `copy()`: Returns a shallow copy of the set.

Example:

```
set_a = {1, 2, 3}
set_b = set_a.copy()

print(set_b) # Output: {1, 2, 3}
```

9. Clearing a Set

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

Example:

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

10. Frozen Sets

- A frozen set is an immutable version of a set.
- Elements cannot be added or removed from a frozen set.

Example:

```
frozen_set = frozenset([1, 2, 3])  
print(frozen_set) # Output: frozenset({1, 2, 3})
```