Day 8
18/07/2025

Development Environment setup

Beginers use less complicated tools

Requirements

① Compiler / Interpreter

② Editor → Source code Editor
  → Text Editor

Source code stored in text format with Extension

- .py
- .java
- .c
- .cpp

Specialized feature

③ Debugger

visual representation of code Execution

④ Project setting

source code library modules images ext ext

The software which have "Debugger project Editor" is called as IDE

IDE → Integreatted Development Environment

$$\begin{tikzcd}
\text{① Compiler / Interpreter} \arrow[r, bend left, "\times"] & \text{⑤ standard}
\end{tikzcd}$$

Libraries

I/O, file, time
other operation's

(2) Editor
(3) Debugger
(4) project

] IDE

only for Python → (6) Documentation

---

(1) Find which Compiler to use
Decide compiler / Interpreter

(2) Find the Right IDE

Development Environment Setup

Introduction

how to set up a **development environment**. This is an essential step for anyone learning programming. Whether you are just starting or already working in a company, your tools and setup can vary. But at the core, the essential tools remain the same.

Let’s walk through the key components required to write, run, and debug code effectively.

Environment Setup for Beginners vs. Professionals

Beginners or Students

- Use **simple and lightweight tools**.
- Aim is to reduce complexity and focus on learning the basics.

Professionals

- Work with **large-scale projects** (thousands of files, millions of lines of code).
- Use **robust tools** with advanced features.

Essential Tools in a Development Environment

1. Compiler or Interpreter

To convert high-level programming languages (e.g., Python, Java, C) into machine-understandable binary code.

- **Compiler**: Translates code all at once (e.g., C, C++).
- **Interpreter**: Translates and runs code line-by-line (e.g., Python).

```
# Example: Compile C code
gcc hello.c -o hello
./hello
```

2. Source Code Editor

Used to write and save code.

- Basic: Notepad, vi, Notepad++
- Advanced: Visual Studio Code, Sublime Text, Atom

Even though different programming languages have different file extensions (.py, .java, .c), the files are still plain text and editable in any text editor.

```
# Python example
print("Hello, World!")
```

3. Debugger

A debugger allows you to **run code step-by-step**, visualize how it executes, and identify logic or syntax errors.

- Useful to see **variable changes**, **loop flow**, and **code behavior**.
- Especially important for understanding control flow.

Analogy:

Think of a debugger as a "slow-motion camera" that lets you observe exactly how your code behaves.

4. Project Structure and Management Tools

Projects aren't just single files — they include:

- Source files
- Library dependencies
- Images/assets
- Configuration files

Project tools help organize and bundle everything logically.

5. IDE (Integrated Development Environment)

An IDE combines all essential tools:

- Editor
- Compiler/interpreter

- Debugger
- Project management

Popular IDEs:

- Visual Studio Code (VS Code)
- PyCharm
- Eclipse
- IntelliJ IDEA
- NetBeans

```
IDE = Editor + Compiler + Debugger + Project Tools
```

> *IDEs simplify development by bringing everything into one window.*

Standard Libraries and Packages

When you install a compiler or interpreter, standard libraries are usually installed too.

These libraries allow you to:

- Handle file operations
- Manage date/time
- Accept input/output
- Perform network operations

```
import os
print(os.getcwd())  # Prints current working directory
```

In some languages like Python, documentation is also bundled and available locally.

How to Choose the Right Tools

Language-Specific Tools

Your first task is to determine:

- Which compiler/interpreter to use?
- Which editor or IDE works best for your chosen language?

Evolution of Tools

- Tools and compilers evolve over time.
- A setup from 5 years ago may not be ideal today.

Pro Tip

Learn **how to research, evaluate, and install** modern tools:

- Check official language documentation.
- Explore developer forums and tutorials.
- Watch updated installation guides and walkthroughs.

Example: Setting Up Python in 2024

1. **Download Python**: Visit python.org (https://www.python.org/)
2. **Install VS Code**: code.visualstudio.com (https://code.visualstudio.com)
3. **Install Python extension in VS Code**
4. **Write code in `hello.py`**
5. **Run code using built-in terminal**

```
print("Setup successful!")
```

Summary

- A development environment includes a compiler/interpreter, editor, debugger, libraries, and optionally an IDE.
- Students and professionals may use different tools, but the core concepts remain the same.
- Learning how to use a debugger and manage project structure is crucial.
- Use an IDE to simplify your workflow.
- Tools evolve, so learn **how to learn** — that’s the most valuable skill.
