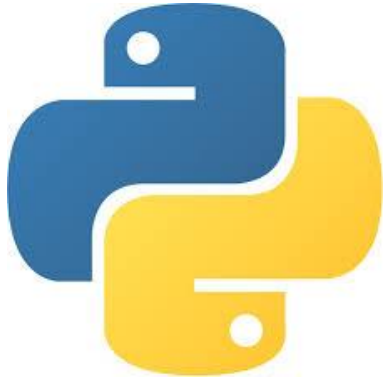
Python - Sets



Shahriar's Intelligence Academy

Sets

Definition:

A set is a built-in data structure in Python used to store unique and unordered elements.

Characteristics:

- **Unique Elements:** A set automatically removes duplicate entries.
- **Unordered:** Elements in a set do not have a specific order, and indexing is not supported.
- **Mutable:** While the set is mutable, it can only contain elements like numbers, strings, or tuples.

Example:

```
# Creating a set
my_set = {1, 2, 3, 4, 4, 5}
print(my_set)  # Output: {1, 2, 3, 4, 5}
```

List-to-Set Conversion

Method:

Use the **set()** function to convert a list into a set.

Example:

```
# List with duplicates
my_list = [1, 2, 2, 3, 4, 4, 5]

# Converting list to set
my_set = set(my_list)

print("Original List:", my_list)  # Output: [1, 2, 2, 3, 4, 4, 5]
print("Set:", my_set)  # Output: {1, 2, 3, 4, 5}
```

Add Set Elements

Key Features:

- Use the **add()** method to add a **single element**.
- Use the **update()** method to add **multiple elements**.

Example:

```
# Creating a set
my_set = {1, 2, 3}

# Adding elements
my_set.add(4)  # Adds 4
my_set.update([5, 6])  # Adds 5 and 6
print("After Adding:", my_set)  # Output: {1, 2, 3, 4, 5, 6}
```

Remove Set Elements

Key Features:

- Use the **remove()** method to delete an element ***(throws an error if the element does not exist).***
- Use the **discard()** method to delete an element ***(does not throw an error if the element does not exist).***

Example:

```
# Using remove() method
my_set.remove(2)  # Removes 2 (if it exists)
print("After remove(2):", my_set)  # Output: {1, 3, 4}

# Using discard() method
my_set.discard(10)  # Does nothing (10 is not in the set)
print("After discard(10):", my_set)  # Output: {1, 3, 4}
```

Set Union

Key Features:

- Combines all unique elements from two or more sets.
- Use the **union()** method or the **|** operator.

Example:

```
# Creating two sets
set_a = {1, 2, 3, 4}
set_b = {3, 4, 5, 6}

# Union
union_set = set_a.union(set_b)  # Using union()
print("Union:", union_set)  # Output: {1, 2, 3, 4, 5, 6}
```

Set Intersection

Key Features:

- Returns only the common elements between two or more sets.
- Use the **intersection()** method or the **&** operator.

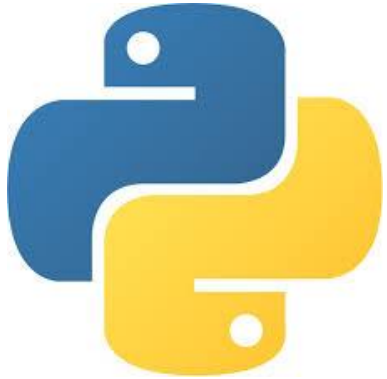
Example:

```
# Intersection
intersection_set = set_a.intersection(set_b)  # Using intersection()
print("Intersection:", intersection_set)  # Output: {3, 4}
```

Code with Operators:

```
# Using operators
print("Union with |:", set_a | set_b)  # Output: {1, 2, 3, 4, 5, 6}
print("Intersection with &:", set_a & set_b)  # Output: {3, 4}
```

Thank You!



Shahriar's Intelligence Academy