

Sets

Tushar B. Kute,
<http://tusharkute.com>



Set

- A set is a collection of unique data. That is, elements of a set cannot be duplicate. For example,
- Suppose we want to store information about student IDs. Since student IDs cannot be duplicate, we can use a set.



Set of Student ID

Creating as set

- # create a set of integer type
- `student_id = {112, 114, 116, 118, 115}`
- `print('Student ID:', student_id)`

- # create a set of string type
- `vowel_letters = {'a', 'e', 'i', 'o', 'u'}`
- `print('Vowel Letters:', vowel_letters)`

- # create a set of mixed data types
- `mixed_set = {'Hello', 101, -2, 'Bye'}`
- `print('Set of mixed data types:', mixed_set)`

Creating empty set

- # create an empty set
- `empty_set = set()`

- # create an empty dictionary
- `empty_dictionary = {}`

- # check data type of empty_set
- `print('Data type of empty_set:', type(empty_set))`

- # check data type of dictionary_set
- `print('Data type of empty_dictionary', type(empty_dictionary))`

Duplicate items in a set

- `numbers = {2, 4, 6, 6, 2, 8}`
- `print(numbers) # {8, 2, 4, 6}`

Add items in a set

- `numbers = {21, 34, 54, 12}`
- `print('Initial Set:', numbers)`
- `# using add() method`
- `numbers.add(32)`
- `print('Updated Set:', numbers)`

Update the set

- `companies = {'Lacoste', 'Ralph Lauren'}`
- `tech_companies = ['apple', 'google', 'apple']`
- `companies.update(tech_companies)`
- `print(companies)`

Remove element from the set

- `languages = {'Swift', 'Java', 'Python'}`
- `print('Initial Set:', languages)`
- `# remove 'Java' from a set`
- `removedValue = languages.discard('Java')`
- `print('Set after remove():', languages)`

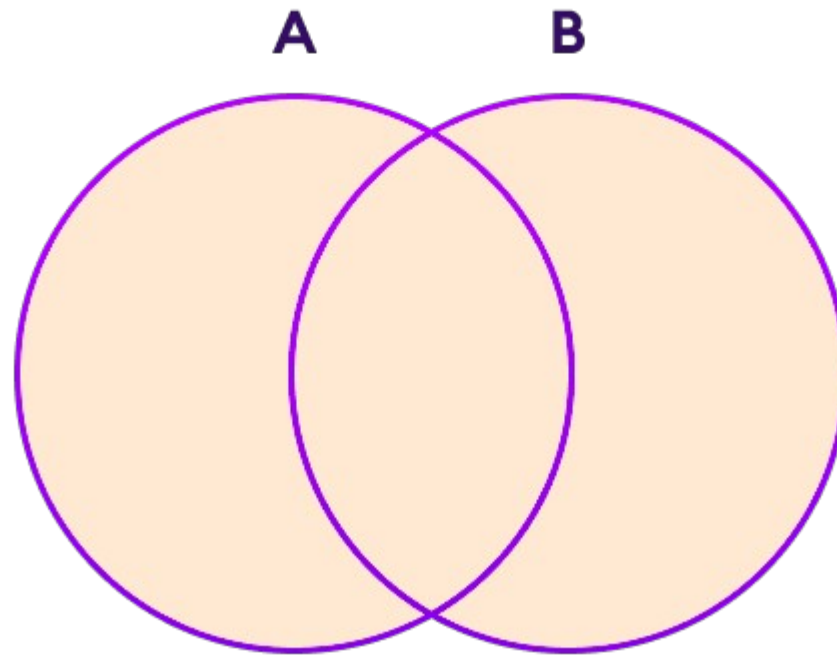
Functions

- `all()`
- `any()`
- `enumerate()`
- `len()`
- `max()`
- `min()`
- `sum()`
- `sorted()`

Iterations

- `fruits = {"Apple", "Peach", "Mango"}`
- `#` for loop to access each fruits
- `for fruit in fruits:`
 `print(fruit)`

Set Operations: Union



Set Operation: Union

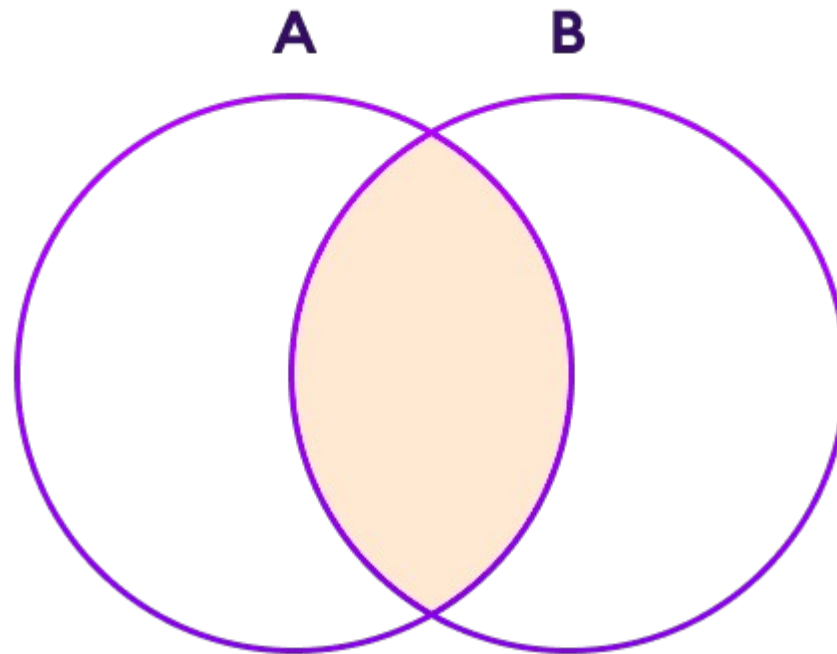
- # first set
- $A = \{1, 3, 5\}$

- # second set
- $B = \{0, 2, 4\}$

- # perform union operation using |
- `print('Union using |:', A | B)`

- # perform union operation using union()
- `print('Union using union():', A.union(B))`

Set Operation: Intersection



Set Operation: Intersection

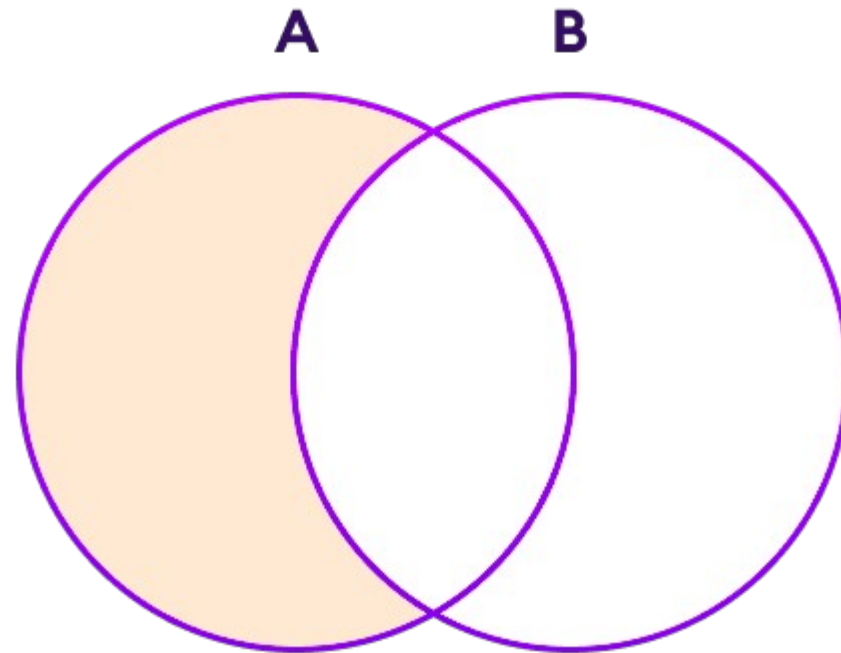
- # first set
- $A = \{1, 3, 5\}$

- # second set
- $B = \{1, 2, 3\}$

- # perform intersection operation using &
- `print('Intersection using &:', A & B)`

- # perform intersection operation using intersection()
- `print('Intersection using intersection():', A.intersection(B))`

Set Operation: Difference



Set Operation: Difference

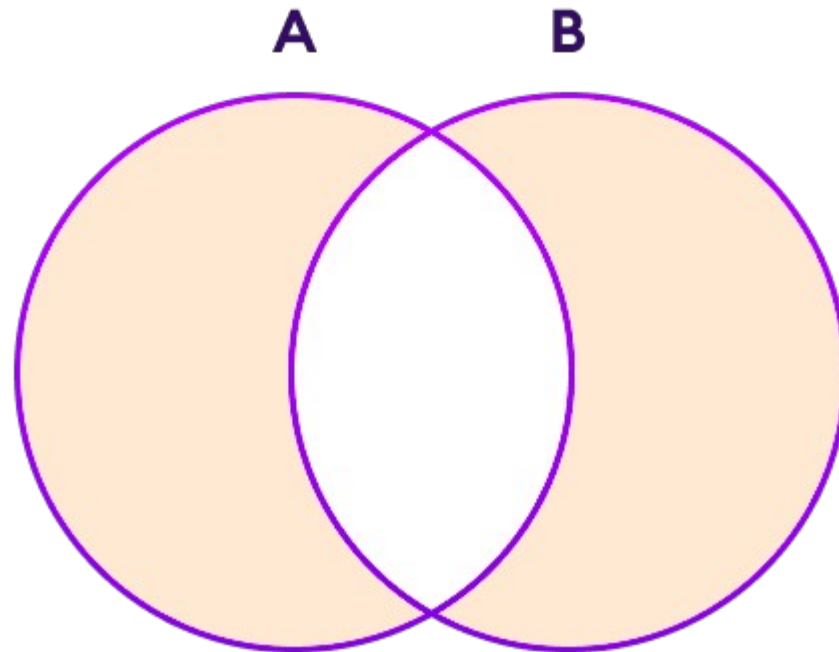
- # first set
- $A = \{2, 3, 5\}$

- # second set
- $B = \{1, 2, 6\}$

- # perform difference operation using &
- `print('Difference using &:', A - B)`

- # perform difference operation using `difference()`
- `print('Difference using difference():', A.difference(B))`

Set Operation: Symmetric Difference



Set Operation: Symmetric Difference

- # first set
- $A = \{2, 3, 5\}$

- # second set
- $B = \{1, 2, 6\}$

- # perform difference operation using &
- `print('using ^:', A ^ B)`

- # using `symmetric_difference()`
- `print('using symmetric_difference():', A.symmetric_difference(B))`

Checking for equality

- # first set
- $A = \{1, 3, 5\}$

- # second set
- $B = \{3, 5, 1\}$

- # perform difference operation using &
- if $A == B$:
 - print('Set A and Set B are equal')
- else:
 - print('Set A and Set B are not equal')

Thank you

This presentation is created using LibreOffice Impress 5.1.6.2, can be used freely as per GNU General Public License



@mitu_skillologies



/MITuSkillologies



@mitu_group



/company/mitu-
skillologies



MITUSkillologies

Web Resources

<https://mitu.co.in>

<http://tusharkute.com>

contact@mitu.co.in

tushar@tusharkute.com