Python Cheat Sheet: Set, Frozenset and Dictionary

Set

A set is a collection of unique items. Duplicate values are automatically removed. Sets are unordered and mutable, which means you can add or remove items.

Example 1: Simple set
```
fruits = {"apple", "banana", "apple", "cherry"}
print(fruits) # Output: {'apple', 'banana', 'cherry'}
```
Example 2: Adding and removing items
```
fruits.add("orange") # Adds 'orange'
fruits.remove("banana") # Removes 'banana'
fruits.discard("mango") # No error if 'mango' does not exist
print(fruits)
```
Example 3: Set operations
```
setA = {1, 2, 3, 4}
setB = {3, 4, 5, 6}
print(setA | setB) # Union: {1, 2, 3, 4, 5, 6}
print(setA & setB) # Intersection: {3, 4}
print(setA - setB) # Difference: {1, 2}
print(setA ^ setB) # Symmetric difference: {1, 2, 5, 6}
```
Example 4: Removing duplicates from a list
```
numbers = [1, 2, 2, 3, 4, 4, 5]
unique_numbers = set(numbers)
print(unique_numbers) # Output: {1, 2, 3, 4, 5}
```
Example 5: Finding common students in two sets
```
math_students = {"Ali", "Sara", "Tom"}
science_students = {"Tom", "Lina", "Sara"}
common = math_students & science_students
print("Students in both subjects:", common) # Output: {'Tom', 'Sara'}
```

Frozenset

A frozenset is an immutable version of a set. You cannot add or remove items, but you can perform set operations like union and intersection. Frozensets can also be used as dictionary keys.

Example 1: Creating a frozenset
```
frozen_colors = frozenset(["red", "green", "blue"])
print(frozen_colors)
```
Example 2: Set operations with frozenset
```
A = frozenset([1, 2, 3])
B = frozenset([3, 4, 5])
print(A | B) # Union: {1, 2, 3, 4, 5}
print(A & B) # Intersection: {3}
```
Example 3: Using frozenset as a dictionary key
```
scores = {frozenset(["math", "science"]): 95}
print(scores)
```

Dictionary

A dictionary stores data in key-value pairs. Keys are unique and immutable, while values can be of any data type. Dictionaries are mutable and unordered.

Example 1: Creating and accessing dictionary
```
student = {"name": "John", "age": 10, "grade": "5th"}
print(student["name"]) # Output: John
```
Example 2: Adding, updating, and deleting
```
student["school"] = "Sunrise High" # Add
student["age"] = 11 # Update
del student["grade"] # Delete
print(student)
```
Example 3: Looping through dictionary
```
for key, value in student.items():
print(key, ":", value)
```
Example 4: Counting words in a sentence
```
sentence = "apple banana apple cherry banana apple"
words = sentence.split()
count = {}
for w in words:
count[w] = count.get(w, 0) + 1
print(count) # Output: {'apple': 3, 'banana': 2, 'cherry': 1}
```
Example 5: Nested dictionary and average marks
```
students = {
"A001": {"name": "Ali", "marks": {"math": 90, "science": 85}},
"A002": {"name": "Sara", "marks": {"math": 95, "science": 92}}
```

```
}
for id, info in students.items():
avg = (info["marks"]["math"] + info["marks"]["science"]) / 2
print(f"{info['name']}'s average: {avg}")
# Output:
# Ali's average: 87.5
# Sara's average: 93.5
```

Summary Table

Feature	Set	Frozenset	Dictionary
Mutable	Yes	No	Yes
Removes duplicates	Yes	Yes	Keys unique
Indexed?	No	No	By keys
Can be dict key?	No	Yes	Yes (if key immutable)
Used for	Unique items	Immutable sets	Key-value storage

Practice Questions with Answers

Question 1: Create a set of numbers with duplicates and remove duplicates.
Answer:
```
numbers = [1, 2, 2, 3, 4, 4, 5]
unique_numbers = set(numbers)
print(unique_numbers) # Output: {1, 2, 3, 4, 5}
```

Question 2: Use a frozenset as a dictionary key.
Answer:
```
key_set = frozenset([1, 2, 3])
my_dict = {key_set: "value"}
print(my_dict) # Output: {frozenset({1, 2, 3}): 'value'}
```

Question 3: Make a dictionary of students with marks and find who scored highest.
Answer:
```
students = {"Ali": 90, "Sara": 95, "Tom": 85}
highest = max(students, key=students.get)
print(highest) # Output: Sara
```

Question 4: Combine two sets and find common elements.
Answer:
```
setA = {1, 2, 3}
setB = {2, 3, 4}
common = setA & setB
print(common) # Output: {2, 3}
```

Question 5: Count how many times each fruit appears in a list.
Answer:
```
fruits = ["apple", "banana", "apple", "cherry"]
count = {}
for f in fruits:
count[f] = count.get(f, 0) + 1
print(count) # Output: {'apple': 2, 'banana': 1, 'cherry': 1}
```