Internet Technology (M)
Setting up your working environment on the lab machines / Glasgow Desktop Anywhere

You need to follow the instructions below to setup your working environment on the lab machines / Glasgow Desktop Anywhere.

Please note that these instructions can be used to setup your environment in your PC/Mac by downloading Anaconda for Windows, macOS or Linux from https://www.anaconda.com/download/

> **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 lab machines already have Anaconda pre-configured (ignore the install instructions in Chapter 2 of the TWD book). Instead you can do the following.

(1) Click the "🔍" symbol (magnifying glass) at the bottom left-hand corner of your screen 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.
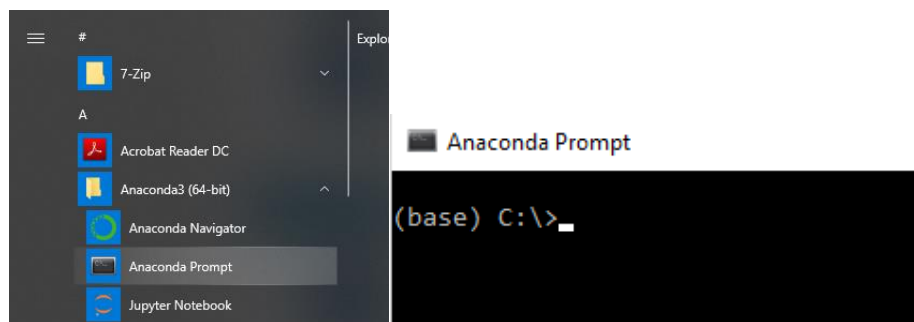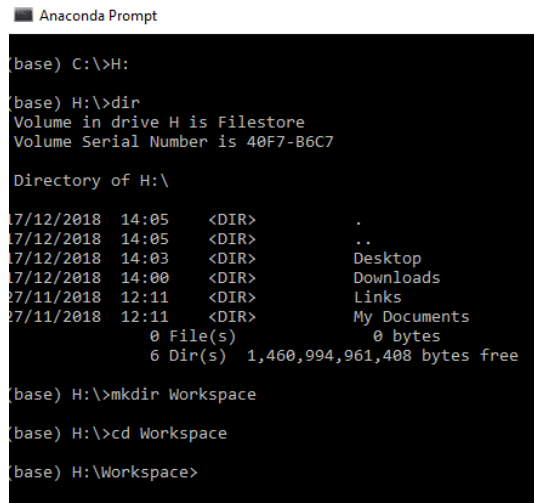

Figure 1. Anaconda Prompt

NOTE: 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) Switch to your home directory by entering "H:" at the command prompt (your network drive as provided by the School) and create a new folder and cd to the newly create folder with the following commands (see Figure 2):

```
mkdir Workspace
cd Workspace
```

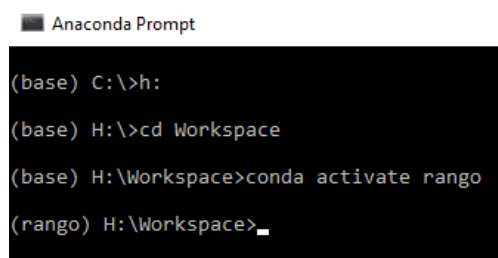

Figure 2. Creating your workspace

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

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

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.

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 2. If not, you will need to activate it. To deactivate it, you can

simply close the window or type **conda deactivate.** Install Django using the following command:

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

We will be using Django 2.1.5 and Python 3.7.5 in ITECH for your own study of the Tango with Django book. For your group coursework you can use more recent versions.

If you want to use a more recent version of Django (e.g. >=5.0), you will also need a more recent version of Python (e.g. >=3.10). For this, you will also need to set up a new conda environment. You can find the combinations of Python and Django versions here: https://docs.djangoproject.com/en/dev/faq/install/#what-python-version-can-i-use-with-django

(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
```

(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 simply close the prompts.

A Git crash course

Work through the appendix in TWD entitled "A Git Crash Course" to familiarise yourself with Git. Next, work through the following steps.

To use Git on the lab machines you need to use a program called "Git Bash" in Windows. **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.

If git is not installed in the lab machines, you can install it in your user profile. So 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); if in doubt, ask your demonstrator for help!

(1) Open Anaconda prompt if you closed 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 "<YourFirstname YourSurname>"
```

(4) Set your user email:

```
git config --global user.email "<yourEmail>@student.gla.ac.uk"
```

(5) Create a ".gitignore" file; if you don't you will end up with extra files in your remote GitHub repository, e.g. `db.sqlite3`, `__pycache__`, Python compiled scripts (`.pyc`), etc. You just need to create a ".gitignore" file for your repository (this file should be created where "manage.py" is located) and add each file name you want to ignore. For example, to ignore both db.sqlite3 and __pychache__ type the following inside .gitignore:

```
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.

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 (if you don't already have one).

Setting up a GitHub repository

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

```
tango_with_django_project
```

2. Change directory to `H:\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 H:\workspace you should have a folder tango_with_django_project. Inside that folder, you should have a file manage.py and another folder tango_with_django_project. Inside the latter folder, you should initially have four files: __init__.py, settings.py, urls.py and wsgi.py. Please do compare if your own remote repository with the example one at https://github.com/wad2/tango_with_django_project.

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 can be run on Rango app later in the course will be likely to 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.