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
- Lectures end with a practical assignment
- Lab for help completing the practical assignment
- Always due the following week unless otherwise noted

In the beginning there was `sh`

- Powershell
- sh
- bash
- csh
- zsh

Getting to the Command Line

- Windows users:
  - PowerShell (built-in)
  - Option: Windows Subsystem for Linux (WSL)
- 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
```

Live Demo!

Installing Python

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

```
❯ python3 --version
Python 3.12.5
```

Text Editor Options

- Visual Studio Code (what I'll be using)
- Sublime Text
- PyCharm
- Notepad
- `nano`

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)
- String concatenation: `"Hello" + " " + "World"`

Control Structures

- If statements:

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

Control Structures (cont.)

- For loops:

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

Live Demo!

Practical Exercise

1. Get everything installed
2. Create an account on GitHub and join this course
3. Create a Python script that prints "Hello, Data Science!"
4. Save it as `hello_ds.py`
5. Run it from the command line
6. Use command line to create a `scripts` folder and move your file into it

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 and More Python

Additional Resources

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