Day 1: Introduction to Python

Learning Objectives:

- What is Python? High-level, interpreted, versatile language.
- Where Python is used: Web, Data Science, AI, Automation.
- Why learn Python: Easy syntax, huge community support.
- First program: print('Hello World').
- Task: Write a program that prints your name.
- IDE choices: VS Code, PyCharm, IDLE.

Day 1: Introduction to Python - Code Examples

- # Print your first message
- print("Hello, Python!")

Day 1: Introduction to Python - Practice Task

- Main: Print your name using print().
- Challenge: Print your name on one line and your favorite color on the next line

Day 1: Introduction to Python - Summary

- Python is beginner-friendly and widely used.
- print() is used to display output.
- Python uses indentation, not braces, to define code blocks.

Day 2: Variables & Data Types

Learning Objectives:

- Variables: Containers for storing data.
- Data types: int, float, str, bool.
- Example: age = 25, pi = 3.14, name = 'Alice', is_active = True.
- Type casting: int(), float(), str().
- Task: Create variables for name, age, and GPA.

Day 2: Variables & Data Types - Code Examples

- # Assigning variables
- name = "Alice"
- age = 25
- height = 5.6

- print(name, age, height)

Day 2: Variables & Data Types - Practice Task

- Main: Create variables for your name, age, and favorite hobby, then print them.
- Challenge: Change one variable's value and print it again.

Day 2: Variables & Data Types - Summary

- Variables store data.
- Common types: int, float, str, bool.
- Variables can be reassigned.

Day 3: Strings

Learning Objectives:

- String creation and printing.
- String indexing and slicing.
- String methods: upper(), lower(), replace(), split().
- f-strings for formatting.
- Task: Take a name and print it in reverse.

Day 3: Strings - Code Examples

- # String operations
- message = "Python is fun"
- print(message.upper())
- print(message.lower())
- print(message.replace("fun", "awesome"))

Day 3: Strings - Practice Task

- Main: Store a sentence and print it in uppercase.
- Challenge: Replace a word in your sentence.

Day 3: Strings - Summary

- Strings are sequences of characters.
- .upper(), .lower(), .replace() modify string display.
- Strings are immutable.

Day 4: Numbers & Math

Learning Objectives:

- Integers and floats.
- Basic operators: +, -, *, /, //, %, **.
- Order of operations (PEMDAS).
- Math functions: abs(), round(), pow().
- Task: Calculate area of a circle.

Day 4: Numbers & Math - Code Examples

- a = 10
- b = 3
- print(a + b, a - b, a * b, a / b)
- print(a // b, a % b, a ** b) # floor division, modulus, exponent

Day 4: Numbers & Math - Practice Task

- Main: Create two numbers and perform all arithmetic operations.
- Challenge: Calculate the area of a rectangle.

Day 4: Numbers & Math - Summary

- Python supports +, -, *, /, //, %, **.
- Integer division (//) returns whole numbers.
- % gives remainder.

Day 5: Lists

Learning Objectives:

- List creation: [1, 2, 3].
- Indexing, slicing, and updating lists.
- List methods: append(), remove(), sort(), reverse().
- Looping through lists.
- Task: Store 5 favorite movies in a list and print them.

Day 5: Lists - Code Examples

```
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
fruits.remove("banana")
print(fruits)
```

Day 5: Lists - Practice Task

- Main: Create a list of 3 favorite movies and add 1 more.
- Challenge: Sort the list alphabetically.

Day 5: Lists - Summary

- Lists store multiple items in one variable.
- Lists are mutable.
- Common methods: .append(), .remove(), .sort().

Day 6: Tuples

Learning Objectives:

- Tuples: Immutable sequences.
- Tuple creation: (1, 2, 3).
- Tuple unpacking.
- When to use tuples vs lists.
- Task: Store coordinates in a tuple.

Day 6: Tuples - Code Examples

```
# Tuple
coordinates = (10, 20)
print(coordinates)

# Set
colors = {"red", "green", "blue"}
colors.add("yellow")
print(colors)
```

Day 6: Tuples - Practice Task

- Main: Create a tuple for your birth date (day, month, year).
- Challenge: Create a set of fruits and add a new one.

Day 6: Tuples - Summary

- Tuples are immutable.
- Sets store unique items.
- .add() adds items to sets.

Day 7: Sets

Learning Objectives:

- Sets: Unordered collections of unique items.
- Set creation: {1, 2, 3}.
- Set methods: add(), remove(), union(), intersection().
- Use cases: Removing duplicates.
- Task: Create a set of unique characters from a string.

Day 7: Sets - Code Examples

- person = {"name": "Alice", "age": 25}
- person["city"] = "New York"
- print(person)

Day 7: Sets - Practice Task

- Main: Create a dictionary for yourself with name, age, and hobby.
- Challenge: Update the hobby value.

Day 7: Sets - Summary

- Dictionaries store data as key-value pairs.
- Keys must be unique.
- .update() and direct assignment modify values.