

GitHub

1. Instantiate a GitHub repository:
 - a. Log in to your GitHub account and go to the Repositories tab.
 - b. Click on New and give the repository a unique name, e.g. “webscraping-workshop”. Add a short description. Leave it on Public. Tick the boxes of adding a README file and a .gitignore (choose Python as template).
 - c. Click Create repository.
2. Open GitHub Desktop:
 - a. Go to File → Clone repository
 - b. Select the repository you just created, e.g. “webscraping-workshop”
 - c. Leave the local path on default, i.e. there will be a GitHub folder in your documents directory.
 - d. Click Clone
 - e. Navigate to the directory manually and confirm that there is a new folder “webscraping-workshop” containing a README.md, a .gitignore as well as a (hidden) .git folder.

Anaconda and Jupyter Notebook

1. You have successfully installed the Anaconda distribution, great!
2. You should be running Python version 3.8 – you can easily check by opening the Anaconda Prompt and type `python --version`
3. First task is to create a virtual environment:
 - a. Type `conda create --name webscraping` (or any other name that allows you to easily identify the environment) and confirm to proceed by entering `y`
 - b. Activate the virtual environment by entering `(conda) activate webscraping`. You should observe that the path prefix changes from `base` to your environment.
 - c. Check which modules are already pre-installed in your virtual environment by typing `pip list` while the environment is activated.
4. Now we will add a Jupyter Notebook Kernel to the environment that allows us to easily switch between the Kernels and environments in the notebook:
 - a. Any additional package / module in Python is installed via `pip` or `conda` (if we’re using the Anaconda distribution)
 - b. To install the additional module type `conda install ipykernel` in your still activated webscraping environment. The prompt will display the progress of further dependencies that are being downloaded and installed. The process has finished after Executing transaction: done is shown.
 - c. Also install Jupyter via the command `pip install jupyter`
 - d. Now re-do `pip list` – you should find more modules appearing in the list of installed modules.
 - e. To create the Jupyter kernel type `python -m ipykernel install --user --name webscraping --display-name “Webscraping Workshop”`
 - f. It is common practice to provide a complete specification of the modules contained in a virtual environment in a requirements format (.txt). To obtain this file navigate to the local GitHub directory by typing `“cd”` followed by the path (you can copy-paste this from the file explorer line) into the Anaconda Prompt and then type `python -m pip freeze > requirements.txt`. The requirements.txt can then be used from another

user when reproducing any code that runs in your virtual environment without bothering about incomplete documentation of dependencies and versions. To do so, the user needs to have activated a raw virtual environment and provide the requirements.txt by the command `python -m pip install -r requirements.txt`

5. Now it's time to open a Jupyter Notebook for the first time:
 - a. In the activated environment webscraping type jupyter notebook in the prompt
 - b. The prompt running in the background will open a new tab in your default browser showing you a file manager like interface. Navigate to Documents → GitHub → webscraping-workshop.
 - c. Create a Jupyter Notebook by clicking on New and select the Kernel "Webscraping Workshop" you created before.
 - d. Rename the notebook and create a markdown cell containing a header and a code cell printing "Hello World!", save the file.
6. Modify the README.md in a text editor of your choice (I use Sublime Text), you may find some helpful formatting hints [here](#) and save the file.
7. Open GitHub Desktop and observe which files are affected from your most recent changes. Type a summarizing phrase into the required field and provide a short description what the commit is about.
8. Click "Commit to main" and then "Push origin" (upper right hand corner). Go back to your repository in the browser and check that the local changes have been applied to the public repository. Congratulations, you have just made your first contribution and are now part of the world's largest Open-Source community!

Optional:

1. In case your auto-complete in Jupyter is disabled (as it was for me first) you can restore it by installing/downgrading the jedi package by typing (in the activated environment) `conda install jedi==0.17.2`