

Week 1: Linux & Git Mastery - Daily Syllabus

Goal: To make the terminal your home and Git your second language. No more fear of the command line.

Day 1: Linux File System & Basic Navigation

Why: Everything is a file in Linux. You need to move around like a pro.

What to Learn:

pwd, ls (-l, -a, -lh flags), cd (.., ~, -)

mkdir, touch, cp, mv, rm (-r flag)

Understanding absolute vs. relative paths.

Project:

Create a project directory structure from the command line for a data project: project\_name/{data/raw,data/processed,scripts,notebooks}

Create dummy files (touch data/raw/sales.csv) within the structure.

Day 2: File Manipulation & Text Processing

Why: You'll live in log files and config files. You need to view and search them quickly.

What to Learn:

cat, less, head, tail (especially tail -f for live log watching)

grep: The most important tool. Learn -i, -v, -n, -r.

wc (word count) with -l for counting lines.

Project:

Download a large log file (e.g., from a Apache server).

Use grep to find all "ERROR" messages.

Use wc -l to count how many errors there are.

Day 3: Superuser Power & Permissions

Why: To understand and avoid "Permission Denied" errors forever.

What to Learn:

sudo: Running commands as the superuser.

chmod: Changing file permissions (understand octal notation: 755, 644).

chown: Changing file ownership.

Understanding ls -l output (drwxr-xr-x).

Project:

Create a script you can't run. Use chmod to make it executable.

Deliberately create a file that requires sudo to edit, and then edit it.

Day 4: Bash Scripting & Automation I

Why: To stop typing the same commands over and over.

What to Learn:

Shebang (#!/bin/bash)

Variables (MY\_VAR="value", echo $MY\_VAR)

Command substitution: $(command)

Passing arguments: $1, $2

Project:

Write a script create\_project.sh that automates your Day 1 project. It should take the project name as an argument and build the folder structure.

Day 5: Bash Scripting & Automation II

Why: To make your scripts smart and powerful.

What to Learn:

Conditional statements: if [ condition ]; then ... fi

Loops: for and while

Exit codes: $?

Project:

Upgrade your create\_project.sh script to check if the project directory already exists before creating it. If it does, exit with an error message.

Day 6: Git Core & Collaboration

Why: To understand how Git actually works, beyond just the commands.

What to Learn:

The Three Areas: Working Directory, Staging Index, Local Repository.

git add, git commit (write good, descriptive commit messages!)

git push, git pull

.gitignore file

Project:

Initialize a Git repo in your project from Day 1. Make commits for each logical step (e.g., "create directory structure", "add raw data file"). Push it to GitHub.

Day 7: Git Branching & Time Travel

Why: This is what makes Git powerful. Mastering branches is non-negotiable.

What to Learn:

git branch, git checkout -b <new-branch>

git merge (fast-forward and recursive)

git log (--oneline, --graph)

git rebase: The key to a clean history. Practice this a lot.

Resolving merge conflicts.

Project:

Create a feature branch. Make a change. Then, rebase it onto your main branch for a clean history. Push the rebased branch and create a Pull Request on GitHub.

Mindset for the Week:

Live in the Terminal: Do everything here. No clicking around in file explorers.

Embrace the Struggle: You will get errors. You will mess up. Read the error message—it usually tells you exactly what's wrong. This is how you learn.

Manual Pages are Your Friend: Get used to man <command> (e.g., man grep) to read the official documentation for a tool.

This week will build the strongest possible foundation for you. Every single thing you learn here will be used every single day in your DevOps career. Master this, and the following weeks will be much smoother.