Set Data Type

03 June 2025 16:04

- A **Set** is a collection of **homogeneous** or **heterogeneous** values.
  - **Homogeneous**: A collection of the same type of values.
    Example: 10, 20, 30, 40
  - **Heterogeneous**: A collection of different types of values.
    Example: 10, 2.5, 9+6j, True
- Set are enclosed within flower braces { }.

Q. How can we create Set?

Syntax: Variable name={value1 , value2 ,value3,…….Value n}

Note: Values inside a Tuple are separated by commas(',').

Q. What is the default value of Set?

- The default value of a Set is an empty parenthesis "set( )".
- An empty set() is considered **False** in a Boolean context.

Q. What types of values can we store inside a Set?

We can store values of any data type in a list, including:

- Integers
- Floats
- Complex
- Boolean

☹ We can't store **duplicate** values. If it store it will remove automatically.

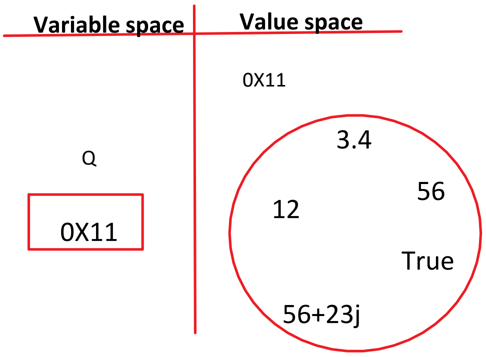
- It will store the data Randomly . It is unorder Collection.

Example: S={12,3.4,6+4j,98,True}
S
Output:- {True, 98, 3.4, (6+4j), 12}

Example 2: we are storing mutable data inside a set

```
t={10,[10,20],(10,30)}
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    t={10,[10,20],(10,30)}
TypeError: unhashable type: 'list'
```

Example: Q={12,3.4,56+23j,True,56,1,12}



Note:1. It will store data randomly and in the form of hash table.
2. in case of set it is not possible to access and modifying the data because those values doesn't contain any index.

```
Example: d={10,20}
         d[0]
        Output:-
        Traceback (most recent call last):
          File "<pyshell#1>", line 1, in <module>
            s[0]
        TypeError: 'set' object is not subscriptable
```

Modifying Set Elements:

- **To add a value to a set**, we use the inbuilt function add().

  Syntax: variable name. add(new value)

  ```
  Example:  Q={True, 3.4, 56, 12, (56+23j)}
            Q.add('py')
            Q
           Output:-{True, 3.4, 56, 'py', 12, (56+23j)}
  ```

- **To remove a specific value from a set**, we use the inbuilt function remove().

  Syntax: variable name. remove(value)
  Example:-

  ```
  Q={True, 3.4, 56, 'py', 12, (56+23j)}
  Q. remove(3.4)
  Q
  Output:  {True, 56, 'py', 12, (56+23j)}
  ```

- **To remove any element (usually the first one in internal order) from a set**, we use the inbuilt function pop().
  Note: Sets are unordered collections, so "first" does not mean the first element you see.[it will eliminate first value]

  Syntax : variable name. pop()

  Example:

  ```
  Q={True, 56, 'py', 12, (56+23j)}
  Q.pop()
  Output: True
  Q.
  Output: {56, 'py', 12, (56+23j)}
  ```