Python Collections Cheat Sheet

1) LISTS (Ordered, Mutable, Allow Duplicates)

Use when you need an ordered collection you can modify.
Syntax: my_list = [1, 2, 3]
Key Operations:
• Append: my_list.append(x)
• Insert: my_list.insert(i, x)
• Remove: my_list.remove(x)
• Sort: my_list.sort()

2) TUPLES (Ordered, Immutable, Allow Duplicates)

Use when values should NOT change.
Syntax: my_tuple = (1, 2, 3)
Key Operations:
• Count: my_tuple.count(x)
• Index: my_tuple.index(x)

3) DICTIONARIES (Unordered, Mutable, Key-Value Pairs)

Use when mapping data: key → value.
Syntax: my_dict = {"name": "Alex", "age": 30}
Key Operations:
• Access: my_dict["name"]
• Add/Update: my_dict["city"] = "Delhi"
• Remove: my_dict.pop("age")
• Keys/Values: my_dict.keys(), my_dict.values()

4) SETS (Unordered, Mutable, Unique Items)

Use for fast membership tests and removing duplicates.
Syntax: my_set = {1, 2, 3}
Key Operations:
• Add: my_set.add(x)
• Remove: my_set.remove(x)
• Union: my_set | other_set
• Intersection: my_set & other_set

Quick Choosing Guide:

✔ Need order + changeable → List
✔ Need order + NOT changeable → Tuple
✔ Key-value mapping → Dictionary
✔ Unique items / fast lookup → Set

Cybersecurity Use Cases:

• List → storing URLs to scan
• Tuple → fixed IP-port pairs
• Dict → user → hashed password map
• Set → blocklists (unique IPs)