Python Dictionaries and Sets

Dictionary Comprehensions

```
# Basic dictionary comprehension
square_dict = {x: x**2 for x in range(6)}
print("Squares:", square_dict)

# Filtered dictionary
filtered_dict = {k: v for k, v in square_dict.items() if v > 10}
print("Filtered (>10):", filtered_dict)

# Dictionary from two lists
keys = ['name', 'age', 'city']
values = ['Alice', 25, 'NYC']
person = {k: v for k, v in zip(keys, values)}
print("\nPerson dict:", person)

# Transform dictionary values
prices = {'apple': 1.0, 'banana': 0.5, 'orange': 0.75}
discounted = {k: v * 0.9 for k, v in prices.items()}
print("\nDiscounted prices:", discounted)
```

Set Operations

```
set_a = {1, 2, 3, 4, 5}
set_b = {4, 5, 6, 7, 8}

print("Set A:", set_a)
print("Set B:", set_b)

# Union (all elements)
print("\nUnion:", set_a | set_b)

# Intersection (common elements)
print("Intersection:", set_a & set_b)

# Difference (in A but not in B)
print("Difference (A-B):", set_a - set_b)

# Symmetric difference (in either but not both)
print("Symmetric diff:", set_a ^ set_b)

# Subset and superset
set_c = {1, 2, 3}
print("\nIs {1,2,3} subset of A?", set_c.issubset(set_a))
print("Is A superset of {1,2,3}?", set_a.issuperset(set_c))
```

Advanced Dictionary Operations

```
# Merging dictionaries (Python 3.9+)
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}
merged = dict1 | dict2
print("Merged:", merged)

# Dictionary with default values
from collections import defaultdict
word_count = defaultdict(int)
text = "hello world hello"
```

```
for word in text.split():
    word_count[word] += 1
print("\nWord count:", dict(word_count))

# Inverting dictionary
original = {'a': 1, 'b': 2, 'c': 3}
inverted = {v: k for k, v in original.items()}
print("\nOriginal:", original)
print("Inverted:", inverted)
```

Set Comprehensions

```
# Basic set comprehension
squares = {x**2 for x in range(10)}
print("Squares set:", squares)

# Filtered set comprehension
evens = {x for x in range(20) if x % 2 == 0}
print("\nEven numbers:", evens)

# From string
unique_chars = {char.lower() for char in "Hello World" if char.isalpha()}
print("\nUnique characters:", unique_chars)

# Find duplicates using sets
numbers = [1, 2, 3, 2, 4, 1, 5, 3]
duplicates = {x for x in numbers if numbers.count(x) > 1}
print("\nDuplicates:", duplicates)
```

Practical Applications

```
# Count unique elements
data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
unique_count = len(set(data))
print("Unique elements:", unique_count)

# Remove duplicates while preserving order
def remove_duplicates(seq):
    seen = set()
    return [x for x in seq if not (x in seen or seen.add(x))]

numbers = [1, 2, 3, 2, 4, 1, 5]
print("\nOriginal:", numbers)
print("No duplicates:", remove_duplicates(numbers))

# Find common elements in multiple lists
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]
list3 = [5, 6, 7, 8, 9]
common = set(list1) & set(list2) & set(list3)
print("\nCommon to all:", common)
```