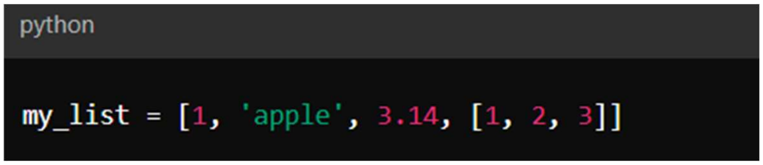
(Chapter- 06) Data Structures in Python

Data structures are essential for organizing and storing data efficiently, enabling quick access and modifications. Python provides several built-in data structures, which can be classified into:

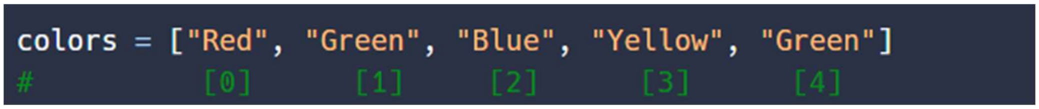
1. List 2. Tuple 3. Set 4. Dictionary

Python Lists: A list is a collection of ordered, mutable elements in Python. Lists can hold any data type (integers, floats, strings, objects, etc.)



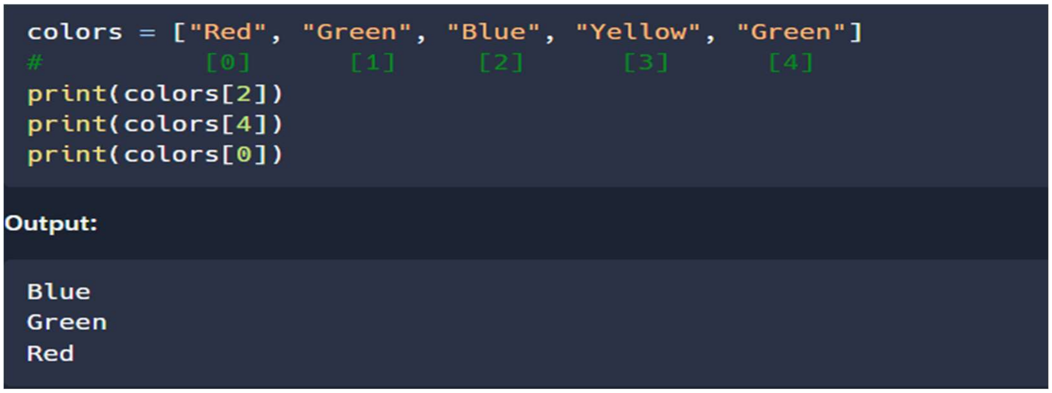
```python
my_list = [1, 'apple', 3.14, [1, 2, 3]]
```

List Index: Each item/element in a list has its own unique index. This index can be used to access any particular item from the list. The first item has index [0], second item has index [1], third item has index [2] and so on.



```
colors = ["Red", "Green", "Blue", "Yellow", "Green"]
#          [0]      [1]     [2]      [3]      [4]
```

- **Positive Indexing:** Start from 0 to positive hole number.

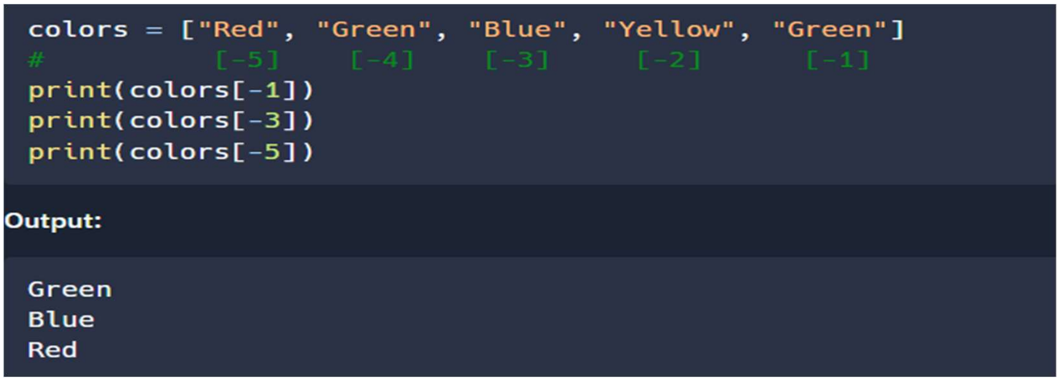


```
colors = ["Red", "Green", "Blue", "Yellow", "Green"]
#          [0]      [1]     [2]      [3]      [4]
print(colors[2])
print(colors[4])
print(colors[0])
```

Output:

```
Blue
Green
Red
```

- **Negative Indexing:** Similar to positive indexing, negative indexing is also used to access items, but from the end of the list. The last item has index [-1], second last item has index [-2], third last item has index [-3] and so on.

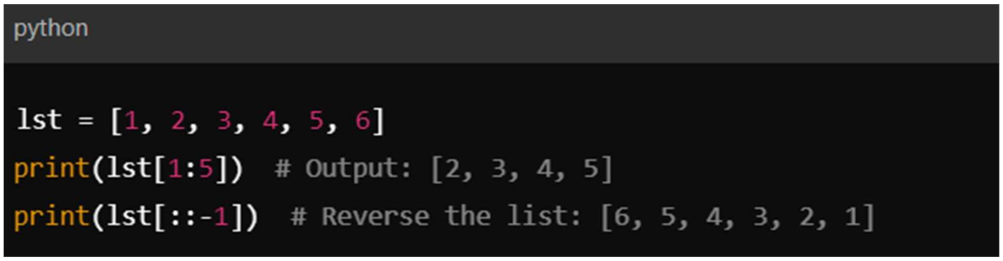


```
colors = ["Red", "Green", "Blue", "Yellow", "Green"]
#          [-5]    [-4]    [-3]     [-2]     [-1]
print(colors[-1])
print(colors[-3])
print(colors[-5])
```

Output:

```
Green
Blue
Red
```

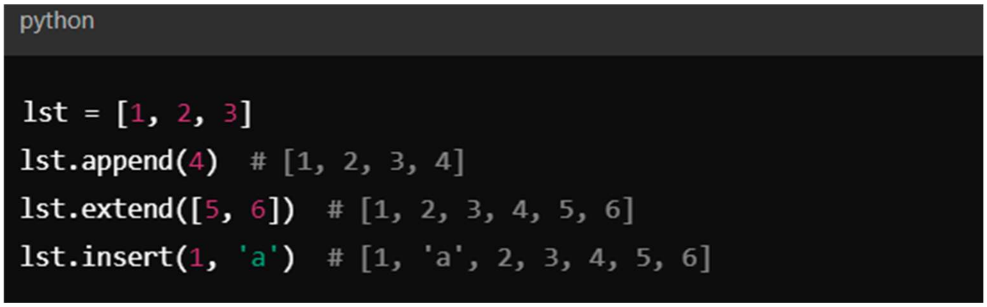
Slicing and Indexing: Slicing allows access to sub lists using the format **list[start: stop: step]**.



```python
lst = [1, 2, 3, 4, 5, 6]
print(lst[1:5])  # Output: [2, 3, 4, 5]
print(lst[::-1])  # Reverse the list: [6, 5, 4, 3, 2, 1]
```

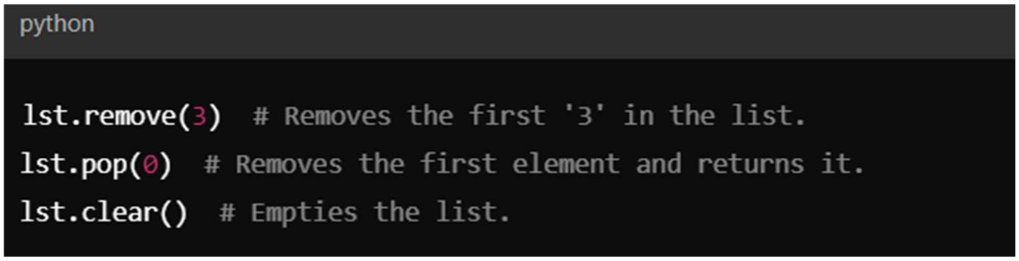
List Operations and Methods:

- ### Adding elements:
    - **append():** Adds a single element.
    - **extend():** Adds multiple elements from another list or iterable.
    - **insert():** Inserts an element at a specific index



```python
lst = [1, 2, 3]
lst.append(4)  # [1, 2, 3, 4]
lst.extend([5, 6])  # [1, 2, 3, 4, 5, 6]
lst.insert(1, 'a')  # [1, 'a', 2, 3, 4, 5, 6]
```

- ### Removing elements:
    - **remove():** Removes the first occurrence of the element.
    - **pop():** Removes and returns the element at the given index.
    - **clear():** Removes all elements from the list.



```python
lst.remove(3)  # Removes the first '3' in the list.
lst.pop(0)  # Removes the first element and returns it.
lst.clear()  # Empties the list.
```