Python Course - Day 004: Variables and Code Execution

007_ Variables Part 1: Fundamentals

In Python, variables are used to store and manipulate data. They act as containers for values that can be changed throughout your program.

Variable Declaration Syntax

```
variable_name = value
```

This syntax assigns the `value` to the `variable_name`. Python uses dynamic typing, which means you don't need to declare the variable type explicitly.

Name Convention and Rules

- ✓ Can start with letters (a-z, A-Z) or underscore "_"
- ✗ Cannot start with numbers (0-9) or special characters (@, -, %, etc.)
- ✓ Can include numbers (0-9) or underscore "_" after the first character
- ✗ Cannot include special characters anywhere in the name
- ⚠ Are case-sensitive (myVar and myvar are different variables)

Best Practices for Variable Naming

One word: Normal
Example: `name`

Two words: camelCase
Example: `myName`

Two or more words: snake_case
Example: `my_full_name`

Choose descriptive names that reflect the purpose or content of the variable. This improves code readability and maintainability.

008_ Variables Part 2: Code Execution Process

Understanding how Python code is executed is crucial for effective programming. Here are key concepts related to code execution:

Source code: The original code written by the programmer in a high-level language (like Python)

Translation: The process of converting source code into machine language that the computer can understand and execute

Compilation: A method where the entire source code is translated into machine code before runtime

Runtime: The period during which a program is executing its instructions

Interpreted: A method where code is translated line by line during execution, as opposed to being compiled all at once

Python's Execution Model

Python is typically described as an interpreted language, but it actually uses a hybrid approach:

1. Python source code is first compiled to bytecode
2. This bytecode is then interpreted by the Python Virtual Machine (PVM) at runtime

This approach combines some of the speed benefits of compilation with the flexibility of interpretation.

Multiple Variable Assignment

Python allows you to assign values to multiple variables in a single line:

```
a, b, c = 1, 2, 3
```

This is equivalent to:

```
a = 1
b = 2
c = 3
```

This feature, known as tuple unpacking, can make your code more concise and readable when working with related variables.

Summary and Best Practices

- Use descriptive variable names that indicate the purpose of the data
- Follow PEP 8 guidelines for naming conventions (snake_case for variables and functions)
- Balance between conciseness and clarity when naming variables
- Use comments to explain complex logic or non-obvious variable purposes
- Avoid reusing variable names in the same scope to prevent confusion