Day 3 – Python Data Structures: Extensive Detailed Guide

Trainer: Your Name – Senior Data Scientist

This document thoroughly covers Python's built-in data structures with extensive examples, explanations, and practical usage.

# 1. Python Lists

Lists are mutable sequences used extensively in Python.

## Creating Lists

numbers = [1, 2, 3, 4, 5]

## Accessing and Modifying Lists

numbers[0] = 10  
numbers.append(6)  
numbers.remove(2)  
popped = numbers.pop()

## List Comprehensions

squares = [x\*\*2 for x in range(10)]

# 2. Python Tuples

Tuples are immutable ordered sequences.

## Creating Tuples

location = (40.7128, -74.0060)

## Tuple Unpacking

lat, lon = location

# 3. Python Sets

Sets are unordered unique collections.

## Creating Sets and Set Operations

a = {1, 2, 3}  
b = {3, 4, 5}  
union = a | b  
intersection = a & b  
difference = a - b

# 4. Python Dictionaries

Dictionaries are collections of key-value pairs.

## Creating and Manipulating Dictionaries

user = {'name': 'John', 'age': 30}  
user['email'] = 'john@example.com'  
del user['age']

# 5. Real-world Applications

## Lists in Task Management

tasks = ['Email clients', 'Prepare report', 'Team meeting']  
tasks.append('Schedule review')  
completed\_task = tasks.pop(0)

## Tuples for Geographic Data

locations = [('NY', (40.7128, -74.0060)), ('SF', (37.7749, -122.4194))]

## Sets for Data Deduplication

raw\_data = [1, 2, 2, 3, 4, 4]  
unique\_data = set(raw\_data)

## Dictionaries for JSON Data Handling

import json  
data = json.loads('{"name": "Alice", "age": 25}')  
data['city'] = 'New York'

# Bonus Section: 20 Practical Coding Examples

This section provides practical coding problems and examples using Python data structures, loops, and conditionals.

## Example 1: Filter even numbers from a list

numbers = [1, 2, 3, 4, 5, 6]  
evens = [num for num in numbers if num % 2 == 0]  
print(evens)

## Example 2: Remove duplicates from a list

data = [1, 2, 2, 3, 4, 4, 5]  
unique\_data = list(set(data))  
print(unique\_data)

## Example 3: Reverse a list using a loop

items = ['a', 'b', 'c', 'd']  
reversed\_items = []  
for i in range(len(items)-1, -1, -1):  
 reversed\_items.append(items[i])  
print(reversed\_items)

## Example 4: Create a new list of squares if number > 3

nums = [1, 2, 3, 4, 5]  
squares = [n\*\*2 for n in nums if n > 3]  
print(squares)

## Example 5: Unpack and process tuples in a list

coordinates = [(1, 2), (3, 4), (5, 6)]  
for x, y in coordinates:  
 print(f"Sum: {x + y}")

## Example 6: Find max value in a tuple

values = (10, 25, 17, 30)  
print("Max value:", max(values))

## Example 7: Check if a value exists in a tuple

colors = ('red', 'green', 'blue')  
if 'green' in colors:  
 print("Green is available.")

## Example 8: Find common elements using intersection

a = {1, 2, 3, 4}  
b = {3, 4, 5, 6}  
common = a & b  
print(common)

## Example 9: Filter unique words from a sentence

sentence = "the cat and the hat"  
unique\_words = set(sentence.split())  
print(unique\_words)

## Example 10: Check if a set is a subset

a = {1, 2}  
b = {1, 2, 3, 4}  
print(a.issubset(b))

## Example 11: Count word frequencies

words = ["apple", "banana", "apple", "orange", "banana"]  
count = {}  
for word in words:  
 count[word] = count.get(word, 0) + 1  
print(count)

## Example 12: Filter students with grade > 80

grades = {"Alice": 90, "Bob": 75, "Charlie": 85}  
high\_achievers = {name: score for name, score in grades.items() if score > 80}  
print(high\_achievers)

## Example 13: Convert dictionary to a list of tuples

info = {"name": "John", "age": 25}  
tuple\_list = list(info.items())  
print(tuple\_list)

## Example 14: Update values conditionally in dict

prices = {"apple": 50, "banana": 20}  
for fruit in prices:  
 if prices[fruit] < 30:  
 prices[fruit] += 5  
print(prices)

## Example 15: Filter products with price > 100

products = [{"name": "Phone", "price": 299}, {"name": "Pen", "price": 20}]  
filtered = [p for p in products if p["price"] > 100]  
print(filtered)

## Example 16: Flatten a list of tuples into one list

pairs = [(1, 2), (3, 4), (5, 6)]  
flat = [x for tup in pairs for x in tup]  
print(flat)

## Example 17: Group words by their length using dict

words = ["hi", "hello", "sun", "world"]  
grouped = {}  
for word in words:  
 length = len(word)  
 grouped.setdefault(length, []).append(word)  
print(grouped)

## Example 18: Create dictionary from list of tuples

data = [("a", 1), ("b", 2), ("c", 3)]  
result = dict(data)  
print(result)

## Example 19: Loop through nested dictionary

students = {  
 "Alice": {"math": 85, "science": 90},  
 "Bob": {"math": 70, "science": 60}  
}  
for name, subjects in students.items():  
 for subject, mark in subjects.items():  
 print(f"{name} scored {mark} in {subject}")

## Example 20: Inventory summary from list

inventory = [  
 {"item": "pen", "qty": 10},  
 {"item": "pencil", "qty": 5},  
 {"item": "pen", "qty": 3}  
]  
  
summary = {}  
for entry in inventory:  
 item = entry["item"]  
 summary[item] = summary.get(item, 0) + entry["qty"]  
  
print(summary)