Lesson 2: Syntax and Your First App

◇ Introduction

In this lesson, we're going to dive into the core of Python's syntax and build your first Python application!
By the end of this lesson, you'll:

- Understand how to write and run Python code.
- Learn about the basic syntax rules of Python.
- Create a simple application using variables, input/output, and basic logic.

◇ Python Syntax: A Quick Overview

Python syntax is known for being clean and easy to read. Let's break down the basic rules:

1. Case Sensitivity
   Python is case-sensitive, meaning that myvariable and MyVariable are different variables.
2. Indentation
   Unlike many other programming languages that use braces {} to define blocks of code, Python uses indentation (spaces or tabs). Indentation helps define which lines of code belong to loops, functions, and classes.
   - Example:
     ```
     if True:
         print("Hello, Python!")
     ```
3. Comments
   Comments in Python are created by using the # symbol. Python ignores anything after the # on a line.
   ```
   # This is a comment
   print("Hello, World!")  # This prints text
   ```
4. Variables and Data types
   Python is dynamically typed, meaning you don't have to explicitly declare a variable type. Python automatically detects it.
   ```
   my_number = 10      # Integer
   my_name = "John"    # String
   my_pi = 3.14        # Float
   ```
5. Print Statement
   Use print() to display output in Python.
   ```
   print("Hello, World!")
   ```
6. Basic Operators
   Python supports all the basic arithmetic operators:
   - + (addition)
   - - (subtraction)
   - * (multiplication)
   - / (division)
   - % (modulo)

◇ Building Your First Python App: A Simple Calculator

Now that we understand the basic syntax, let's build a simple calculator app that can add, subtract, multiply, and divide numbers based on user input.

```
print("Welcome to the Simple Calculator App!")

# Take user input for the two numbers
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Display operation choices
print("\nSelect operation:")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")

# Take the user's choice
operation = input("\nEnter the operation (1/2/3/4): ")

# Perform the operation
if operation == '1':
    result = num1 + num2
    print(f"\n{num1} + {num2} = {result}")
elif operation == '2':
    result = num1 - num2
    print(f"\n{num1} - {num2} = {result}")
elif operation == '3':
    result = num1 * num2
    print(f"\n{num1} * {num2} = {result}")
elif operation == '4':
    if num2 != 0:
        result = num1 / num2
        print(f"\n{num1} / {num2} = {result}")
    else:
        print("\nError! Division by zero is not allowed.")
else:
    print("\nInvalid Input! Please select a valid operation.")
```

◇ Breaking Down the Code

- Input: We use the input() function to get input from the user. The user enters two numbers, and we use float() to ensure they can enter decimal values.
- Conditionals: Using if, elif, and else, we determine which operation to perform based on the user's choice.
- Error Handling: We also include a check for division by zero, which is handled gracefully by an if condition.

Running the App

When you run the above code, the app will ask the user to input two numbers and select an operation (addition, subtraction, multiplication, or division). Based on the operation, it will calculate and display the result

Welcome to the Simple Calculator App!
Enter the first number: 12
Enter the second number: 4

Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide

Enter the operation (1/2/3/4): 4

12.0 / 4.0 = 3.0

Outro

Congratulations! In this lesson, you've:

- Learned the basic Python syntax: variables, operators, and functions.
- Built your first Python app — a simple calculator.
- Got hands-on experience with taking user input and performing operations.