

Getting started with Python

- Three approaches
 - **Jupyter Hub**
 - Easiest: no installation needed
 - Most limited
 - **Anaconda GUI**
 - **Terminal interface**
 - Most flexible
 - Largely necessary for research in astronomy



Jupyter Hub

- University has a free Jupyter Hub
 - **jupyter.utoronto.ca/hub/**
 - Access to a Jupyter Notebook/ Lab in your browser
 - Most major packages you'll need already provided



Jupyter Hub

- University has a free Jupyter Hub
 - jupyter.utoronto.ca/hub/
 - Access to a Jupyter Notebook/ Lab in your browser
 - Most major packages you'll need already provided

Some classes may also have their own Jupyter Hub

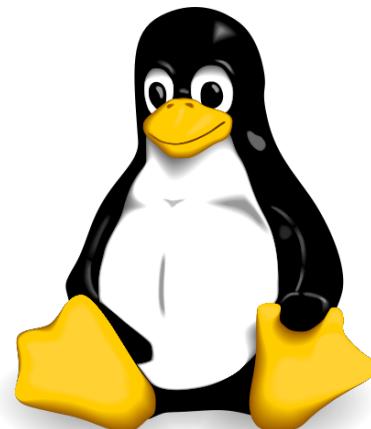
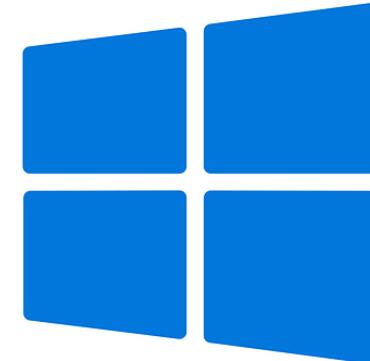
- For example AST301

**Warning: Storage on Jupyter hubs
may be extremely limited!
Keep in mind the size of your
datasets while using.**



Quick Question

- What type of computer do you use?
 - Windows
 - Mac
 - Linux
 - Other



Anaconda GUI

- Only Windows / Mac
 - Can open up variety of programmes including Jupyter from this application
 - To install follow instructions on
**[docs.anaconda.com/
anaconda/install/](https://docs.anaconda.com/anaconda/install/)**

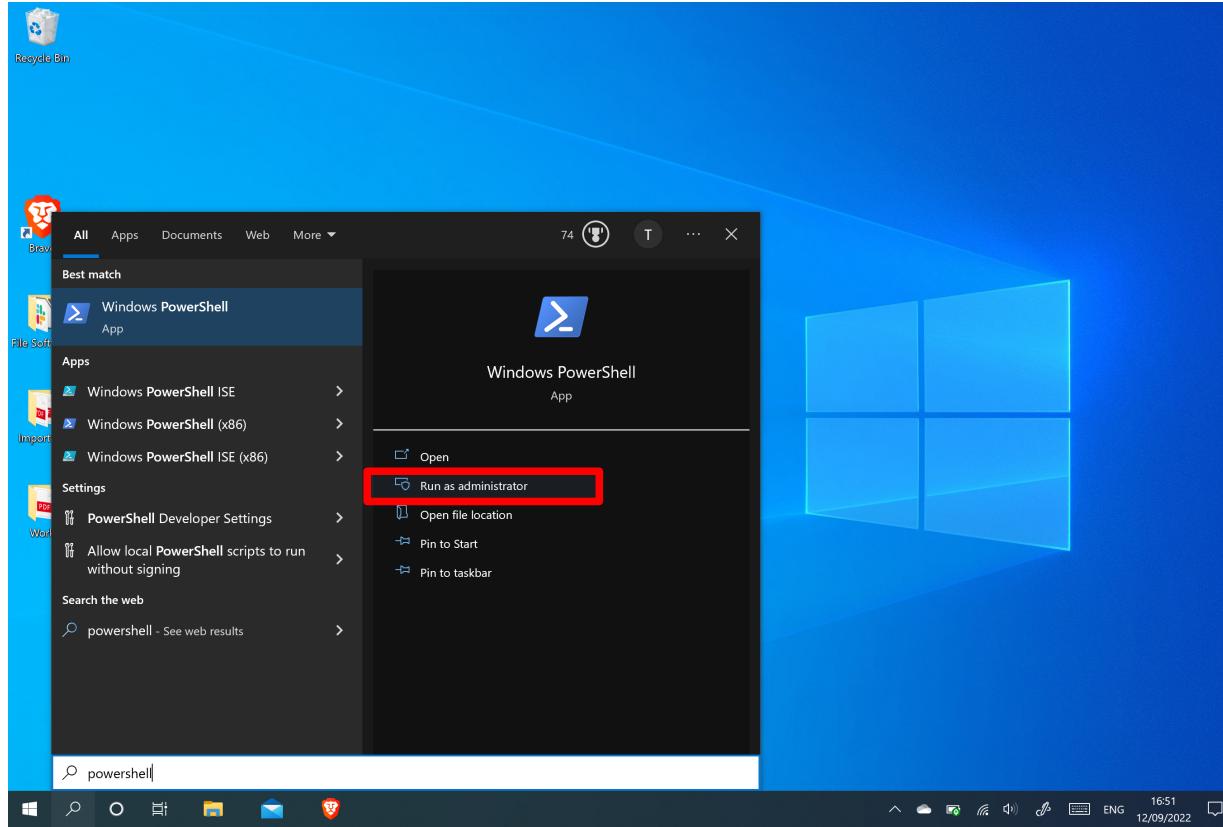


Terminal for Windows

- Most scientific computing in astronomy is developed on Unix-based systems (Linux, Mac) with heavy emphasis around the command line interface
 - Unix environments incompatible with default Windows OS
 - Built in workaround for Windows users: “Windows Subsystem for Linux” (WSL)
 - Allows Windows users to run a native Unix command line and file system in addition to normal Windows

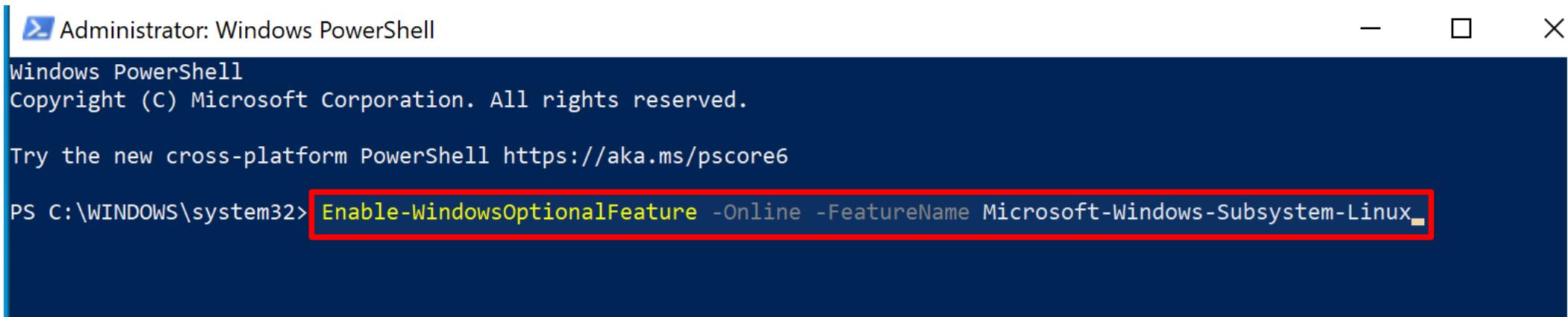
Terminal for Windows

- Open Powershell in administrator mode



Terminal for Windows

- Activate option for WSL in Windows



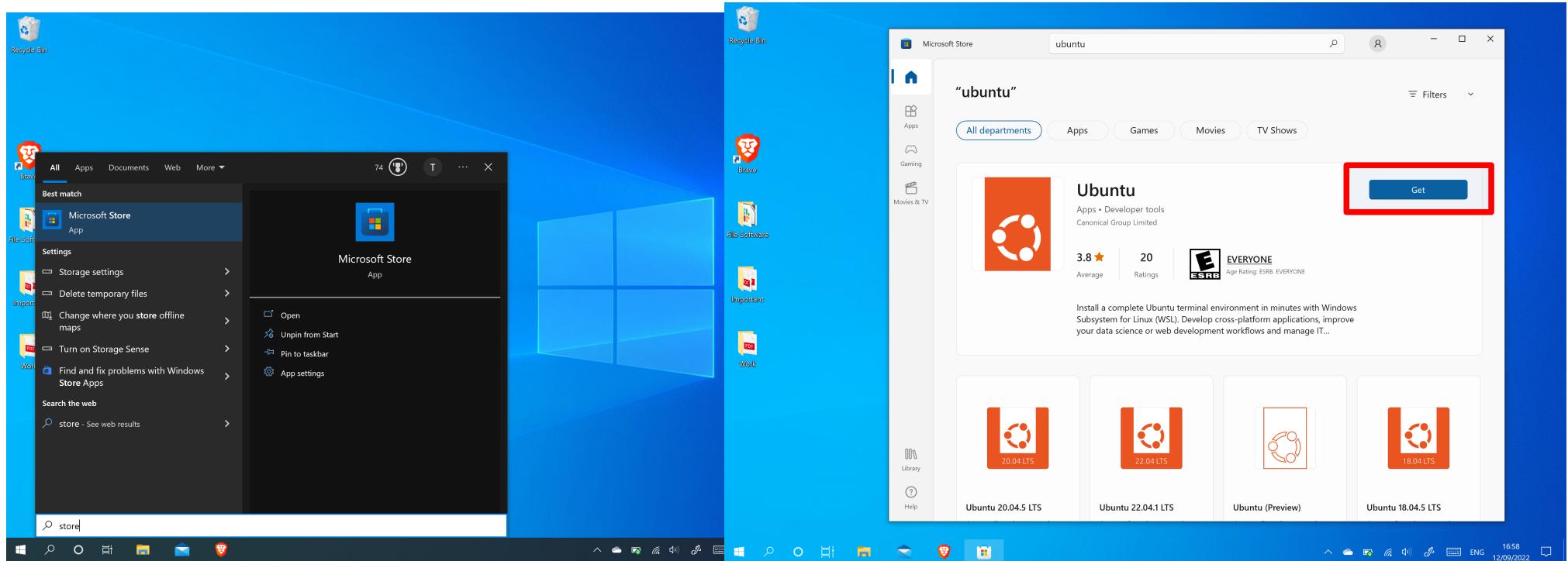
A screenshot of an "Administrator: Windows PowerShell" window. The title bar shows the application name and status. The window content displays standard PowerShell startup information, followed by a recommendation to try the cross-platform PowerShell. At the bottom, a command is entered in the console:

```
PS C:\WINDOWS\system32> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

The entire command line is highlighted with a red rectangular box.

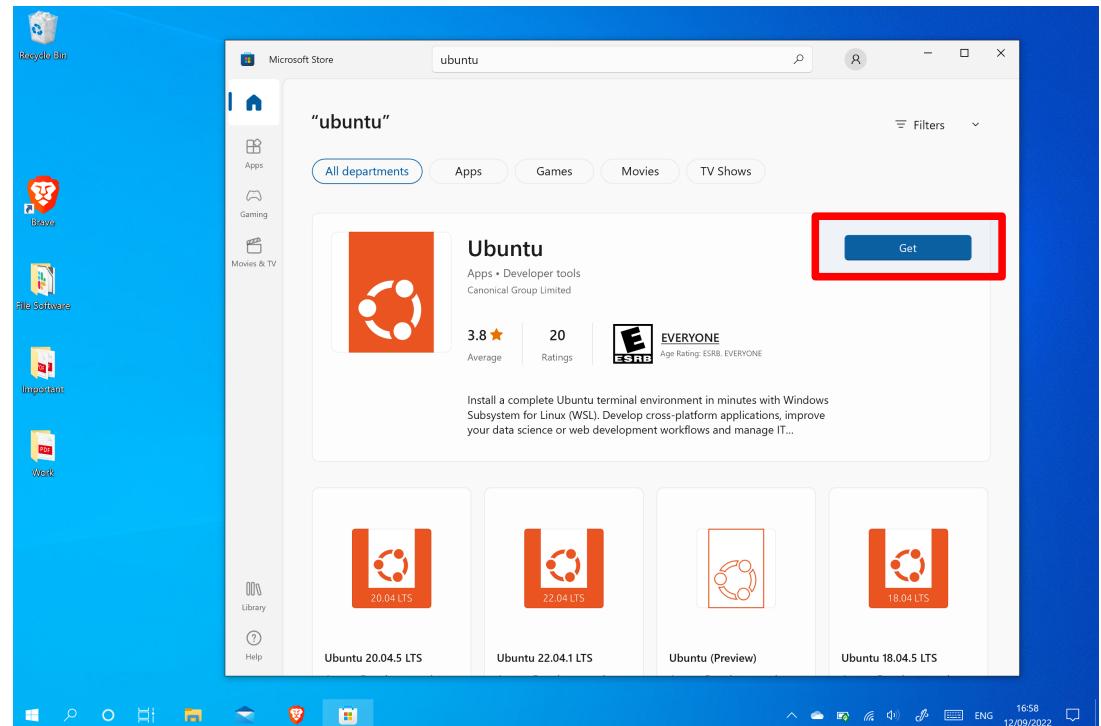
Terminal for Windows

- After running the previous command and restarting, install a linux distribution from the Microsoft Store



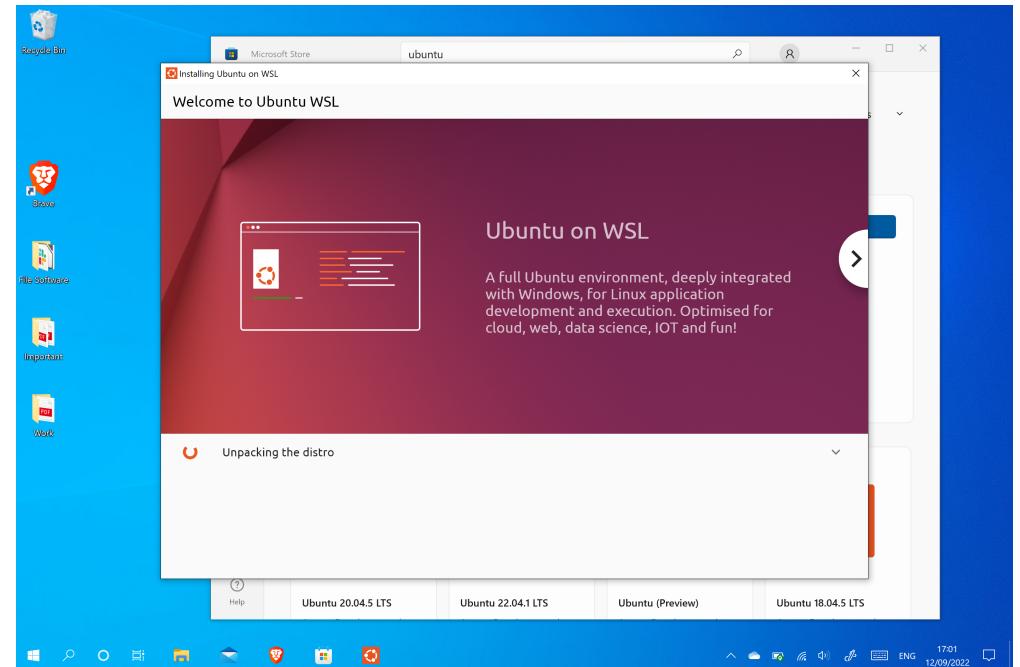
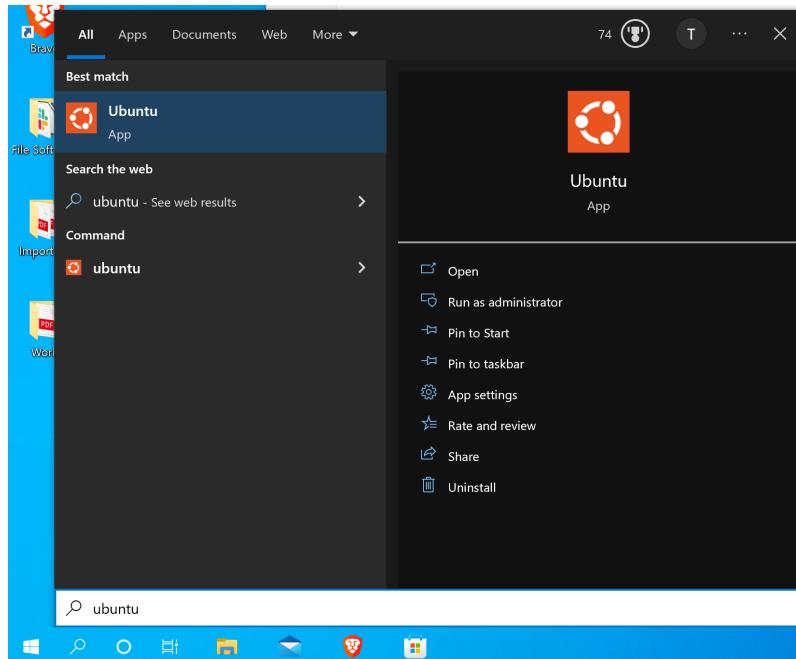
Terminal for Windows

- After running the previous command and restarting, install a linux distribution from the Microsoft Store
- Note: ubuntu is not the only option (just the one I used)
 - Debian, Fedora, Kali, Alpine, ...
 - **Just make sure to use one of the free options** and keep in mind the following steps may be slightly different than what is shown if you choose a different distribution



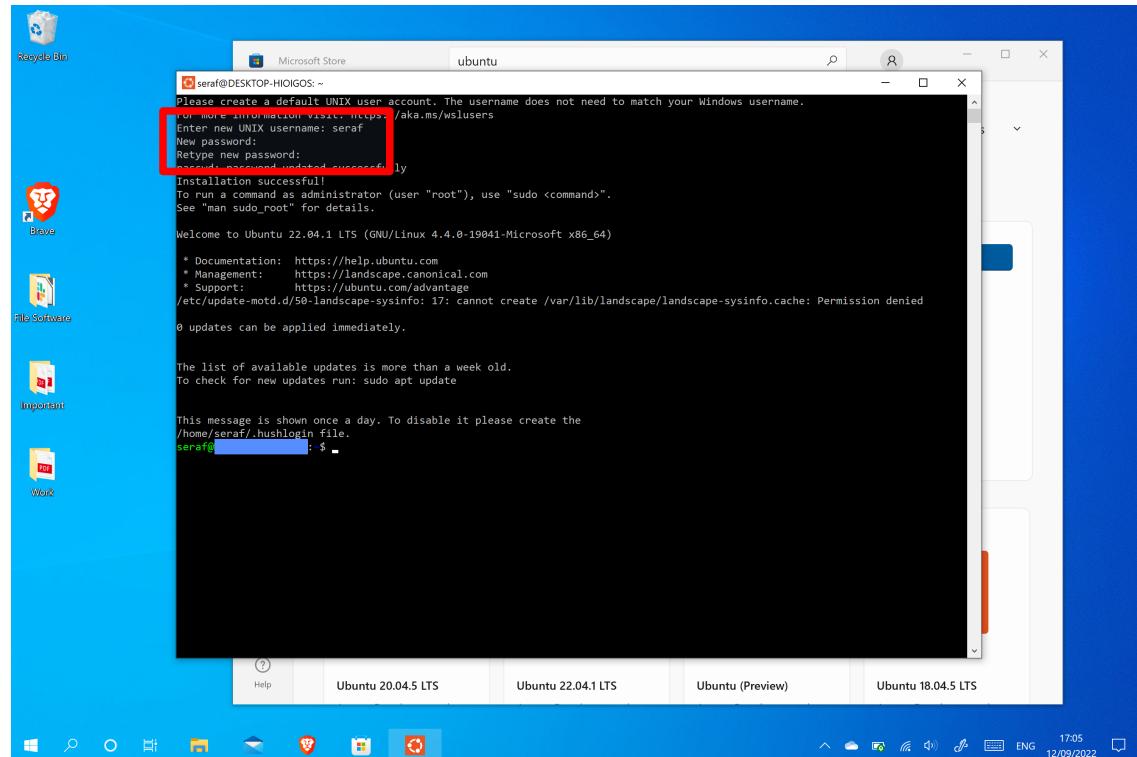
Terminal for Windows

- Run the downloaded Linux application to complete setup



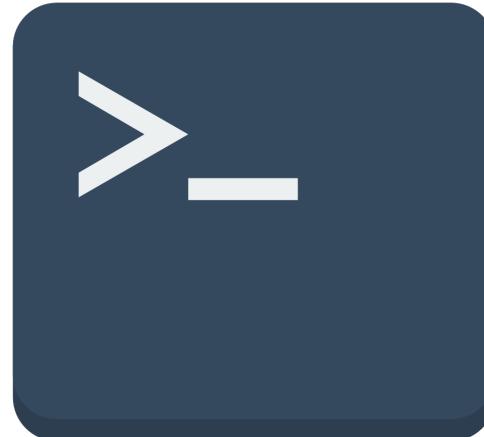
Terminal for Windows

- To finish off, set you Linux user and password
- By default you can access your Windows file system from the Linux setup at “/mnt/c/”

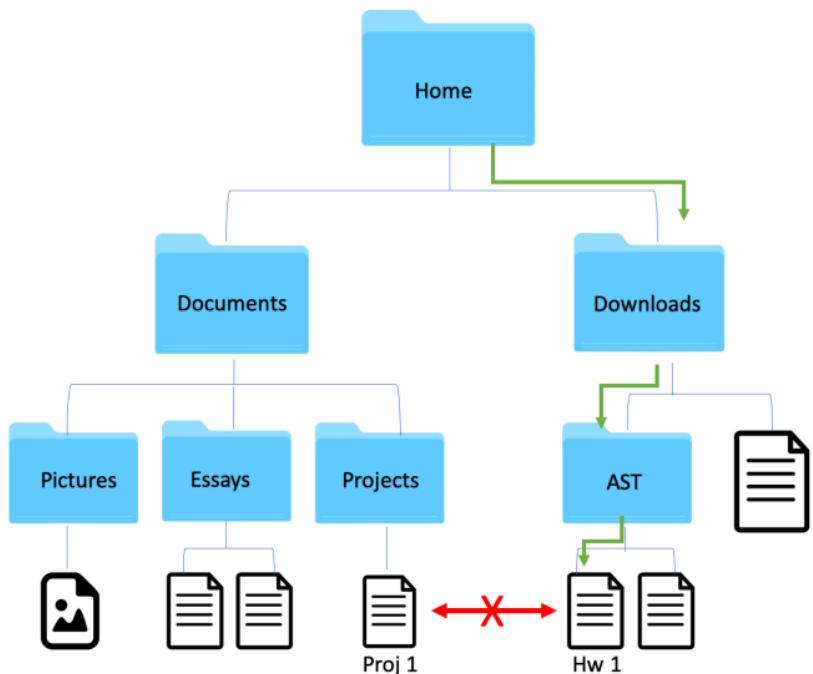


Terminal Basics

- Accessing the command line
 - Windows OS with WSL
 - Search for the WSL app installed from the Microsoft Store
 - Mac OS and Linux OS
 - Search for terminal



Terminal Basics: File System



- The file system works like a branching path.
 - Depending on where you are along said path, you'll need to give a different locations for the same file.
 - For example, from “Downloads”:
 - AST/Hw1
 - Whereas from “Projects”:
 - /Home/Downloads/AST/Hw1

Terminal Basics: Commands

Windows Command	Mac OSX/ Linux command	Description
dir	ls	List files and directories contained in current directory
cd	pwd	Print the full path to the current directory you are in
cd <dir>	cd <dir>	Change directories: replace <dir> with directory you want to enter
del <file>	rm <file>	Delete file. Cannot be undone on linux
move <file> <destination>	mv <file> <destination>	Rename a file or directory, and/or move it to a new path
md <dir>	mkdir <dir>	Create new directory

- You will also want a text editor of choice.
 - Some common options: sublime, emacs, vim, nano, ...
 - Up to you to see which you like best

Python with Terminal: Setup

- Default Python fairly bare bones.
 - Anaconda is very convenient way to install all the most common packages at once to get started
 - Link to instructions:
 - docs.anaconda.com/anaconda/install/linux (Windows OS with WSL and Linux OS)
 - docs.anaconda.com/anaconda/install/mac-os/#using-the-command-line-install (Mac OS)



Python with Terminal: Setup

- Download the desired file
 - https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh (Windows OS with WSL and Linux OS)
 - https://repo.anaconda.com/archive/Anaconda3-2022.05-MacOSX-x86_64.sh (for Mac OS)

```
seraf@[REDACTED]:~$ pwd
/home/seraf
seraf@[REDACTED]:~$ mkdir Software
seraf@[REDACTED]:~$ cd Software
seraf@[REDACTED]:~/Software$ wget https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh
--2022-09-12 17:24:20-- https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 2606:4700::6810:8203, 2606:4700::6810:8303, 104.16.130.3,
..
Connecting to repo.anaconda.com (repo.anaconda.com)|2606:4700::6810:8203|:443... failed: Resource temporarily
unavailable.
Connecting to repo.anaconda.com (repo.anaconda.com)|2606:4700::6810:8303|:443... failed: Resource temporarily
unavailable.
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.130.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 690850711 (659M) [application/x-sh]
Saving to: 'Anaconda3-2022.05-Linux-x86_64.sh'

Anaconda3-2022.05-Linux-x86 100%[=====] 658.85M  8.41MB/s   in 95s

2022-09-12 17:26:37 (6.96 MB/s) - 'Anaconda3-2022.05-Linux-x86_64.sh' saved [690850711/690850711]

seraf@[REDACTED]:~/Software$
```

Python with Terminal: Setup

- Start the installation

```
seraf@[REDACTED]:~/Software$ bash Anaconda3-2022.05-Linux-x86_64.sh
```

```
Welcome to Anaconda3 2022.05
```

```
In order to continue the installation process, please review the license  
agreement.
```

```
Please, press ENTER to continue
```

```
>>> [REDACTED]
```

Python with Terminal: Setup

- Accept the license

```
You must comply with all domestic and international export laws and regulations that apply to the software, which include restrictions on destinations, end users, and end use. Anaconda Distribution includes cryptographic software. The country in which you currently reside may have restrictions on the import, possession, use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check your country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted. See the Wassenaar Arrangement http://www.wassenaar.org/ for more information.
```

```
Anaconda has self-classified this software as Export Commodity Control Number (ECCN) EAR99 which includes mass market information security software using or performing cryptographic functions with asymmetric algorithms. No license is required for export of this software to non-embargoed countries.
```

```
The Intel Math Kernel Library contained in Anaconda Distribution is classified by Intel as ECCN 5D992.c with no license required for export to non-embargoed countries.
```

```
The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to cryptography.
```

```
Last updated February 25, 2022
```

```
Do you accept the license terms? [yes|no]  
[no] >>>
```

```
Please answer 'yes' or 'no':  
>>>
```

Python with Terminal: Setup

- Specify the installation location (if you want it to be different than the suggested default)

```
Please answer 'yes' or 'no':  
>>> yes  
  
Anaconda3 will now be installed into this location:  
/home/seraf/anaconda3  
  
- Press ENTER to confirm the location  
- Press CTRL-C to abort the installation  
- Or specify a different location below  
  
[/home/seraf/anaconda3] >>> /home/seraf/Software/anaconda3.
```

Python with Terminal: Setup

- Oops! I made a mistake. Here I should have entered “yes”, but instead I just pressed enter...

```
done
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> [REDACTED]
You have chosen to not have conda modify your shell scripts at all.
To activate conda's base environment in your current shell session:

eval "$(/home/seraf/Software/anaconda3/bin/conda shell.YOUR_SHELL_NAME hook)"

To install conda's shell functions for easier access, first activate, then:

conda init

If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Anaconda3!

=====
Working with Python and Jupyter is a breeze in DataSpell. It is an IDE
designed for exploratory data analysis and ML. Get better data insights
with DataSpell.

DataSpell for Anaconda is available at: https://www.anaconda.com/dataspell
seraf@[REDACTED]:~/Software$
```

Python with Terminal: Setup

- Not to worry, we can fix this after the fact.

```
seraf@[REDACTED]:~$ ls Software/anaconda3/
LICENSE.txt    condabin  include   man      plugins   sbin   translations
bin           doc        lib       mkspecs  pyodbc.pyi share   var
compiler_compat envs      libexec  phrasebooks  qml      shell  x86_64-conda-linux-gnu
demos          etc        libtcl   pytz      qt5      site    x86_64-conda_cos6-linux-gnu

seraf@[REDACTED]:~$ source Software/anaconda3/bin/activate
(base) seraf@[REDACTED]:~$ conda init
no change  /home/seraf/Software/anaconda3/condabin/conda
no change  /home/seraf/Software/anaconda3/bin/conda
no change  /home/seraf/Software/anaconda3/bin/conda-env
no change  /home/seraf/Software/anaconda3/bin/activate
no change  /home/seraf/Software/anaconda3/bin/deactivate
no change  /home/seraf/Software/anaconda3/etc/profile.d/conda.sh
no change  /home/seraf/Software/anaconda3/etc/fish/conf.d/conda.fish
no change  /home/seraf/Software/anaconda3/shell/condabin/Conda.ps1
no change  /home/seraf/Software/anaconda3/shell/condabin/conda-hook.ps1
no change  /home/seraf/Software/anaconda3/lib/python3.9/site-packages/xontrib/conda.xsh
no change  /home/seraf/Software/anaconda3/etc/profile.d/conda.csh
modified   /home/seraf/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

(base) seraf@[REDACTED]:~$ .
```

Python with Terminal: Setup

- If everything is set up correctly then the following check should output something similar to this:

```
base) seraf@:~> conda --help
usage: conda [-h] [-V] command ...
conda is a tool for managing and deploying applications, environments and packages.

options:
positional arguments:
command
  clean      Remove unused packages and caches.
  compare    Compare packages between conda environments.
  config     Modify configuration values in .condarc. This is modeled after the git config command. Writes to the user .condarc file (/home/seraf/.condarc) by default.
  create     Create a new conda environment from a list of specified packages.
  help       Displays a list of available conda commands and their help strings.
  info       Display information about current conda install.
  init       Initialize conda for shell interaction. [Experimental]
  install   Installs a list of packages into a specified conda environment.
  list      List linked packages in a conda environment.
  package   Low-level conda package utility. (EXPERIMENTAL)
  remove    Remove a list of packages from a specified conda environment.
  uninstall Alias for conda remove.
  run       Run an executable in a conda environment.
  search   Search for packages and display associated information. The input is a MatchSpec, a query language for conda packages. See examples below.
  update   Updates conda packages to the latest compatible version.
  upgrade  Alias for conda update.

optional arguments:
-h, --help      Show this help message and exit.
-V, --version  Show the conda version number and exit.
```

Python with Terminal

- Once a python script has been created with any text editor, it can be ran with python
`python <insert_script.py>`
- Alternatively you can open an interactive python shell with the command
`ipython`



Python with Terminal

- To run your own Jupyter notebook or lab from the command line, and view it in a browser simply run

jupyter notebook --no-browser

or

jupyter lab --no-browser

```
(base) seraf@[REDACTED]:~$ jupyter notebook --no-browser
[I 23:55:59.913 NotebookApp] JupyterLab extension loaded from /home/seraf/Software/anaconda3/lib/python3.7/site-packages/jupyterlab
[I 23:55:59.915 NotebookApp] JupyterLab application directory is /home/seraf/Software/anaconda3/share/jupyter/lab
[I 23:55:59.917 NotebookApp] Serving notebooks from local directory: /home/seraf
[I 23:55:59.917 NotebookApp] The Jupyter Notebook is running at:
[I 23:55:59.917 NotebookApp] http://localhost:8888/?token=6b227e4b553c0c4f1b36c1cb5ee5fdcc8f6d8be8bc68c5f1
[I 23:55:59.918 NotebookApp] or http://127.0.0.1:8888/?token=6b227e4b553c0c4f1b36c1cb5ee5fdcc8f6d8be8bc68c5f1
[I 23:55:59.918 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 23:55:59.930 NotebookApp]
```

To access the notebook, open this file in a browser:

file:///home/seraf/.local/share/jupyter/runtime/nbserver-297-open.html

Or copy and paste one of these URLs:

http://localhost:8888/?token=6b227e4b553c0c4f1b36c1cb5ee5fdcc8f6d8be8bc68c5f1

or http://127.0.0.1:8888/?token=6b227e4b553c0c4f1b36c1cb5ee5fdcc8f6d8be8bc68c5f1