Python

Full stack Skills Bootcamp

Introducing Python Sets

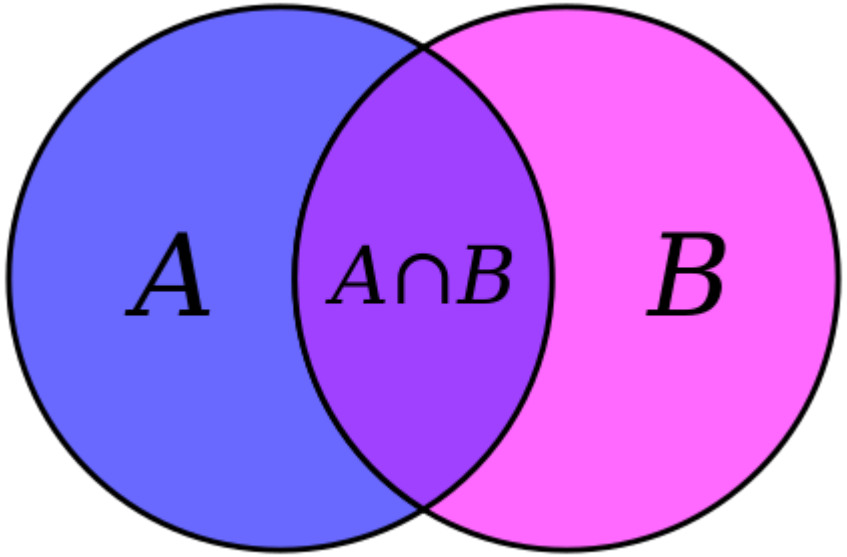
What are Sets?

Python Sets are unordered collection of unique elements. Unlike lists or tuples, sets:

- Do not allow duplicate values (each element must be unique)
- Are mutable, meaning that elements can be added or removed after the set is created

Why use sets?

- Sets are useful for performing mathematical set operations such as union, intersection and difference.



Creating Sets in Python

- Code Example

```python
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
```

  - Sets are defined using curly braces {} and can contain elements like numbers or strings.
  - Sets automatically discard duplicate values, ensuring that all elements are unique. For example, if {1, 2, 2, 3} is entered, the resulting set will be {1, 2, 3}

- Key point

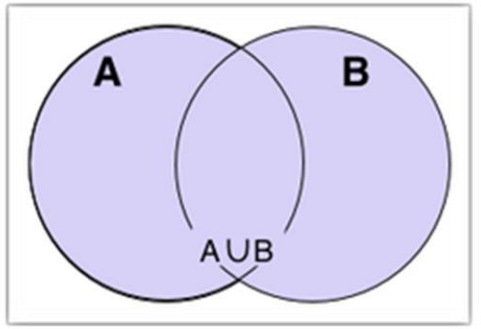
  - Unlike lists and tuples, sets don’t guarantee order, so their elements may not be stored or displayed in the order in which you define them.

Union of Sets

What is Union ?

Union combines elements from both sets, removing any duplicates. This operation is helpful when you want to merge two datasets while ensuring that repeated values appear only once.

```python
union_set = set1 | set2
print(union_set)  # Output: {1, 2, 3, 4, 5, 6, 7, 8}
```



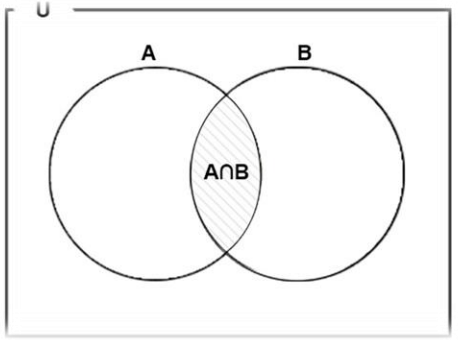
- The | operator is used to perform a union of two sets.
- This operation returns a new set that contains all the elements from both sets, without any duplicates.
- Union is great when working with datasets where you need all unique elements from multiple sources, such as merging customer lists without duplicates.

Intersection of Sets

What is Intersection ?

Intersection returns only the elements that are common between two sets. It's used when you want to find commonalities between different datasets.

```python
intersection_set = set1 & set2
print(intersection_set)  # Output: {4, 5}
```



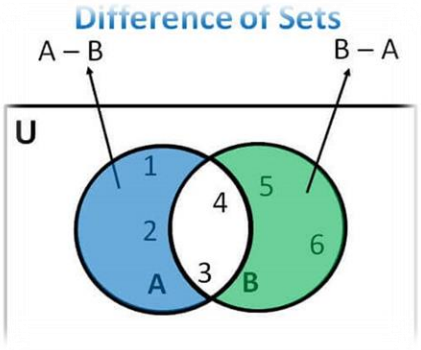
- The & operator is used to perform an intersection of two sets.
- This operation returns a new set containing only the elements that are found in both sets.
- Intersection is often used when comparing two datasets to find common elements, such as identifying customers who made purchases in both Store A and Store B.

Difference of Sets

What is Difference ?

The difference operation returns elements that are in the first set but not in the second. It's used when you need to filter out elements that exist in another dataset.

```python
difference_set = set1 - set2
print(difference_set)  # Output: {1, 2, 3}
```



- The - operator is used to perform the difference operation.
- This operation creates a new set containing the elements that are in set1 but not in set2.
- Difference is useful when you want to exclude certain items, such as identifying customers who only shopped at Store A and not at Store B.

Adding and Removing of Sets

- ## Adding Elements:

  You can add elements to a set using the .add() method. This allows you to dynamically expand your set with new unique items.

```python
set1.add(9)
print(set1)  # Output: {1, 2, 3, 4, 5, 9}
```

- ## Removing Elements:

:

```python
set1.remove(9)
print(set1)  # Output: {1, 2, 3, 4, 5}
```

  To remove an element, use the .remove() method. Be cautious because removing an element that doesn't exist will raise an error.

Conclusion

- Key Points
  - Python sets are versatile data structures that allow for the storage of unique elements.
  - Sets are ideal for use cases like removing duplicates from a collection of items, comparing datasets to find common entries, and performing membership tests quickly.
  - Keep exploring Python sets and try incorporating them into your coding projects to enhance your data management capabilities!.