1. Set in Python

In Python, a set is an unordered collection of unique items. Defined using curly braces {}, sets are mutable, allowing for the addition and removal of elements. However, they don't support indexing or slicing due to their unordered nature. Sets are commonly used for operations like union, intersection, and difference.

Set Operations

1. Creating a set

```
In [2]: my_set = {1, 2, 3, 4, 5}
```

2. Adding an element

```
In [3]: my_set.add(6)
```

3. Removing an element

```
In [4]: my_set.discard(1)
```

4. Union of sets

```
In [5]: another_set = {5, 6, 7, 8}
        union_set = my_set.union(another_set)
```

5. Intersection of sets

```
In [6]: intersection_set = my_set.intersection(another_set)
```

```
In [7]: intersection_set

Out[7]: {5, 6}
```

2. Dictionary in Python

A dictionary in Python is a collection of key-value pairs, where each key is unique. Defined using curly braces {}, dictionaries allow for efficient data retrieval by key. They are mutable, meaning keys and values can be added or modified. The order of items is maintained as of Python 3.7, based on their insertion order.

Dictionary Operations

1 . Creating a dictionary

In [8]:
```
my_dict = {'a': 1, 'b': 2, 'c': 3}
```

2. Accessing a value by key

In [9]:
```
value_a = my_dict['a']
```

3. Setting a value by key

In [10]:
```
my_dict['d'] = 4
```

4. Removing a key-value pair

In [11]:
```
del my_dict['a']
```

5. Iterating through keys and values

In [12]:
```
for key, value in my_dict.items():
    print(key, value)
```

```
b 2
c 3
d 4
```

In [ ]: