Lecture 1: Introduction to Python and Command Line Basics

- Getting to the command line
- Navigating the file system with `*sh`
- Basic file operations (creating, moving, copying, deleting)
- Pipes and command chaining
- Introduction to shell scripting, variables, and `cron`
- Running Python scripts from the command line
- Python basics: syntax, data types, and control structures

Class structure

- Lectures cover new material with hands-on demos
- Lectures end with a practical assignment
- Lab for help completing the practical assignment
- Always due the following week unless otherwise noted

Resources: Command Line

- LinuxCommand.org
- The Linux Command Line book
- The Missing Semester

Resources: Python

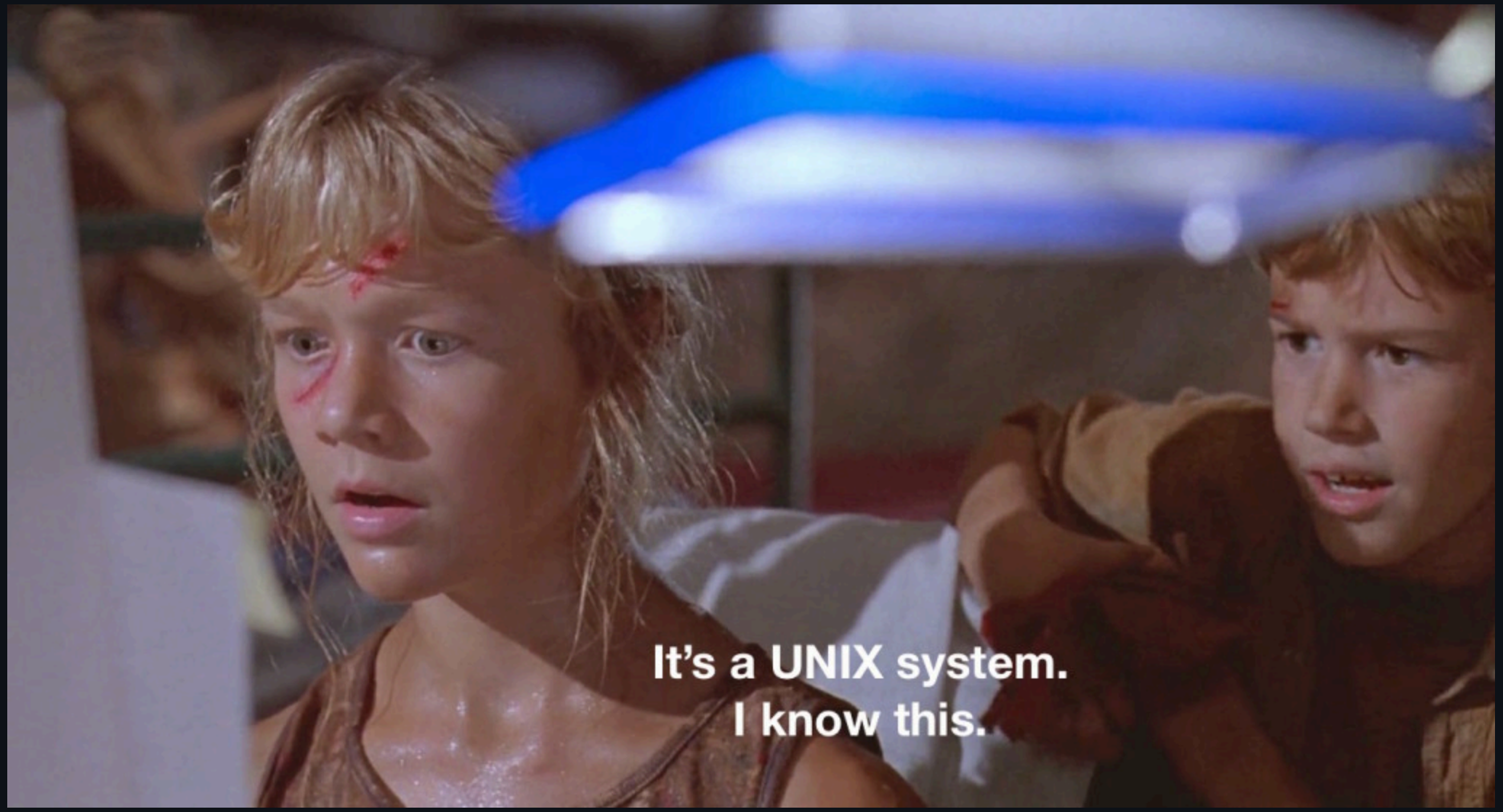
- *Whirlwind Tour of Python*, VanderPlas - author's website
- *Think Python*, Downey - purchase or read at Green Tea Press
- *Hitchhiker's Guide to Python!* - official documentation

Getting Help

- RTFM `man COMMAND`
- Ask the computer `COMMAND --help`
- Ask a bigger computer (Claude, ChatGPT)
- Come to lab!

In the beginning there was `sh`

- sh
- bash
- csh
- zsh
- Powershell


It’s a UNIX system.
I know this.

Getting to the Command Line

- Windows users:
  - PowerShell (built-in)
  - Option: Windows Subsystem for Linux (WSL)
    - `wsl --install`
- Mac users:
  - Terminal (built-in)
- Cloud options:
  - GitHub Codespaces

Command Line Navigation

- Print Working Directory:
  - `pwd` (Unix/Mac)
  - `Get-Location` (PowerShell)
- List Directory Contents:
  - `ls` (Unix/Mac)
  - `dir` (PowerShell)
- Change Directory: `cd path/to/directory`
- Special directories `~`, `.` and `..`

File Operations

- Create Directory: `mkdir new_folder`
- Create Files:
  - `touch file.txt` (Unix/Mac)
  - `New-Item file.txt` (PowerShell)
- Copy Files: `cp source destination`
- Move/Rename: `mv old_name new_name`
- Remove: `rm file.txt` (use with caution!)

Viewing File Contents

- Display entire file:
  - `cat file.txt` (Unix/Mac)
  - `Get-Content file.txt` (PowerShell)
- View beginning/end:
  - `head file.txt` / `tail file.txt` (Unix/Mac)
  - `Get-Content file.txt -Head 10` (PowerShell)

Simple Text Manipulation

- Search for patterns:
  - `grep pattern file.txt` (Unix/Mac)
  - `Select-String pattern file.txt` (PowerShell)
- Chaining commands with pipe `|` (more on pipes in a future lecture)

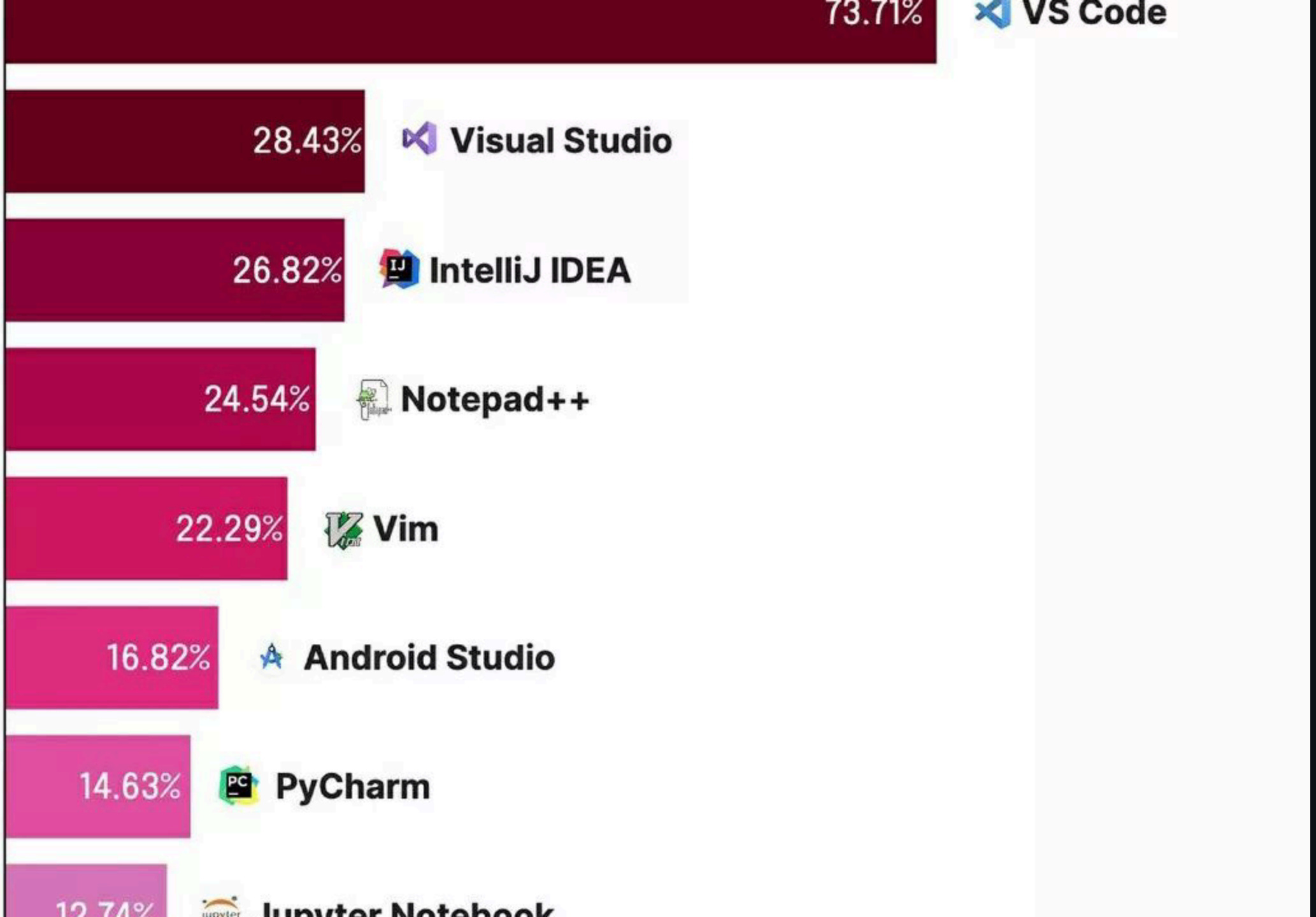
```
cat file.txt | grep pattern
```

Make it Stop!

- `Control-c` for cancel!
- `exit()`

Text Editor Options

- Visual Studio Code (what I'll be using, also available on GitHub Codespaces)
- Sublime Text
- PyCharm
- Notepad
- `nano` for quick fixes from the command line
- `code` to open a file in VS Code


73.71%
VS Code
28.43%
Visual Studio
26.82%
IntelliJ IDEA
24.54%
Notepad++
22.29%
Vim
16.82%
Android Studio
14.63%
PyCharm

Live Demo!

- Download and install Python
  - Windows: python.org or winget
    - `winget install -e --id Python.Python.3.12`
  - WSL Ubuntu - `apt install python3`
- Mac: python.org or Homebrew (recommended)
  - Install Homebrew from brew.sh
  - Windows: python.org or winget
    - `winget install -e --id Python.Python.3.12`
  - WSL Ubuntu - `apt install python3`
- Mac: python.org or Homebrew (recommended)
  - Install Homebrew from brew.sh
  - Install python `brew install python3`
- Verify installation:

```
❯ python3 --version
```

Running Python

- Interactive mode (Python REPL)

```
$ python
>>> print("Hello, World!")
```

- Running scripts

```
$ python my_script.py
```

Python Syntax Overview

- Indentation matters!
- Comments use `#`

```
# This is a comment
print("This is code")
```

Basic Data Types

- Integers: `x = 5`
- Floats: `y = 3.14`
- Strings: `name = "Alice"`
- Variables, ducks, and assignment

Simple Operations

- Arithmetic: `+`, `-`, `*`, `/`, `**` (power)
- Modulus: `%`
- String concatenation: `"Hello" + " " + "World"`

Control Structures

- Equals `==` and Not Equals `!=`
- `<`, `>`, `<=`, `>=`
- `in`, and `not in`
- `^` as `not`

Control Structures II

- If statements:

```
if x > 0:
    print("Positive")
elif x < 0:
    print("Negative")
else:
    print("Zero")
```

Control Structures III

- Compound if statements with `|` as `or`, and `&` as `and`:

```
if (x > 0) | (x < 0):
        print(“Non-zero”)
elif (x == 0) and (x == 0):
        print(“Very zero”)
else:
        # See string catenation below
        print(f“What kind of number is {x}?”)
```

Control Structures (cont.)

- For loops:

```
for i in range(5):
    print(i)
```

Printing variables

- String concatenation: `print("Hello, " + variable + "!")`
- Printing with f-strings: `print(f"Hello, {variable}!`

Live Demo!

ii. Join the course and see next week's assignmnet

5. Create a Python script that prints "Hello, Data Science!"
    i. Save it as `hello_ds.py`
    ii. Run it from the command line
    iii. Use command line to create a `scripts` folder and move your file into it
6. Write a Python script to solve the following:

> If we list all the natural numbers below 10 that are multiples of 3 or 5, we get (3, 5, 6, 9). The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000.

7. Email me:
    i. GitHub username
    ii. Answer to the problem above + script you wrote to solve it
    iii. Brief introduction (who are you, why are you here, anything you're specifically hoping to get out of the course)
    iv. Save it as `hello_ds.py`
    v. Run it from the command line

Wrap-up

- We've covered Python basics and essential command line operations
- Assignment: Practice these concepts with provided exercises
- Next lecture: Version control with Git, shell scripting and more Python

Additional Resources

- Official Python documentation
- PowerShell documentation
- Bash manual
- Codecademy Python course