Installing Anaconda, VS Code, Github

October 2023

1 Introduction

The following is a quick guide to setting up the components we need to get going with Python programming on this course. In this note I'll describe setting up on a Mac, but other operating systems are similar. As I mentioned there are many other guides on the Web to help you do this, or you can find help from your fellow students or via the University IT support teams. I will also cover setting up your working environment, and keeping things organised. Have a go with the notes below, and I will go through this in the tutorials this week.

In what follows, it is best to install Anaconda first, so that VS Code will automatically detect that Python is installed.

The key steps to get going this week are Anaconda and VS Code, we will work up to using Git and GitHub during the course, so you are proficient and confident with it by the end, and you can use what you have learnt on your final assignment, but do go ahead and make sure you have a GitHub account set up and try and get Git installed too.

2 Anaconda

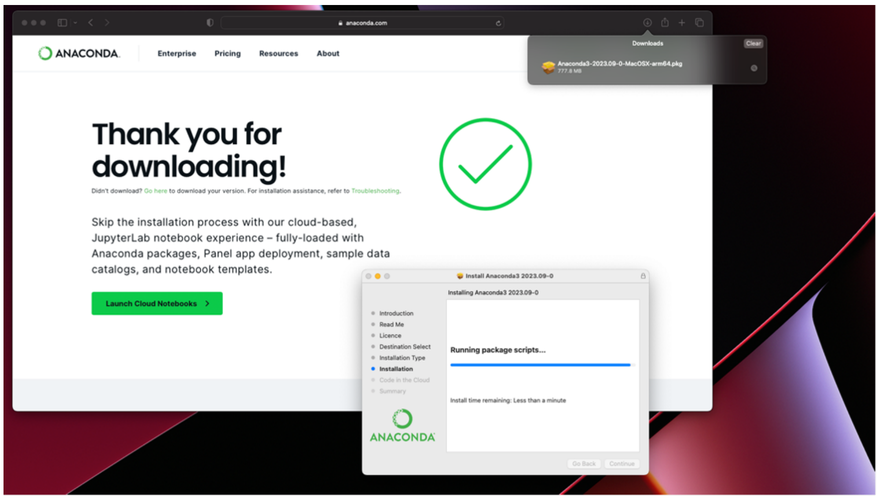
Anaconda is a package of software containing a Python interpreter and several Python packages / code libraries designed for use in Data Science. All the libraries you will need for the course are included in Anaconda. This is the software that interprets and executes our python programs (sets of instructions).

1. Visit Anaconda.com, and find the link to download the package. Click on this and select your operating system.
2. On a modern Mac, Anaconda will usually install in a folder 'User\opt\Anaconda3' or 'User\Anaconda3'where 'User' is your Mac account name.
3. Once the file has downloaded, run the install script. You should see something like Figure 1.
4. You should then see, or be able to open, the Anaconda Navigator, as per Figure 2.
5. If you then click on the 'jupyter notebook' tab in Navigator, you should see a notebook file interface open in your browser (Figure 3).

3 Organising Your Work

Keeping your work organised in some kind of folder/directory structure on your PC is important. I would suggest you set up a top level directory for your course, and then sub directories for each module. You can then set up further sub directories for your coursework / projects. See, for example the structure in Figure 4.

We could then, from the Jupyter File Manager screen (Figure 3) open a new Jupyter Notebook in the directory 'Project 1 Example', and in the first cell run our first 'hello world' style piece of Python code (Figure 5). A Jupyter Notebook is one way to run Python code (in a browser). This is a nice way to present and run code to colleagues. To develop anything serious, you will want to use an IDE, as discussed in class, which has better debugging facilities and ways of structuring larger projects. We will now install VS Code, just such an IDE.



Thank you for downloading!

Didn't download? Go here to download your version. For installation assistance, refer to Troubleshooting.

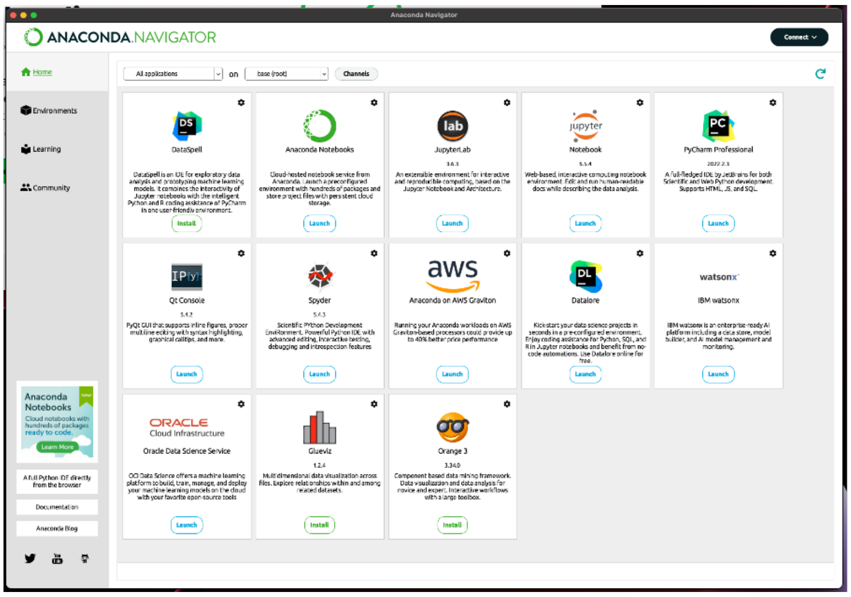
Skip the installation process with our cloud-based, JupyterLab notebook experience – fully-loaded with Anaconda packages, Panel app deployment, sample data catalogs, and notebook templates.

Installing Anaconda3 2023.09-0

Running package scripts...

Install time remaining: Less than a minute

Figure 1: Downloading and Installing Anaconda on a Mac



DataSpell
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.
Install

Anaconda Notebooks
Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.
Launch

JupyterLab
3.6.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

Notebook
6.5.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

PyCharm Professional
2022.2.3
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Launch

Qt Console
5.4.2
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

Spyder
5.4.3
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

Anaconda on AWS Graviton
Running your Anaconda workloads on AWS Graviton-based processors could provide up to 40% better price performance
Launch

Datalore
Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.
Launch

IBM watsonx
IBM watsonx is an enterprise-ready AI platform including a data store, model builder, and AI model management and monitoring.
Launch

Oracle Data Science Service
OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools
Launch

Glueviz
1.2.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

Orange 3
3.34.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

Figure 2: The Anaconda Explorer

4 Installing VS Code

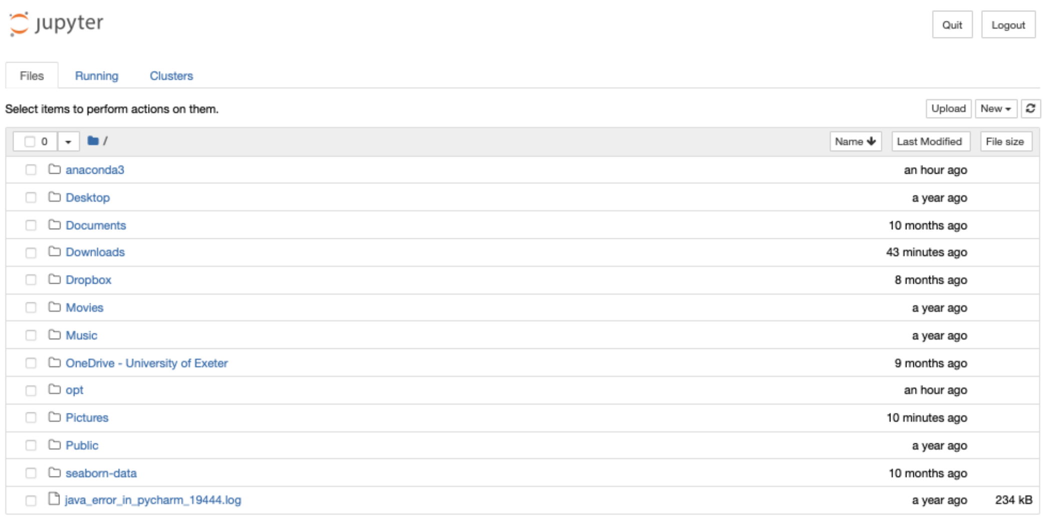
1. Download and install VS Code from the link in the Resources section of ELE or from here . Run through the setup and select a colour scheme etc. that you are happy with.

2. VS Code can be used to develop code in many different computer languages. We now customise it to use Python.

3. We need to use the VS Code Python extension from (details here).

4. To install this with VS Code, open VS Code and select 'View\Extensions'. Search for 'Python microsoft' and select as per Figure 6.

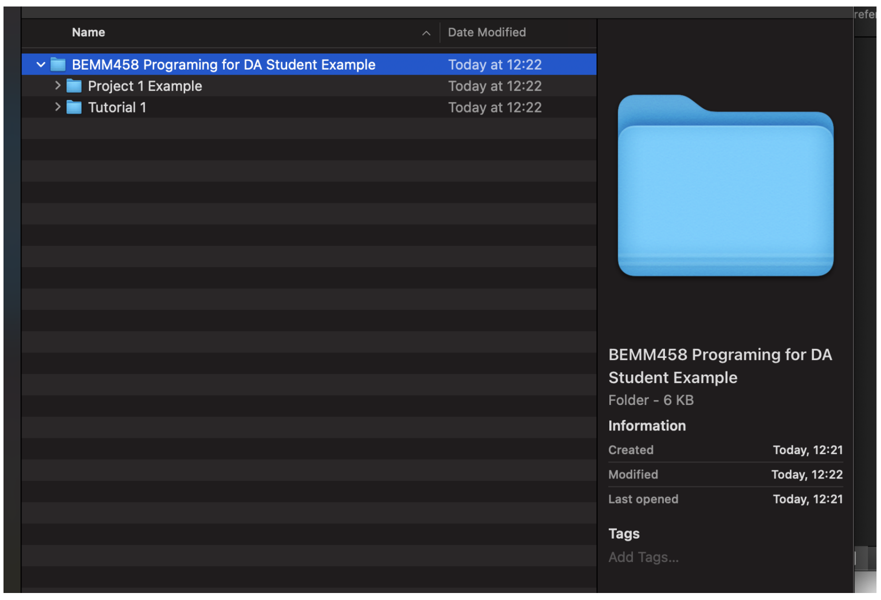
We can now use VS Code to create a Python program. We shall use the same python 'Hello Exeter' code snippet that we used in Jupyter Notebook above. We start by opening the directory created above (for me it was 'Project 1 Example'). And create a new file, we'll call it 'example1.py'. This is essentially just a text file, but with a '.py' extension which tells your computer operating system that this is a Python code file.



Select items to perform actions on them.

Name	Last Modified	File size
anaconda3	an hour ago	
Desktop	a year ago	
Documents	10 months ago	
Downloads	43 minutes ago	
Dropbox	8 months ago	
Movies	a year ago	
Music	a year ago	
OneDrive - University of Exeter	9 months ago	
opt	an hour ago	
Pictures	10 minutes ago	
Public	a year ago	
seaborn-data	10 months ago	
java_error_in_pycharm_19444.log	a year ago	234 kB

Figure 3: The Jupyter Notebook File Window



Name	Date Modified
BEMM458 Programing for DA Student Example	Today at 12:22
Project 1 Example	Today at 12:22
Tutorial 1	Today at 12:22

BEMM458 Programing for DA Student Example
Folder - 6 KB

Information

Created	Today, 12:21
Modified	Today, 12:22
Last opened	Today, 12:21

Tags
Add Tags...

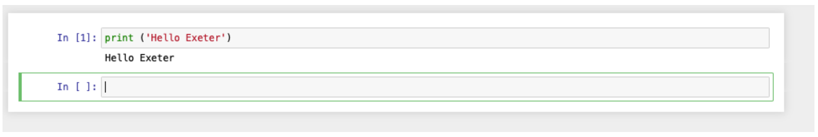
Figure 4: Example File Structure

Note that if you have other versions of Python installed on your computer, you may need to select a version of Python for VS code to use to run your code (i.e. the one installed above using Anaconda). To do this, consult the instructions here.

If we create a new file, type the python command 'print('Hello world')', save and run this code (by clicking the run or 'play' triangle button in the top right hand corner of the VS Code window) we should see a terminal window open and our code run within it. We will cover several other ways of doing this, and running the code in 'debug' mode in this week's lecture.

5 Using GitHub

Git / GitHub is a system for *Software Version Control*. Basically, we use this to keep track of the development process of our software. This is good practice for each of us if we are building a software project, and being able to using Git / GtHub is an extremely valuable addition to your CV as a data analyst. Git is an essential part of working on joint products in data analytics, as it simplifies the process of revising and updating code written by different team members. Be aware that the learning curve is quite steep, but it will be worth it... eventually... There are some videos on Git here.



```
In [1]: print ('Hello Exeter')
        Hello Exeter

In [ ]: |
```

Figure 5: First line of real Python code in a Jupyter Notebook

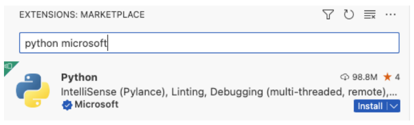

EXTENSIONS: MARKETPLACE
python microsoft
Python
98.8M ★ 4
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote),...
Microsoft
Install

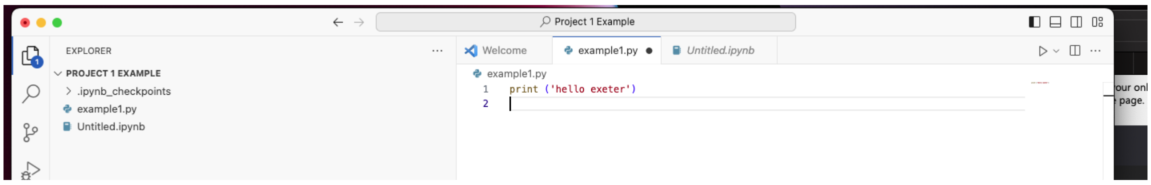
Figure 6: Installing the VSCode Python Extension

To do this you will need to use the command line / Terminal on your machine. I promise to keep this to a minimum. You will need to find out how to use this on your machine (its available via launchpad on a Mac).

5.1 Some Github Jargon

Unfortunately, like many things in programming, GitHub comes with its own jargon. What follows is an attempt to clarify some of this. (Source [?] and GitHub help). We won't necessarily use all of this, but it's here as an introduction.

- *Repository* - Similar to a folder on Dropbox, except hosted on GitHub. Repositories sync with Git so that users can view a human-readable format of a project's version control history on GitHub. In GitHub Classroom, each student assignment is a repository
- *Commit* - Rather than saving a file with a new name after file revisions, users can commit their revised work with Git. Each commit is associated with a commit message and a unique ID, allowing users to revisit changes made throughout the project. An example usage would be students committing changed code after writing a new function.
- *Push* - After a user makes changes to files within a GitHub repository on their local computer, they can push these changes back to GitHub, so the changes can be seen by other users with access to the repository. This is similar to updating a file on a local computer and syncing the changes to Dropbox. An example usage would be students pushing a completed assignment to GitHub.
- *Pull* - If a user wants to obtain the most up-to-date version of a GitHub repository, they can pull the changes to their local computer. This is similar to syncing changed files from a Dropbox folder to a local computer. An example usage would be an instructor pulling updated student assignment repositories.
- *Pull request* - If a student makes changes to a shared repository, such as lecture notes, they can submit a pull request for the instructor to review these changes and to choose whether or not to integrate these changes into the shared repository.
- *Branch* If a student clones the course materials to their local computer, they can create their own branch. Changes made on this branch will not be reflected in the main version of the course materials, but can be merged using a pull request.
- *Issue* An issue is a suggested improvement to a GitHub repository, made through GitHub. For each issue, users can discuss the suggested changes in a forum. An example usage would be students submitting an issue for a mistake in an lecture notes, within the lectures repository.
- *Clone* Users can clone a full GitHub repository onto their local computer, so they can edit code locally. Similar to downloading a folder from Dropbox, cloned repositories are synced to GitHub repositories, allowing for users to easily push and pull changes. An example usage is students cloning the repository which contains lecture notes.



```
print ('hello exeter')
```

Figure 7: First .py file

5.2 Setting up and Installing Git / Github

Ok. Here goes. This is going to hurt. There is a good guide to doing all this stuff here.

1. Register for a GitHub account here *USING YOUR EXETER EMAIL ACCOUNT*. I recommend using a username that incorporates your actual name. Ensure that you use your Exeter email account as this is how I will send you tutorial exercises etc.

2. Install Git. This is the basic version control system. GitHub is the central cloud based web store for your code, and front end that we will use. Follow the guide. You don't need to do the R related components mentioned here, we'll deal with all of that via VS Code. Note that these instructions require use of the Mac / Windows/ Linux Terminal as mentioned above.

3. We need to set up Git to use your GitHub credentials. Open a Terminal Window, and enter the following, replacing the 'myusername' and 'myemail' with the appropriate values:

   ```
   git config --global user.name "myusername"
   git config --global user.email "myemail@exeter.ac.uk"
   ```

   Check these by typing:

   ```
   git config --global --list
   ```

4. Install GitHub Desktop. This removes the need to use the Terminal / Command line in future for most GIT commands. GitHub Desktop is here.

5. When we start using GitHub, we will need the VSCode Git extention, this will be available for you via a link when required.