Setting Up Your Development Environment

Python, Git, and Directories

Before you start creating your application, there a few things that will make it easier and faster for you to develop as an individual and member of a team.

1. Directories: Computers and spaces don't go well together. Many of the steps that we will guide you through in the upcoming weeks use the terminal (Mac), shell (Linux), or PowerShell (Windows). If you have used spaces in the directory path it will be much harder to navigate your file system in the terminal and some development tools will be more challenging to configure. With that in mind, a few recommendations to make it easier for you to work in the terminal. If you prefer Windows but struggle with PowerShell you can alternatively consider Git Bash to enjoy many of the features of working in a Mac or Linux environment on a Windows OS.

    i. lowercase everything. usernames, filenames, directory names are all easier to type if you avoid capital letters. (There are exceptions to this, and even some languages where it isn't the standard practice, but you should use those by exception only)

    ii. single word. Name your directories and files with a single word. Generally, try to pick ones that differ in the first few characters. It will make it faster for you to type and you can use tab completion to avoid mistakes and speed up your navigation.

    iii. snake case. If you must use multiple words in a file or directory name, separate them with underscores, these interrupt your ability to read less than hyphens. Never use spaces.

2. Editors: We recommend you use Visual Studio Code (VSCode) as your code editor. There are many editors available and the choice of which you want to use is highly preference based. Until you are experienced in the language you are working in, we **do not** recommend you use a fully featured Integrated Development Environment (IDE) like PyCharm. A more simple text editor is better for learning the fundamentals. VSCode can grow with you as you can add extensions that speed up your development but at its core, it is just a text editor.

3. Python: There are many versions of Python available and currently supported. For most of what you do, the version will not matter so you should pick the one that is available in your **deployment** environment. If it isn't specified, generally use the latest stable release. If you are using Mac or Linux, use a package manager like Homebrew, apt, or yum.

4. Virtual Environments: Virtual environments are a critical tool for Python developers. The language comes with a builtin virtual environment creation tool – venv – we recommend you use for all your Python work. In week 9 we will provide you with the basic commands to get started in a virtual environment, but for now a few first principles.

    i. environment name. Name your virtual environment the same thing in every project, '.venv' is the standard naming convention.

    ii. create once, initialize always. Only create your virtual environment when you start a new project, initialize it everytime you work on it. Some editors (like VSCode) will do this for you automatically if properly configured.

5. Source Control: To be an effective member of a team that builds software (and realize all machine learning resides in software) you must be familiar with Git. There are books written on the topic and we will not replicate that here. We highly recommend you read, at minimum, Chapters 1 & 2 in Pro Git. We will provide you with commands to guide you in working with this tool over the coming weeks, but you will need to expand your knowledge of this tool to be an effective contributor to your team.