

## Python omgeving opzetten

1) We creëren een Python omgeving ai1tutorials:

```
(base) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\tutorials>conda create -n ai1tutorials python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Anaconda3\envs\ai1tutorials

  added / updated specs:
    - python=3.7

The following NEW packages will be INSTALLED:

  ca-certificates  pkgs/main/win-64::ca-certificates-2020.1.1-0
  certifi          pkgs/main/win-64::certifi-2019.11.28-py37_0
  openssl          pkgs/main/win-64::openssl-1.1.1d-he774522_4
  pip              pkgs/main/win-64::pip-20.0.2-py37_1
  python           pkgs/main/win-64::python-3.7.6-h60c2a47_2
  setuptools       pkgs/main/win-64::setuptools-45.2.0-py37_0
  sqlite           pkgs/main/win-64::sqlite-3.31.1-he774522_0
  vc               pkgs/main/win-64::vc-14.1-h0510ff6_4
  vs2015_runtime   pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_1
  wheel            pkgs/main/win-64::wheel-0.34.2-py37_0
  wincertstore     pkgs/main/win-64::wincertstore-0.2-py37_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate ai1tutorials
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\tutorials>conda activate ai1tutorials
(ai1tutorials) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\tutorials>
```

2) We installeren Jupyter Notebook in deze omgeving:

```
(ai1tutorials) D:\TIBL\artificial_intelligence_kdg\artificial_intelligence_1\tutorials>conda install jupyter
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Anaconda3\envs\ai1tutorials

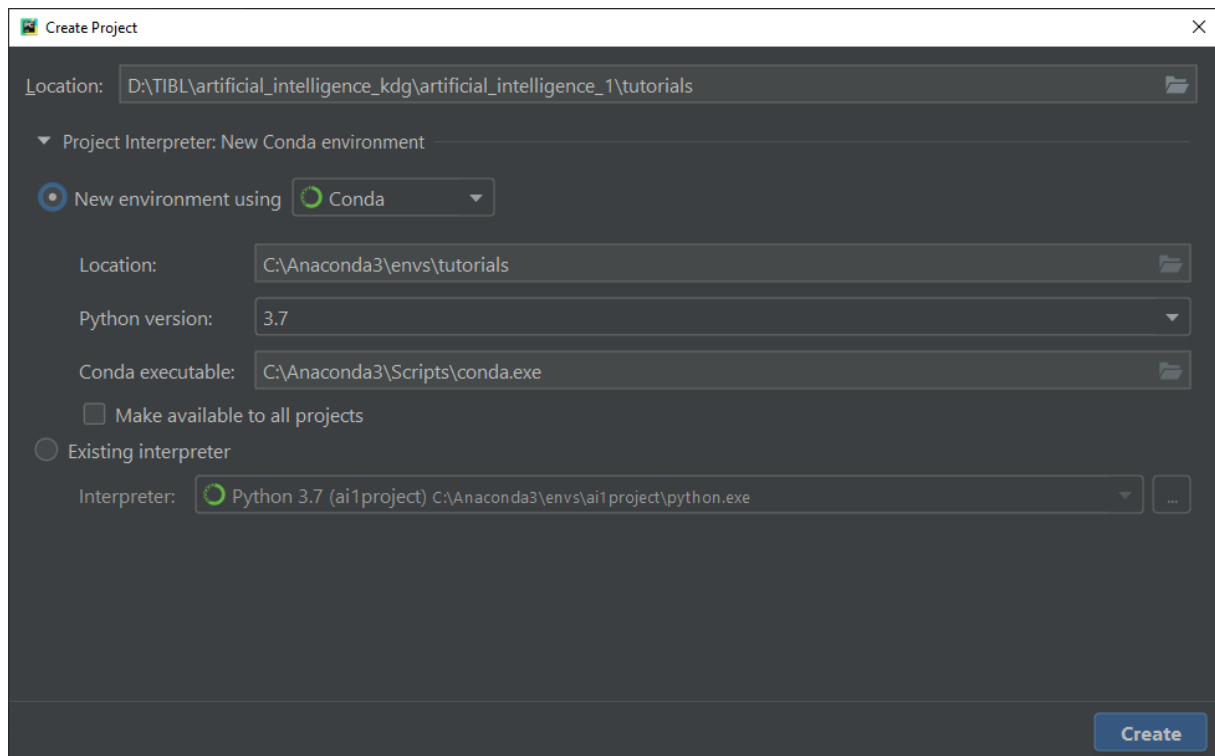
  added / updated specs:
    - jupyter

The following packages will be downloaded:
```

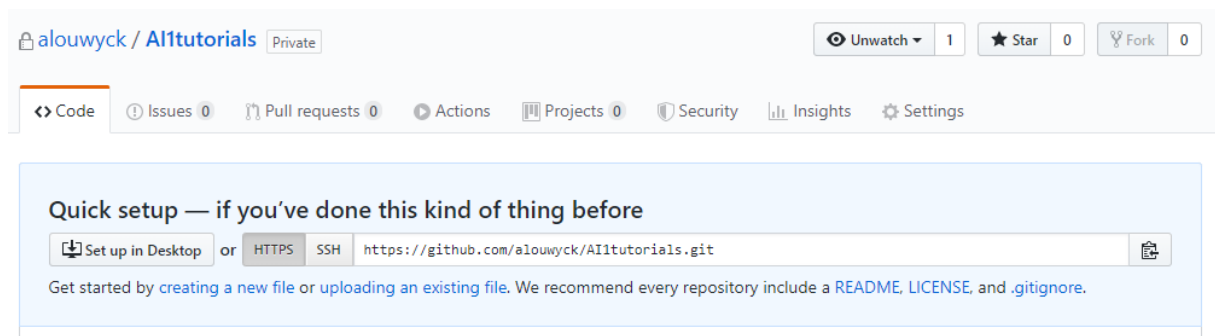
Op dezelfde manier installeren we ook de packages numpy en matplotlib:

Commando's: “conda install numpy” en “conda install matplotlib”

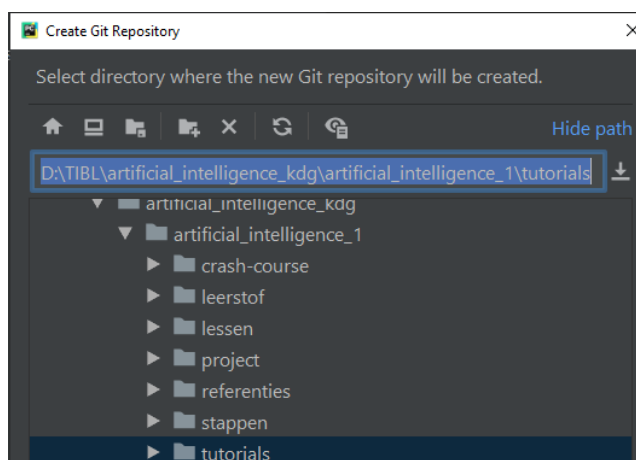
3) We starten PyCharm op en creëren via menu “File > New Project...” een project in dezelfde subfolder:



4) We creëren een nieuwe private repository op GitHub:

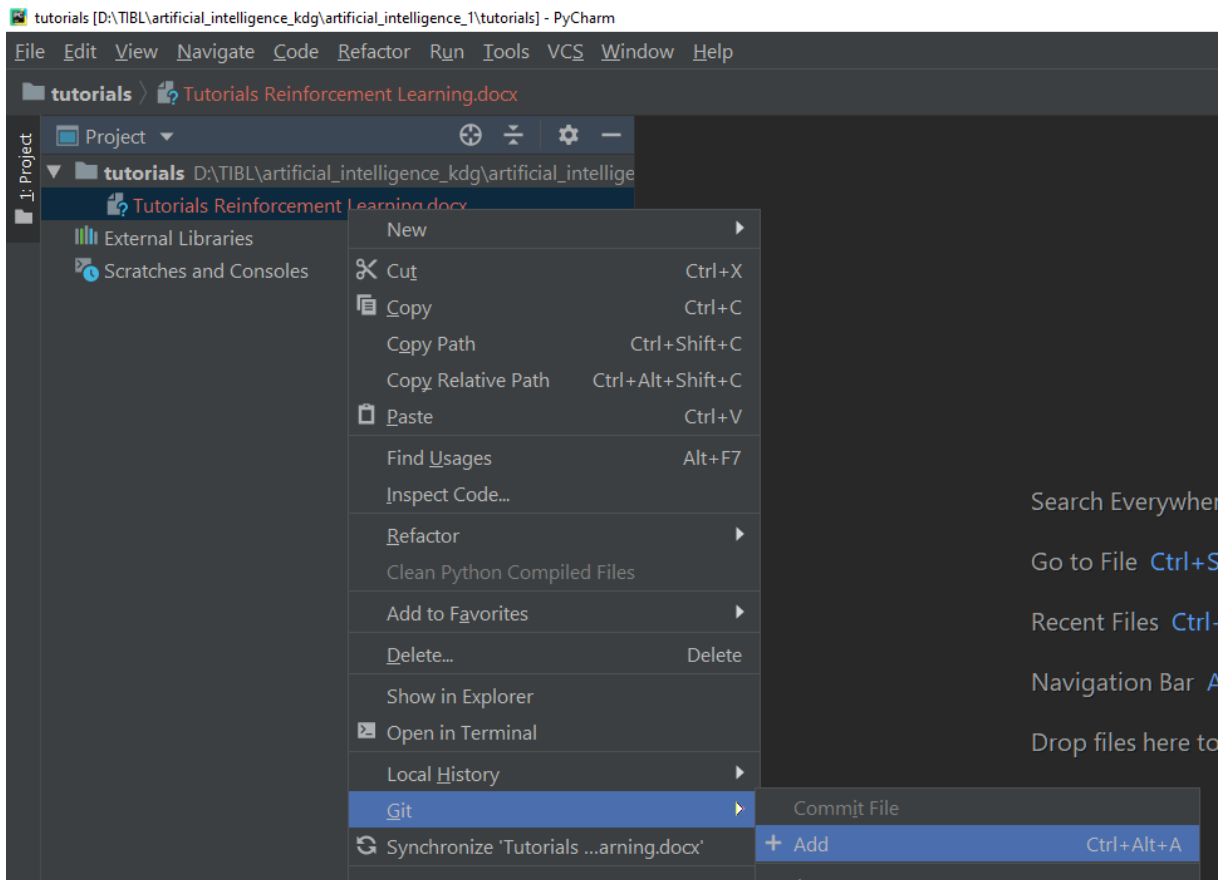


5) In PyCharm creëren we een lokale repo in de projectfolder via menu “VCS > Import into Version Control... > Create Git Repository...”:

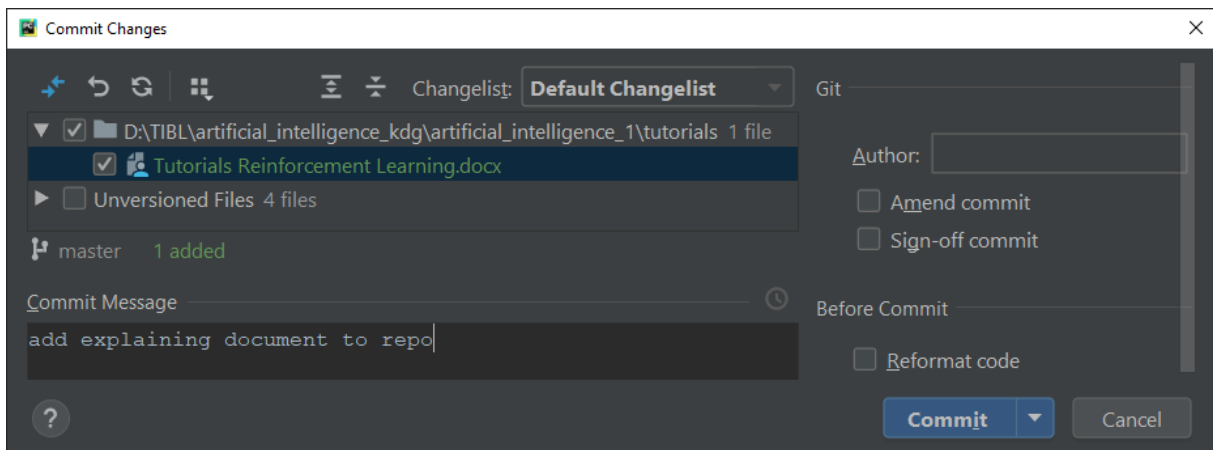


6) We gaan nu dit document inchecken (= commiten + pushen):

We rechtsklikken op dit document in het “Project” navigatiescherm en klikken dan op “Git > Add”:



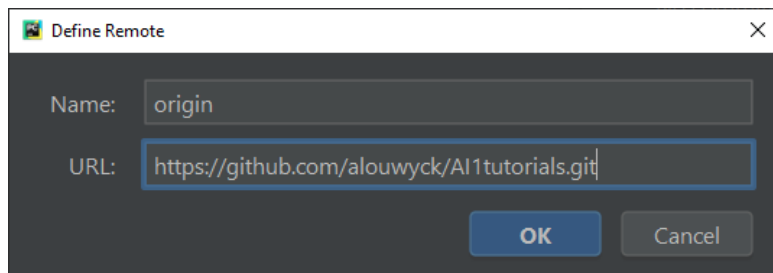
We klikken op het menu “VCS > Commit...” en krijgen dit scherm:



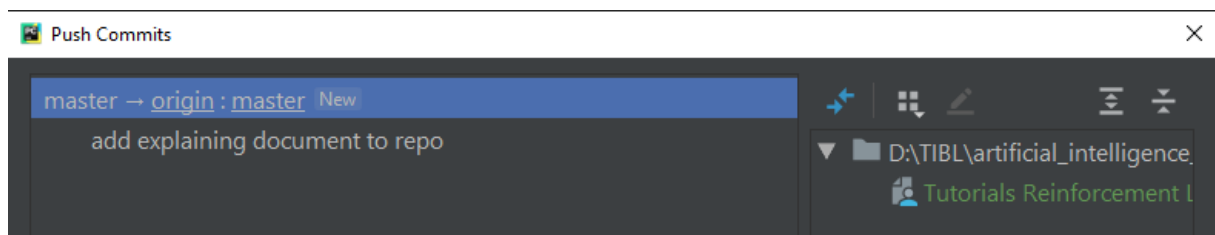
We klikken op het menu “VCS > Git > Push...” en krijgen dit scherm:



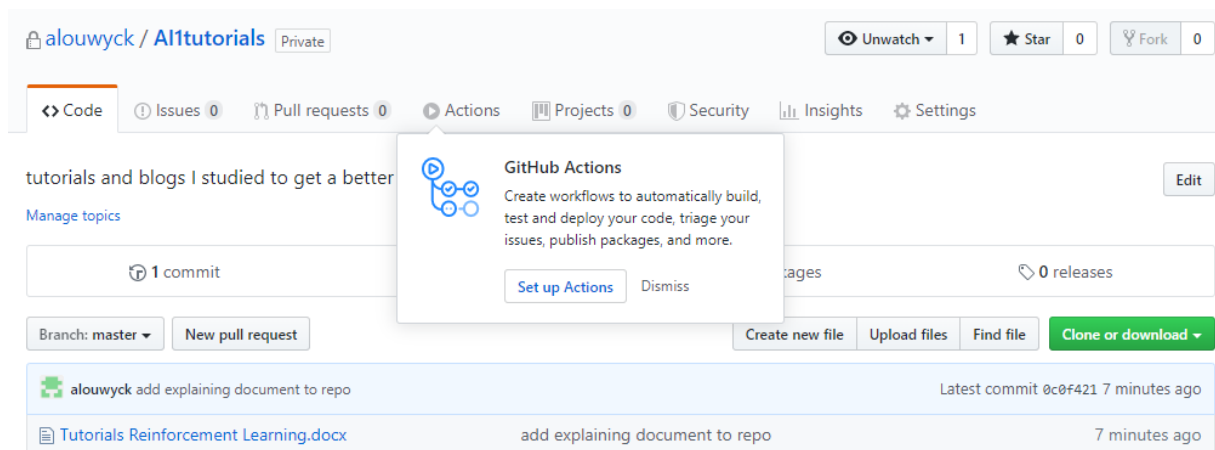
We klikken op “Define remote” en geven de URL van onze GitHub repo op:



Als de connectie is gelukt, dan wordt de commit weergegeven die we net hebben gedaan:

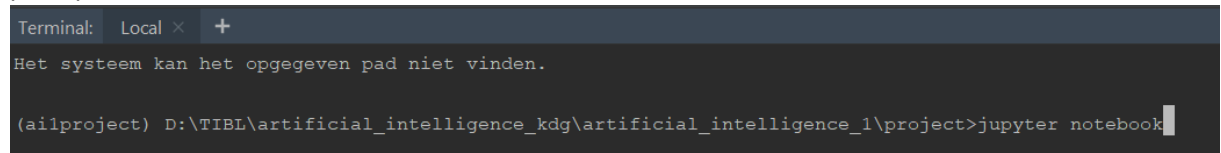


We drukken op “Push”. In GitHub zien we nu onder “Code” dat het document inderdaad is toegevoegd:



## 7) Opstarten van Jupyter Notebook:

Onderaan een terminal venster openen en het commando “jupyter notebook” uitvoeren aan de prompt:



Normaal wordt automatisch de browser geopend die met de localhost connecteert. Daar kan een nieuwe notebook worden geopend, die op dezelfde manier werkt als een python script.

## 8) Runnen van scripts in Pycharm:

Om code te debuggen is het wellicht handiger om in Pycharm een script te schrijven en te runnen. De eerste keer dat je een script wil runnen, moet je via menu “Run” gaan en daar het item “Run...” kiezen. Er opent zich een venster waar je het script kan selecteren. Na je keuze

wordt het script ook direct uitgevoerd in het “run” venster onderaan. Wil je het script opnieuw runnen, dan kan je gewoon op de groene pijl in het snelmenu klikken.

Je kan het script natuurlijk ook in de “Python console” runnen. Deze console open je onderaan (waar je ook de terminal en run vensters kan openen). Daar run je het script door het te importeren met het “import” commando.

Een python-file met functies en klassen importeren kan ook. Dit noemt men een module. Na inladen blijven de definities van die functies en klassen in het geheugen zitten. Dus als je wijzigingen in die module aanbrengt, dan moet je die opnieuw importeren. Dat kan door de console opnieuw op te starten. Maar handiger is het om gebruik te maken van package “importlib”:

```
In[113]: import importlib
In[114]: import ReinforcementLearning
In[115]: importlib.reload(ReinforcementLearning)
Backend Qt5Agg is interactive backend. Turning interactive mode on.
Out[115]: <module 'ReinforcementLearning' from 'D:\\TIBL\\artificial_
In[116]: from ReinforcementLearning import *
```