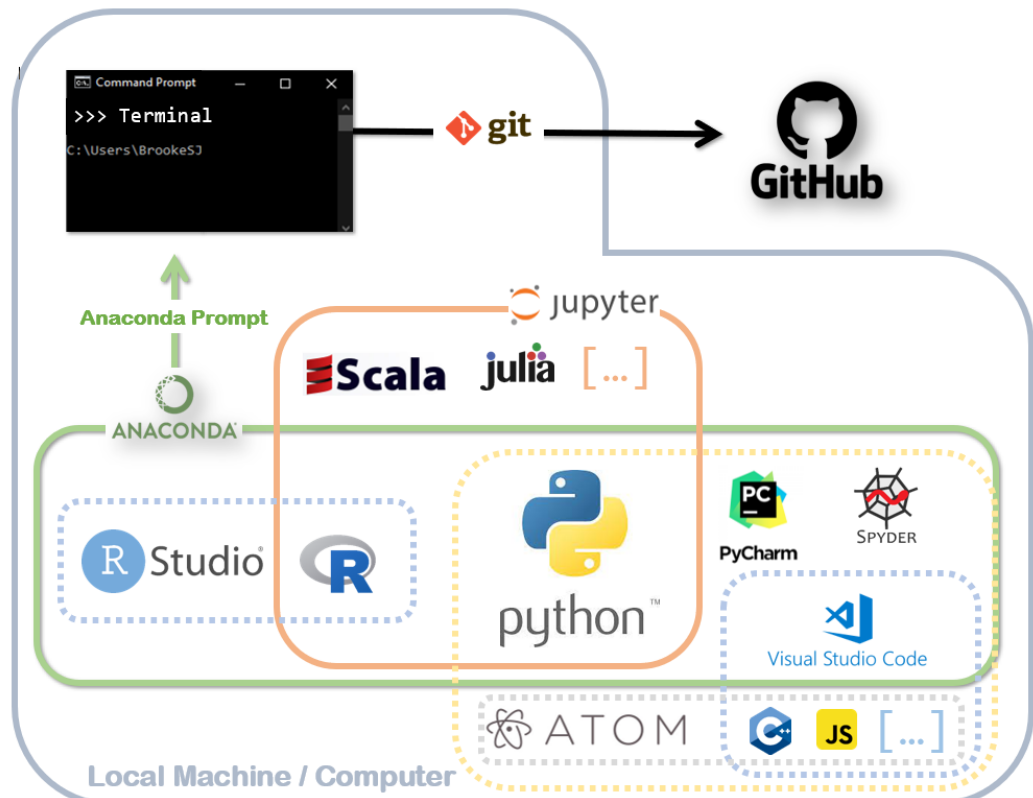**MY470 Computer Programming**

Anaconda, Jupyter, and GitHub

Week 1 Lab

Map of Software Tools and Options

IDEs

- Integrated development environment
- A software application that facilitates computer programming and software development
  - Text editor with syntax highlighting, auto completion and smart indentation
  - Shell with syntax highlighting
  - Popular libraries
  - (Debugger)
- For example:
  - Jupyter Notebooks/Lab
  - Spyder
  - PyCharm
  - **VS Code**

Anaconda

- Freemium open-source cross-platform distribution of the Pyhton and R programming languages
    - `conda` – package management system
    - `pandas`, `numpy`, `statsmodels`, `networkx`, `scikit-learn`, `matplotlib` – packages for data science
    - Anaconda Navigator – graphical user interface
    - Jupyter Notebook – web app for simple creating and sharing code
- We primarily use it as a way to standardise Python installation and package management

Installing Anaconda

- Go to https://www.anaconda.com/download/ (https://www.anaconda.com/download/)
- Select your OS
- Follow instructions

Visual Studio Code

- Visual Studio Code (VS Code) is a free IDE developed by Microsoft
- It can read, edit, and run code in a variety of languages, including Python
- Users can install powerful extensions to customize the IDE
- It features built-in Git(Hub) version control, for managing and sharing code
- We use it to write and run Python code, primarily in Jupyter Notebook files

Installing VS Code

- Go to https://code.visualstudio.com/ (https://code.visualstudio.com/)
- Select your OS
- Follow instructions

Installing Important VS Code Extensions

- If not already prompted, open the `Extensions` panel in VS Code by clicking on the icon in the left-hand toolbar
- Search for and install:
  1. `Python` extension pack by Microsoft
  2. `Jupyter` extension pack by Microsoft
  3. `GitHub Pull Requests and Issues` extension by GitHub

Jupyter Notebooks

- A popular format for creating and sharing code in documents with:
  - Live code
  - Equations
  - Visualizations
  - Explanatory text
- Supports more than 40 programming languages, including Python and R
- Notebook files have *.ipynb* extension and can be easily shared, e.g. on GitHub

Using Jupyter Notebooks in VS Code

- `New File → Jupyter Notebook`
- `+ Markdown` to insert a markdown cell
  - Lightweight markup language for text, hyperlinks, images, etc.
  - See cheatsheet: [https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
  - `CTRL+ENTER` to render
  - Double-click to edit
- `+ Code` to insert a code cell
  - `CTRL+ENTER` to run
  - On first run, you will be prompted to select a kernel (the version of Python you want to use)
  - Select `Python Environments → base (Python 3.11.x) _anaconda3/bin/python_ __Conda Env__`
    - *The exact wording may be slightly different, but you want to select the option that has* `Conda Env / Anaconda` *in it*

Basics of Coding in Jupyter Notebooks

- Code cells are run by pressing `CTRL+ENTER` or by clicking the `Run Cell` button
- The output of the last line of code in a cell is displayed below the cell
- Any *variables* you create are shared between cells in the same notebook, and can be explored by clicking the `Variables` button in the toolbar
- If code is taking too long to run, you can stop it by pressing the interrupt button
- If you want to restart the kernel (i.e. clear all variables from memory), you can do so by pressing the `Restart` button
  - This can help if you are encountering bugs that you can't figure out
- `Run All` runs all cells in the notebook from top to bottom. Variables from earlier cells can then be used in later cells. Your code should run from top to bottom without any errors.

Before submitting assignments, always `Restart Kernel` and `Run All` to make sure your code runs fully from a clean slate.

Alternative Python Workflow

- Use another IDE (e.g. PyCharm, or lightweight Jupyter Notebooks IDE)

or

- Use text editor (e.g. Sublime Text, VS Code again) to create .py files
- Run files in Terminal/cmd (or VS Code again)

```
> cd Path/to/file
> python filename.py
```

Working with GitHub

If you have not done this already:

- Create personal account on https://github.com/ (https://github.com/)
- Go to https://education.github.com/ (https://education.github.com/) and get the Student Developer Pack for some cool freebies

Four ways to interact with GitHub:

1. Browser
2. VS Code
3. GitHub Desktop App
4. Command line

Viewing Course Materials on GitHub with Browser

- Syllabus and lectures at http://github.com/lse-my470/lectures (http://github.com/lse-my470/lectures)
- Answers to assignments at https://github.com/lse-my470/answers-to-assignments (https://github.com/lse-my470/answers-to-assignments)
  - To view this repository, you need to be added to the Students team on the MY470 organization on GitHub
  - Registered students should provide their GitHub username by filling out the **survey on Moodle**
  - Auditors should e-mail their GitHub username to m.tsvetkova@lse.ac.uk (mailto:m.tsvetkova@lse.ac.uk)
  - Do not forget to **accept the invitation** to join the team and organization!

Submitting Assignments to GitHub with Browser (Similar for GitHub Desktop App)

1. Wait for e-mail with link to assignment (sent on Monday evening).
2. Accept invitation to assignment. This will automatically create a new repository on GitHub.com with your username.
3. **Clone/download** the repository. You can obtain the link when you click the "Clone or download" button on the GitHub page for the repository.
4. Use VS Code to make and save changes to downloaded file.
5. Use the GitHub web interface to upload and **commit** changed file **directly to the master/main branch**. Do this before the deadline.
6. We will automatically download all assignment repositories when the deadline has passed. We will then comment and mark your assignment directly in the main file you submitted.
7. Wait for a new commit from us to view our feedback on the GitHub web interface.

Cloning Course Materials from GitHub with VS Code

This is the recommended way to use Git(Hub) with the course materials

(You may need to first install and set up `git`. Follow instructions in *User_guides/MY470_Git_Setup.md (../User_guides/MY470_Git_Setup.md)*, or here: https://help.github.com/articles/set-up-git/ (https://help.github.com/articles/set-up-git/))

Firstly, if not done already, sign in to GitHub in VS Code:

1. Click on the `Accounts` icon in the bottom left toolbar.
2. Click `Sign in with GitHub`.
3. Follow the instructions to sign in to GitHub in your browser.

From the VS Code welcome screen:

1. Click `Clone Git Repository`.
2. Paste the URL of the repository you want to clone (i.e. http://github.com/lse-my470/lectures (http://github.com/lse-my470/lectures) and/or https://github.com/lse-my470/answers-to-assignments (https://github.com/lse-my470/answers-to-assignments)).
3. Select a location on your computer to save the repository (e.g., Documents/LSE/MY470).
4. Click `Clone`.
5. Open the repository by clicking `Open Repository` in the notification that appears.
6. Browse the folders, make notes, etc. You should `Commit` your changes to preserve them.

Submitting Assignments to GitHub with VS Code

This is the recommended way to use Git(Hub) with the assignments

1. Wait for e-mail with link to assignment (sent on Monday evening).
2. Accept invitation to assignment. This will automatically create a new repository on GitHub.com with your username.
3. `Clone` the repository (see VS Code instructions above).
4. Use VS Code to make and save changes in downloaded file.
5. Select the `Source Control` icon in the left-hand toolbar.
   A. Click + to `Stage` all changes (prepare them for commit).
   B. Add a short **message** summarising your changes.*
   C. `Commit` changed file **directly to the master/main branch** to record your changes.**
   D. `Sync`/**push** changes to upload to GitHub.com. Do this before the deadline.
6. We will automatically download all assignment repositories when the deadline has

passed. We will then comment and mark your assignment directly in the main file you submitted

7. Wait for a new commit from us to view our feedback (GitHub web interface / **pull** changes using VS Code).

* If you do not add a message here, VS Code will prompt you to do so when you try to commit by opening a text editor window. You should write your message in the first line of this window, and then save and close the window.

** You may be prompted to configure your Git settings the first time you commit. Follow the instructions in the notification that appears, with more detail in _User_guides/MY470_Git_Setup.md (../User_guides/MY470_Git_Setup.md)_.

Submitting Assignments on GitHub: General Notes

- Do not forget to `Sync`/**push** changes.
- Always check if the changes are online in your repository on GitHub.com. If you cannot see them after refreshing, no one else can.
- GitHub does all the version control for you. **Do not duplicate and rename files!**

Week 1 Exercise

- Write a simple program in a Jupyter notebook and submit it on GitHub
- E-mail with link to assignment will be sent by end of Monday

Optional/Advanced: Cloning Course Materials from GitHub with Command Line

*Install and set up `git`

Follow instructions here: https://help.github.com/articles/set-up-git/ (https://help.github.com/articles/set-up-git/)

Cloning

```
> cd Path/to/directory

> git clone https://github.com/lse-my470/lectures.git
```

Updating

```
> cd Path/to/lectures

> git pull
```

- Similarly, clone and update `https://github.com/lse-my470/answers-to-`

assignments
- Use VS Code to annotate your local copies
- `add` and `commit` before `pulling`

Submitting Assignments on GitHub (Command Line)

1. Accept invitation to assignment. This will automatically create a new repository with your username
2. **Clone** the repository (Terminal)

```
> cd Path/to/directory

> git clone link.git
```

3. Make changes in downloaded file (VS Code)
4. **Commit** changed file and **push to the master branch**. Do this before the deadline (Terminal)

```
> cd Path/to/directory

> git add --all

> git commit -m 'Submitting assignment'

> git push
```

5. We will automatically download all assignment repositories when the deadline has passed. We will then comment and mark your assignment directly in the main file you submitted
6. **Pull** the new version we commit to view our feedback (Terminal)

```
> cd Path/to/directory

> git pull
```

Submitting Assignments on GitHub (Command Line)

```
                         git pull
          ←─────────────────────────────────────────────────
┌──────────────┐      ┌──────────────┐      ┌──────────────┐      ┌──────────────┐
│   Working    │      │   Staging    │      │    Local     │      │    Remote    │
│  directory   │      │     area     │      │     repo     │      │     repo     │
└──────────────┘      └──────────────┘      └──────────────┘      └──────────────┘
          ──────────────→      ──────────────→      ──────────────→
            git add              git commit            git push
         (track by git)
```