

Revisiting Python Concepts for Basic Set Operations & Visualization

Lesson Objectives

By the end of this lesson, students will:

- Recall Python basics needed for working with **sets**.
 - Understand how to **create sets, add/remove elements, and perform operations**.
 - Learn how to **use functions, conditionals, and loops** in the project.
 - Get exposure to **visualization with matplotlib** for drawing Venn diagrams.
-

1. Introduction

We already know Python has **lists, tuples, and dictionaries**. Another important data type is the **Set**, which helps us deal with unique elements and perform mathematical set operations (union, intersection, difference, complement).

This project requires:

- **Set Creation Tool**
- **Visual Operation Performer**
- **Cardinality Calculator**

2. Python Concepts to Revisit

a) Set Basics

- A set is defined using `{ }` or the `set()` function.

```
A = {1, 2, 3, 4}
B = set([3, 4, 5, 6])
```

- Properties:
 - No duplicate elements
 - Unordered

b) Adding & Removing Elements

```
A.add(5)          # Adds element
A.remove(2)        # Removes element (error if not found)
A.discard(10)      # Removes element if exists, no error
```

c) Mathematical Set Operations

```
union_set = A | B          # Union
intersection_set = A & B    # Intersection
difference_set = A - B      # Difference
symmetric_set = A ^ B      # Symmetric Difference
```

d) Cardinality (Size of a Set)

```
len(A)    # Returns number of elements
```

e) Functions for Reusability

```
def perform_union(set1, set2):
    return set1 | set2
```

f) Loops & Conditionals

- **Loops** help iterate through sets.

```
for item in A:
    print(item)
```

- **Conditionals** help in checking membership.

```
if 3 in A:
    print("3 is in A")
```

g) Visualization with matplotlib

- We can use `matplotlib_venn` library to draw Venn diagrams.

```
from matplotlib_venn import venn2
import matplotlib.pyplot as plt

venn2([A, B], set_labels=("Set A", "Set B"))
plt.show()
```