

text2-1

February 3, 2025

```
[1]: # Creating two sets
set_a = {1, 2, 3, 4, 5}
set_b = {4, 5, 6, 7, 8}

# 1. add() - Add a new element to set_a
set_a.add(6)
print("After add(6):", set_a)

# 2. remove() - Remove an element from set_a
set_a.remove(2) # This will remove element 2 from set_a
print("After remove(2):", set_a)

# 3. discard() - Discard an element from set_b
set_b.discard(10) # No error will be raised even if element is not found
print("After discard(10):", set_b)

# 4. pop() - Remove and return a random element from set_a
popped_element = set_a.pop()
print("Popped element:", popped_element)
print("After pop():", set_a)

# 5. clear() - Clear all elements from set_b
set_b.clear()
print("After clear():", set_b)

# 6. copy() - Create a shallow copy of set_a
set_copy = set_a.copy()
print("Copy of set_a:", set_copy)

# 7. union() - Union of two sets
union_set = set_a.union({6, 7, 8})
print("Union of set_a and {6, 7, 8}:", union_set)

# 8. intersection() - Intersection of two sets
intersection_set = set_a.intersection({4, 5, 6})
print("Intersection of set_a and {4, 5, 6}:", intersection_set)
```

```

# 9. difference() - Difference between two sets
difference_set = set_a.difference({4, 5, 6})
print("Difference between set_a and {4, 5, 6}:", difference_set)

# 10. symmetric_difference() - Symmetric difference between two sets
sym_diff_set = set_a.symmetric_difference({4, 5, 6})
print("Symmetric difference between set_a and {4, 5, 6}:", sym_diff_set)

# 11. issubset() - Check if set_a is a subset of another set
is_subset = set_a.issubset({1, 3, 4, 5})
print("Is set_a a subset of {1, 3, 4, 5}?", is_subset)

# 12. issuperset() - Check if set_a is a superset of another set
is_superset = set_a.issuperset({3, 4})
print("Is set_a a superset of {3, 4}?", is_superset)

# 13. isdisjoint() - Check if two sets have no common elements
is_disjoint = set_a.isdisjoint({7, 8})
print("Is set_a disjoint with {7, 8}?", is_disjoint)

```

```

After add(6): {1, 2, 3, 4, 5, 6}
After remove(2): {1, 3, 4, 5, 6}
After discard(10): {4, 5, 6, 7, 8}
Popped element: 1
After pop(): {3, 4, 5, 6}
After clear(): set()
Copy of set_a: {3, 4, 5, 6}
Union of set_a and {6, 7, 8}: {3, 4, 5, 6, 7, 8}
Intersection of set_a and {4, 5, 6}: {4, 5, 6}
Difference between set_a and {4, 5, 6}: {3}
Symmetric difference between set_a and {4, 5, 6}: {3}
Is set_a a subset of {1, 3, 4, 5}? False
Is set_a a superset of {3, 4}? True
Is set_a disjoint with {7, 8}? True

```

```

[2]: # Sample string for demonstration
my_string = " Hello, World! "

# 1. strip() - Removes leading and trailing whitespaces
stripped_string = my_string.strip()
print("After strip:", stripped_string) # Output: "Hello, World!"

# 2. lower() - Converts all characters to lowercase
lowercase_string = my_string.lower()
print("Lowercase:", lowercase_string) # Output: " hello, world! "

# 3. upper() - Converts all characters to uppercase

```

```

uppercase_string = my_string.upper()
print("Uppercase:", uppercase_string) # Output: " HELLO, WORLD! "

# 4. title() - Capitalizes the first letter of each word
title_string = my_string.title()
print("Title:", title_string) # Output: " Hello, World! "

# 5. capitalize() - Capitalizes the first letter and makes others lowercase
capitalized_string = my_string.capitalize()
print("Capitalized:", capitalized_string) # Output: " hello, world! "

# 6. replace() - Replaces occurrences of 'World' with 'Python'
replaced_string = my_string.replace("World", "Python")
print("After replace:", replaced_string) # Output: " Hello, Python! "

# 7. split() - Splits the string into a list by the delimiter (comma here)
split_string = my_string.split(",")
print("After split:", split_string) # Output: [' Hello', ' World! ']

# 8. find() - Returns the index of the first occurrence of the substring
↳ (returns -1 if not found)
find_index = my_string.find("World")
print("Index of 'World':", find_index) # Output: 8

# 9. count() - Counts how many times a substring appears in the string
count_substring = my_string.count("o")
print("Count of 'o':", count_substring) # Output: 2

# 10. isalpha() - Checks if all characters in the string are alphabetic
is_alpha = "Hello".isalpha()
print("'Hello' is alphabetic:", is_alpha) # Output: True

# 11. isnumeric() - Checks if all characters in the string are numeric
is_numeric = "12345".isnumeric()
print("'12345' is numeric:", is_numeric) # Output: True

# 12. startswith() - Checks if the string starts with the specified prefix
starts_with_hello = my_string.startswith("Hello")
print("Starts with 'Hello':", starts_with_hello) # Output: False (due to
↳ leading spaces)

# 13. endswith() - Checks if the string ends with the specified suffix
ends_with_world = my_string.endswith("World!")
print("Ends with 'World!':", ends_with_world) # Output: False (due to trailing
↳ spaces)

# 14. isdigit() - Checks if all characters in the string are digits

```

```

is_digit = "123".isdigit()
print("'123' is digit:", is_digit) # Output: True

# 15. join() - Joins elements of an iterable (list) into a string with a
↪separator
my_list = ["Hello", "World"]
joined_string = " ".join(my_list)
print("After join:", joined_string) # Output: "Hello World"

```

```

After strip: Hello, World!
Lowercase:  hello, world!
Uppercase:  HELLO, WORLD!
Title:     Hello, World!
Capitalized:  hello, world!
After replace:  Hello, Python!
After split: [' Hello', ' World! ']
Index of 'World': 9
Count of 'o': 2
'Hello' is alphabetic: True
'12345' is numeric: True
Starts with 'Hello': False
Ends with 'World!': False
'123' is digit: True
After join: Hello World

```

```

[3]: def calculate_bill():
    print("Welcome to the Bill Calculation System!")

    # User input for items and prices
    items = int(input("Enter number of items: "))
    total_amount = 0

    for _ in range(items):
        price = float(input("Enter price of item: $"))
        total_amount += price

    # User input for discount and tax
    discount = float(input("Enter discount percentage: "))
    tax = float(input("Enter tax percentage: "))

    # Calculate discount and tax
    discount_amount = (discount / 100) * total_amount
    tax_amount = (tax / 100) * (total_amount - discount_amount)

    # Final amount after discount and adding tax
    final_amount = total_amount - discount_amount + tax_amount

```

```
# Print the bill
print("\n--- Bill Summary ---")
print(f"Total amount: ${total_amount}")
print(f"Discount ({discount}%): -${discount_amount}")
print(f"Tax ({tax}%): +${tax_amount}")
print(f"Final amount to pay: ${final_amount}")

# Run the function
calculate_bill()
```

Welcome to the Bill Calculation System!

Enter number of items: 2

Enter price of item: \$ 12

Enter price of item: \$ 12

Enter discount percentage: 12

Enter tax percentage: 12

--- Bill Summary ---

Total amount: \$24.0

Discount (12.0%): -\$2.88

Tax (12.0%): +\$2.5344

Final amount to pay: \$23.654400000000003

[]: