

Chapter 13 - 8Sets

Theory:

Sets are unordered collections of unique elements.

Code Example:

```
# ■ Python Set Basics
# -----
# ■ What is a Set?
# A set is a collection of well-defined, unique, and unordered objects.
# Properties:
# - Unindexed
# - Mutable (can change)
# - No duplicate values allowed
# Creating sets
s = {1, 5, 32}
print("Original Set:", s)
# ■ Empty set
s = set() # This creates an empty set (NOT {})
print("Empty Set:", s)
# ■ Set with mixed types
s = {1, 5, 32, "harry"}
print("Mixed Set:", s)
# -----
# ■ Set Methods
# -----
# ■ add() → Adds a single element
s.add(31)
print("After add(31):", s)
s.add(3)
print("After add(3):", s)
# ■ update() → Adds multiple elements
s = {"apple", "banana"}
s.update(["grape", "melon"])
print("After update:", s)
# ■ remove() → Removes element (error if not present)
number = {1, 3, 5, 6, 7}
number.remove(3)
print("After remove(3):", number)
# ■ discard() → Removes element (no error if not present)
number = {1, 4, 6, 8, 66, 55, 33}
number.discard(4)
print("After discard(4):", number)
# ■ clear() → Removes all elements
number = {1, 4, 6, 8, 66, 55, 33}
number.clear()
print("After clear():", number)
# ■ union() → Combines all unique elements
s1 = {1, 3, 4, 5}
s2 = {6, 7, 9, 8}
print("Union of s1 and s2:", s1.union(s2))
# ■ intersection() → Common elements
a = {1, 2, 3, 4}
b = {4, 2, 7, 8}
```

```
print("Intersection of a and b:", a.intersection(b))
# ■ difference() → Elements in a but not in b
a = {1, 2, 3}
b = {2, 4}
print("Difference (a - b):", a.difference(b)) # Output: {1, 3}
# ■ symmetric_difference() → Elements in either set but not both
a = {1, 2, 3}
b = {3, 4}
print("Symmetric Difference:", a.symmetric_difference(b)) # Output: {1, 2, 4}
# ■ copy() → Creates a shallow copy of the set
a = {1, 2, 3}
b = a.copy()
print("Copied set b:", b)
```