Week in Review: Git, Terminal, Algorithms & Flowcharts

A recap of our journey into version control, command-line navigation, and project planning this week.

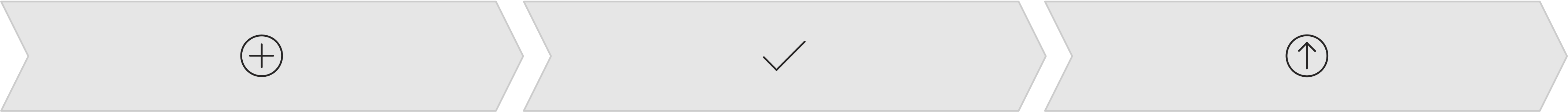
CORE CONCEPTS

Mastering the Git Workflow

We explored the fundamental Git commands that form the backbone of version control. Understanding how **git add**, **git commit**, and **git push** work together is crucial for managing your code effectively.

Think of it like packing for a trip: you add items to your suitcase (git add), seal it up (git commit), and then ship it off (git push) to its destination.

The Git Triple Play



git add

Stage your changes—tell Git which files you want to track

git commit

Save a snapshot with a message describing what changed

git push

Upload your commits to GitHub for the world to see

These three commands work as a team. You can't push without committing, and committing without adding won't save your work. Master this sequence and you've unlocked the power of version control!

Understanding the Terminal Ecosystem

Kernel

The core of your operating system—it manages hardware and system resources

Shell

Your interpreter that translates commands into actions the kernel understands

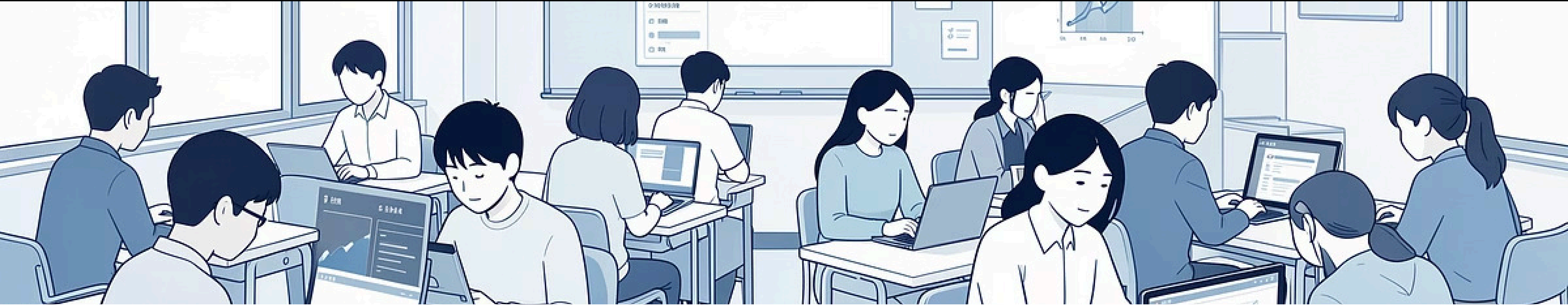
Terminal

The window where you type commands—it's your interface to the shell

Console

The physical device (or virtual equivalent) that connects you to the system

These four components work together seamlessly. When you type a command in the terminal, it passes through the shell to the kernel, which executes it. Understanding this relationship helps you troubleshoot issues and communicate more effectively about technical topics.



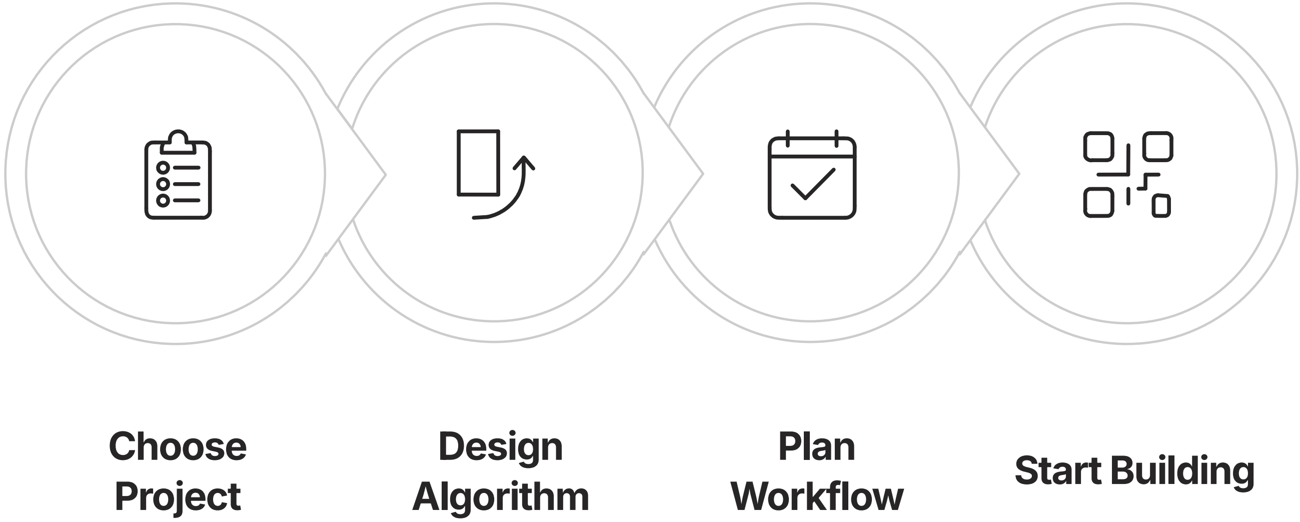
Getting Connected

We successfully logged into **Schoology** this week, setting up our accounts and exploring the platform. Schoology will be our hub for assignments, resources, and communication throughout the course.

Make sure you can access it easily—you'll be using it regularly to submit work and check announcements.

☆ PROJECT PLANNING

From Idea to Implementation



Choose Project

Design Algorithm

Plan Workflow

Start Building

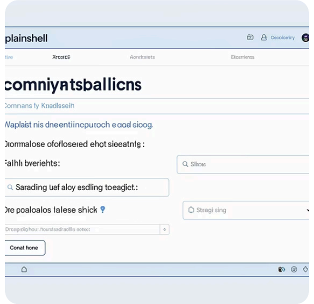
This systematic approach ensures you think through your project before diving into code.

You Picked Your Project!

Everyone selected a project this week and created both an **algorithm** (the step-by-step logic) and a **workflow** (the plan for getting it done).

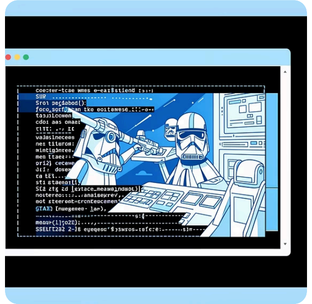
This planning phase is critical—it helps you spot potential problems early and keeps you organized as you build.

Helpful Resources We Discovered



explainshell.com

An amazing tool that breaks down any shell command into plain English. Paste in a confusing command and get a detailed explanation of every part. Bookmark this one!



Terminal Star Wars

We watched the classic Star Wars film recreated entirely in ASCII art, running right in the terminal. It's a fun demonstration of what's possible with command-line tools and creativity.

Key Takeaways

1

Git commands work as a sequence

Add, commit, and push are interconnected steps in version control

2

The terminal is powerful

Understanding kernel, shell, terminal, and console relationships unlocks deeper system knowledge

3

Planning prevents problems

Creating algorithms and workflows before coding saves time and reduces frustration

4

Resources are your friends

Tools like explainshell.com make learning command-line concepts much easier

What's Next?

You've built a solid foundation this week. Now it's time to put these skills into practice as you work on your projects. Keep experimenting with Git commands, get comfortable in the terminal, and don't hesitate to use explainshell.com when you encounter unfamiliar commands.

Remember: Every developer started exactly where you are now. Keep practicing, stay curious, and enjoy the journey!