Internet Technology (M) – 2021
Setting up your working environment

You need to follow the instructions below to set up your working environment on the Glasgow Anywhere Desktop (GA Desktop) and skip the instructions provided in Chapter 2 of the Tango with Django Book. However, you still need to read Chapter 2 to consolidate your knowledge about the technologies used in this course.

To access the GA Desktop, click on the following link:

https://www.gla.ac.uk/myglasgow/anywhere/desktop/

and click on "Connect to Glasgow Anywhere Desktop via your web browser". You will need to enter your login details. After you have logged on, scroll down to find the COSE Desktop under the College of Science Engineering tab. Double-click on the COSE Desktop icon to open a remote desktop instance.

These instructions can also be used to set up your environment in your PC/Mac by downloading Anaconda for Windows, macOS or Linux from:

https://www.anaconda.com/download/

After downloading the "Anaconda Individual Edition" installer, double-click on it and follow the on-screen instructions.

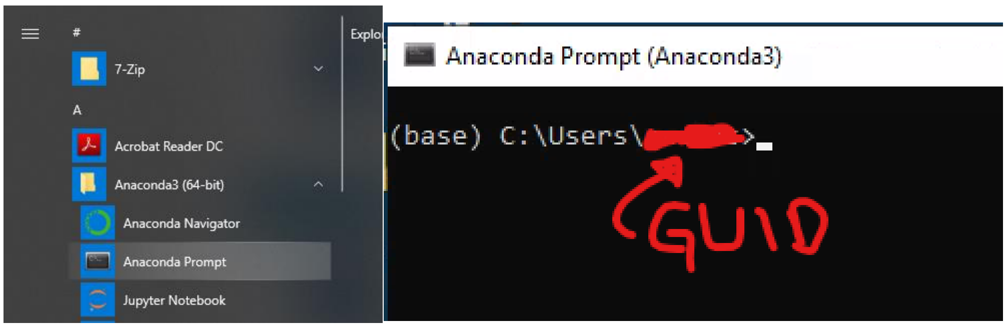
Do not copy/paste any command in this document. You will get Word/PDF characters mapped to the command prompt which might not give you the desired result.

The instructions given in this guide are for a Windows machine, but the same instructions and commands are valid for macOS and Linux. Figures will be different depending on the Terminal you use in macOS and Linux.

Setting up a virtual environment

The GA Desktop already have Anaconda pre-configured. If you are using your own PC, we will assume that you have already installed Anaconda, as noted in the previous section. To begging setting up your virtual environment,

(1) Click the "🔍" symbol (magnifying glass) at the bottom left-hand corner of your GA Desktop (or your PC) and type "anaconda" into the search box; then click on "Anaconda Prompt". This will start up the Anaconda command prompt with the "base" environment, as shown in Figure 1.



```
(base) C:\Users\[redacted]>
```

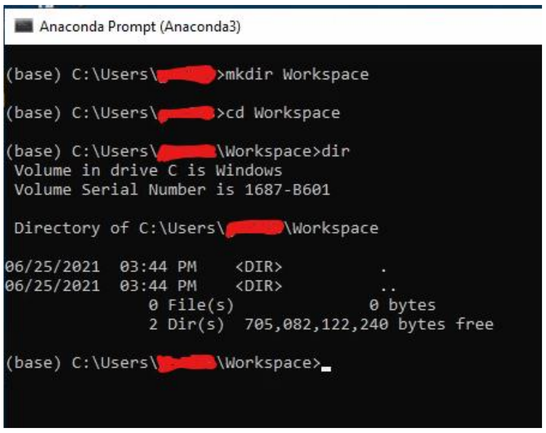
Figure 1. Anaconda Prompt (the Command Prompt on the right may be a bit different)

<u>NOTE:</u> In macOS and Linux, the Anaconda prompt doesn't exist. You can use any terminal emulator.

(2) You can list files in the current directory by entering the command "`dir`" for Windows ("`ls`" for macOS and Linux) and change directories with "`cd`".

(3) When you open the Anaconda Command Prompt, you should land into your home directory, i.e. "`C:\User\<guid>`" where `<guid>` is your student number. At this location, you should create a new folder and cd to the newly created folder with the following commands (see Figure 2):

```
mkdir Workspace
cd Workspace
```



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\[redacted]>mkdir Workspace

(base) C:\Users\[redacted]>cd Workspace

(base) C:\Users\[redacted]\Workspace>dir
 Volume in drive C is Windows
 Volume Serial Number is 1687-B601

 Directory of C:\Users\[redacted]\Workspace

06/25/2021  03:44 PM    <DIR>          .
06/25/2021  03:44 PM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  705,082,122,240 bytes free

(base) C:\Users\[redacted]\Workspace>
```

Figure 2. Creating your workspace

You may choose to create this new directory somewhere else, just make sure that it is within your home directory so it can persist between sessions! If you are using your own PC, you can create this folder wherever you like, just remember where you created it!

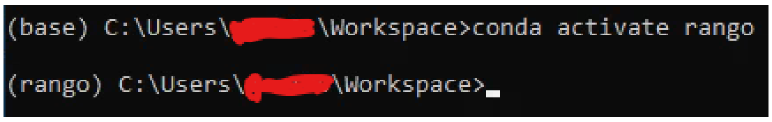
(4) Create a virtual environment for your development of the Rango application using the following command and follow the on-screen instructions:

```
conda create -n rango python=3.7.5
```

You can replace "`rango`" with another virtual environment name if you prefer – this will be useful when you start working in your group project but for now use "`rango`". The command prompt will not activate your new environment automatically, so you need to run the following command to activate it:

```
conda activate rango
```

At this point, you are now inside an Anaconda virtual environment; see Figure 3.



```
(base) C:\Users\[redacted]\Workspace>conda activate rango

(rango) C:\Users\[redacted]\Workspace>_
```

Figure 3. Rango environment activated.

(5) For Steps 5-10 to work, you need to be in your virtual environment – the command prompt window should look like Figure 3. If not, you will need to activate it. To deactivate it, you can close the window or type **`conda deactivate`.** With your virtual environment activate, now install Django using the following command:

```
conda install django==2.1.5
```

<u>We will be using Django 2.1.5 and Python 3.7.5 in ITECH, please make sure to use these versions as problems may arise because of version mismatches and these are difficult to debug and spot!</u>

(6) Install Pillow: **`conda install pillow`**

Now that you have your virtual environment created with Django and Pillow installed, you can create a project.

(7) To create a new Django project, issue the following command within `H:\Workspace`

```
django-admin startproject tango_with_django_project
```

You can replace "`tango_with_django_project`" with another project name if you prefer.

(8) Change the directory to the top-level folder of the project you have just created:

```
cd tango_with_django_project
```

(9) Run the app:

```
python manage.py runserver
```

(10) Open a web browser and visit the link http://127.0.0.1:8000/. You should see a congratulations message. Well done! Press Crtl+C in Anaconda prompt to stop the server, or close the prompts.

<u>Setting up Git</u>

The GA Desktop already has Git installed and you can use it directly in the Anaconda Command Prompt.

If you do not have Git installed on your PC, you can download the latest version from https://git-scm.com/download/win, and run the installer in Windows. Please use the default options while installing Git (i.e. press Next until you complete the installation); <u>if in doubt, ask your lab demonstrator for help during the allocated times!</u> **NOTE:** *you need to install Git for macOS* *and Linux. For macOS, please go to https://git-scm.com/download/mac*; for Linux, please use the default package manager of your distribution.

(1) Open the Anaconda prompt if you close it. If the prompt is still open, skip this step.

(2) You can now use the git commands. CD into your working directory.

(3) Set your user name:

```
git config --global user.name "Firstname Surname"
```

(4) Set your user email:

```
git config --global user.email "2099999z@student.gla.ac.uk"
```

Now you can use clone, add, commit, push, etc on your repository. You will need a Github repository, so create one on www.github.com. Make sure to change the details highlighted in yellow to match your own details!

A Git crash course

Once you have completed the above steps, work through the appendix in TWD entitled "A Git Crash Course" to familiarise yourself with Git. Next, work through the following steps.

1. Create a Github repository for your Rango application, called

`tango_with_django_project`.

2. Change directory to `C:\User\<guid>\Workspace\tango_with_django_project` – this is where your Rango files reside in your workspace, as created earlier by `django-admin startproject`.

3. Issue the following git commands:

```
git init
git add *
git commit -m "first commit"
git remote add origin https://github.com/user/tango_with_django_project.git
git push -u origin master
```

Replace **user** in the above sequence of commands with your Github username. You have now set up a remote repository for Rango on Github and have linked it to your local repository in your workspace.

It is very important to ensure that your git repository is set up correctly. Inside `C:\User\<guid>\Workspace`, you should have a folder tango_with_django_project. Inthis folder, you should have "manage.py" and another folder tango_with_django_project folder. Inside the latter folder, you should initially have four files: __init__.py, settings.py, urls.py and wsgi.py. Please do compare if your remote repository with the example one at https://github.com/wad2/tango_with_django_project.

If you choose not to create a ".gitignore" file while creating your GitHub repository, you will end up with extra files in your remote repository, e.g. `db.sqlite3`, `__pycache__`, Python compiled scripts (`.pyc`), etc. If you haven't created this file, don't worry! You just need to create it for your repository (this file should be created where "manage.py" is located) and add each file name you want to ignore. You can create this file using the Windows File manager by right-clicking inside the folder and choose to create a new Notepad file; type the name and make sure it does not have a file extension. For example, to ignore both db.sqlite3 and __pychache__ type inside .gitignore the following:

```
db.sqlite3
__pycache__/
*.pyc
```

Next time you "add" and "commit" your work, Git will ignore the file names and file extensions found in .gitignore. Consequently, Git will remove them from your local and remote repository. The latter will only affect the latest commit, but, as noted above, this will not affect the automated testing!

If your git repository structure is not analogous to the structure described in the previous paragraph, **do not go any further**. Instead, repeat the above steps again as necessary. It is very important that your directory structure is set up correctly, otherwise, the automated tests that will be run on your Rango app later in the course will likely fail. It is much better to take the time now to complete this step correctly rather than to persist with an incorrect directory structure as this could make life very difficult for you later (as several students have found in previous years).

Setting up Visual Studio Code

There are many code editors, and you are free to choose the one that best works for you. For Internet Technology, we recommend using Visual Studio Code (VS Code). VS Code is already installed in the GA Desktop. If you are using your PC, you can install it via Anaconda Navigator or download it from https://code.visualstudio.com/.

If this is the first time you are launching VS Code, you need to make it work with Anaconda's virtual environments. For this, please follow the instructions below.

(1) Click the "🔎" symbol (magnifying glass) at the bottom left-hand corner and type "Visual Studio Code" to open an instance of VS Code. If you have the Anaconda Command Prompt open, type "code" and press enter.

(2) Now, click on "Add workspace Folder" (just below Start, see Figure 4 next page) and in the "Add Folder to Workspace" window, navigate to `C:\User\<guid>\Workspace\` and click on the "tango_with_django_project" folder. Finally, click Add at the left-bottom corner of the window. You will see that the "tango_with_django_project" folder has been added to the VS Code workspace. Click on it to expand its contents; you should get something like Figure 4 (next page).

(3) Now click on manage.py to open the file. You will notice a small window pops out on the left-bottom side of VS Code asking you whether to install a recommended extension for Python. Click install, and this will take you to VS Code plugin tab.

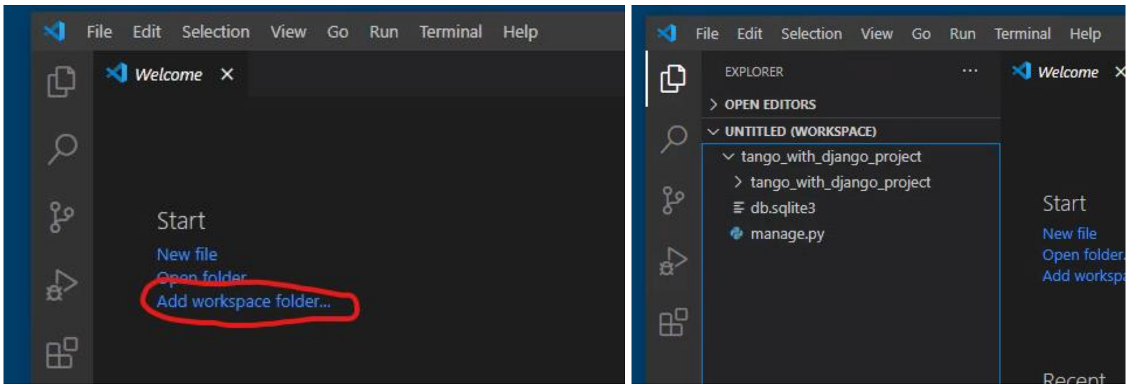
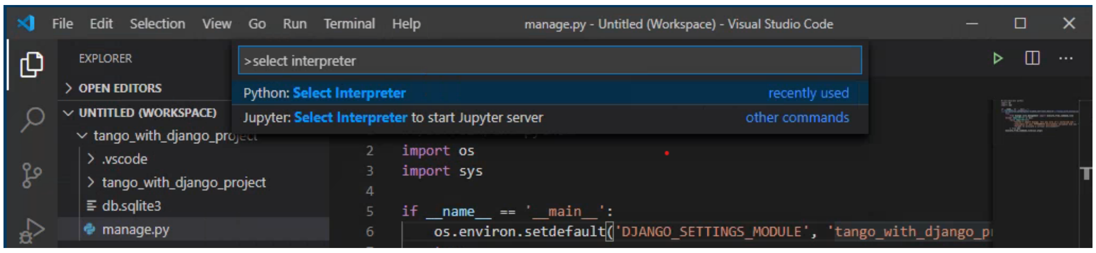


Figure 4. VS Code workspace

(4) To set your Anaconda virtual environment in VS Code, click on View (in the top menu) and select “Command Palette…” (in GA Desktop, keyboard shortcuts may not work). This will open a drop-down terminal where you should type “select interpreter” (see the top figure in Figure 5).


File Edit Selection View Go Run Terminal Help
manage.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
OPEN EDITORS
UNTITLED (WORKSPACE)
tango_with_django_project
.vscode
tango_with_django_project
db.sqlite3
manage.py
recently used
other commands

>select interpreter

Python: Select Interpreter

Jupyter: Select Interpreter to start Jupyter server

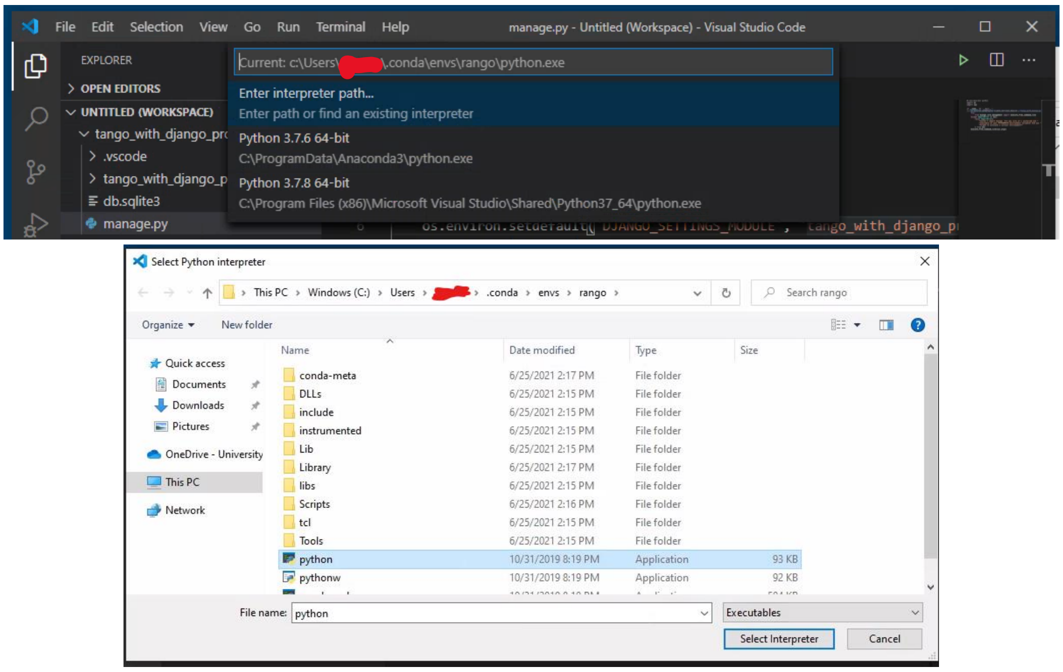
```
import os
import sys

if __name__ == '__main__':
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'tango_with_django_p
```

Figure 6. Command Palette

(5) Now, click on “tango_with_django_project”, then “Enter interpreter path…” (see Figure 6) and then click on “Find”. This will open a window where you need to navigate to the rango environment. This environment is located at `C:\Users\<guid>\.conda\envs\rango`. In this folder, click on “python” and then “Select Interpreter”.

Note: Some GA Desktops may show your rango virtual environment, so just click on it to set the interpreter.



Current: c:\Users\[redacted]\.conda\envs\rango\python.exe

Enter interpreter path...
Enter path or find an existing interpreter
Python 3.7.6 64-bit
C:\ProgramData\Anaconda3\python.exe
Python 3.7.8 64-bit
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe

Name	Date modified	Type	Size
conda-meta	6/25/2021 2:17 PM	File folder	
DLLs	6/25/2021 2:15 PM	File folder	
include	6/25/2021 2:15 PM	File folder	
instrumented	6/25/2021 2:15 PM	File folder	
Lib	6/25/2021 2:15 PM	File folder	
Library	6/25/2021 2:17 PM	File folder	
libs	6/25/2021 2:15 PM	File folder	
Scripts	6/25/2021 2:16 PM	File folder	
tcl	6/25/2021 2:15 PM	File folder	
Tools	6/25/2021 2:15 PM	File folder	
python	10/31/2019 8:19 PM	Application	93 KB
pythonw	10/31/2019 8:19 PM	Application	92 KB

File name: python

Figure 6. Setting a Python interpreter

(6) To verify if you have successfully set your virtual environment, click on Terminal (top menu, Figure 7) and you should see that a terminal opens at the bottom of your VS Code window and automatically activates your environment.

You can now take advantage of VS Code features such as code completion, linters, etc. This will work for all Python packages you install in your Rango virtual environment.

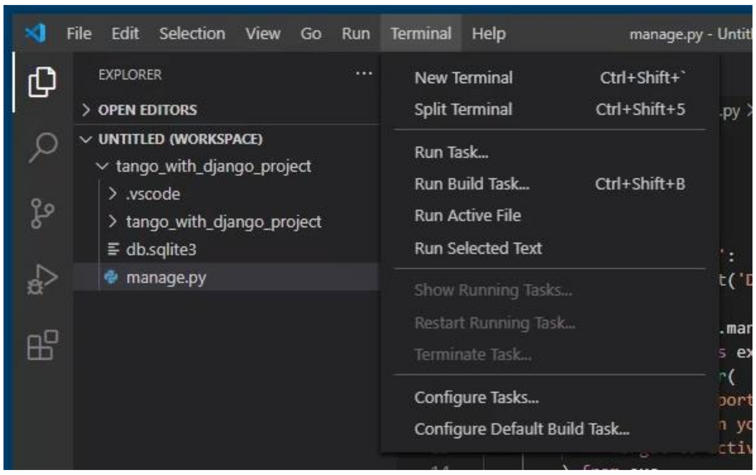
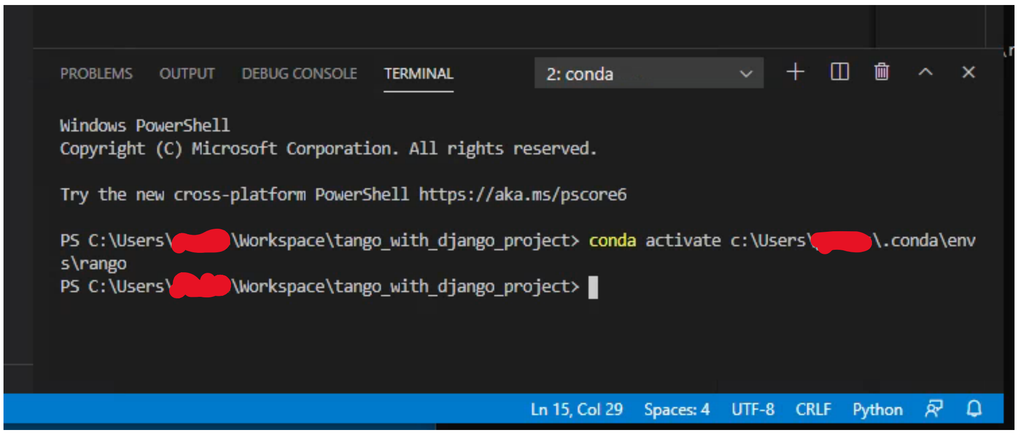


Figure 7. New Terminal



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\[redacted]\Workspace\tango_with_django_project> conda activate c:\Users\[redacted]\.conda\env
s\rango
PS C:\Users\[redacted]\Workspace\tango_with_django_project>
```

Figure 8. Terminal tab at the bottom of VS Code.