To create new environment in anaconda:

conda create –n test python=3.6

to activate this environment

conda activate test

to deactivate current environment

conda deactivate

to see the installed packages list

conda list

to see the all conda environment lists

conda env list

to remove the just created environment

conda env remove –n test

to install Django/To install this package with conda run:

conda install -c anaconda django

conda install -c conda-forge django-crispy-forms

conda install -c conda-forge django-allauth.

To choose interpreter:

Ctrl+shift+P

To see the different kinds of fields you can use in a model, see the Django Model Field Reference at https://docs.djangoproject.com/en/2.2/ref/models /fields/. You won’t need all the information right now, but it will be extremely useful when you’re developing your own apps.

we need to tell Django to modify the database so it can store information related to the model Topic. From the terminal, run the following command:

python manage.py makemigrations learning\_logs

Now we’ll apply this migration and have Django modify the database for us:

python manage.py migrate

Whenever we want to modify the data that Learning Log manages, we’ll follow these three steps: modify models.py, call makemigrations on learning\_logs, and tell Django to migrate the project.

To create a superuser in Django, enter the following command and respond to the prompts:

python manage.py createsuperuser

You can simply reset the database instead of migrating, but that will lose all existing data. It’s good practice to learn how to migrate a database while maintaining the integrity of users’ data. If you do want to start with a fresh database, issue the command **python manage.py flush** to rebuild the database structure. You’ll have to create a new superuser, and all of your data will be gone.

Django bootstrap4 anaconda installation code:

conda install -c conda-forge django-bootstrap4

To see the templates Bootstrap offers, go to <https://getbootstrap.com/>

conda install -c conda-forge django-heroku

conda install -c conda-forge matplotlib

git config --global user.email [shiva.shres1998@gmail.com](mailto:shiva.shres1998@gmail.com)

conda list --export > requirements.txt

better

pip freeze > requirements.txt

While studying this book, definitely refer to the scikit-learn website for more indepth documentation of the classes and functions, and many examples. There is also a video course created by Andreas Müller, “Advanced Machine Learning with scikitlearn,” that supplements this book. You can find it at http://bit.ly/ advanced\_machine\_learning\_scikit-learn.

Supplemental material (code examples, IPython notebooks, etc.) is available for download at <https://github.com/amueller/introduction_to_ml_with_python>.

Userguide for machine learning : <https://scikit-learn.org/stable/user_guide.html>

pip install numpy scipy matplotlib ipython scikit-learn pandas

conda install numpy scipy scikit-learn matplotlib pandas pillow graphviz python-graphviz

conda install -c anaconda ipython

Throughout the book we make ample use of NumPy, matplotlib and pandas. All the code will assume the following imports:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd import mglearn

conda install -c plotly plotly

conda install -c intel scikit-learn

conda install -c conda-forge textract  
conda install -c conda-forge/label/cf201901 textract  
conda install -c conda-forge/label/cf202003 textract

conda install -c conda-forge tweepy

conda install -c anaconda pandas

<https://tweepy.readthedocs.io/en/v3.8.0/api.html>

conda install -c anaconda nltk

<https://www.geeksforgeeks.org/regular-expression-python-examples-set-1/>

<https://www.geeksforgeeks.org/join-function-python/>

conda install -c conda-forge wordcloud

<https://codereview.stackexchange.com/questions/178865/rolling-hash-algorithm>

<https://www.numbers.education/list.html>

board = []

for i in range(6): # create a list with nested lists

board.append([])

for n in range(6):

board[i].append("O") # fills nested lists with data

Result:

[['O', 'O', 'O', 'O', 'O', 'O'],

['O', 'O', 'O', 'O', 'O', 'O'],

['O', 'O', 'O', 'O', 'O', 'O'],

['O', 'O', 'O', 'O', 'O', 'O'],

['O', 'O', 'O', 'O', 'O', 'O'],

['O', 'O', 'O', 'O', 'O', 'O']]

<https://docs.anaconda.com/anaconda/packages/py3.7_win-64/>

**import** numpy **as** np

array1 = np.array([2, 2, 2, 0, 2, 0, 2])

**print** np.where(array1==0, 1, array1)

Python important sites:

<https://awesome-python.com/>

<https://docs.scipy.org/doc/numpy/reference/arrays.ndarray.html>

<https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html>

conda install -c anaconda html5lib

conda install -c anaconda chardet

conda install -c anaconda beautifulsoup4

$ git config --global user.name "Your Name" $ git config --global user.email "yourname@email.com"

git init

git status

git add .

git commit -m “initial commit”

git remote add origin <https://github.com/shivstha/Web-Scraping.git>

git push -u origin master

heroku run python manage.py createsuperuser

Django features deprecation warnings299 that can and should be run for each new release by typing python -Wa manage.py test.

Even better there is a command we can run to automate Django’s recommendations, python manage.py check --deploy, that will check if a project is deployment ready. It uses the Django system check framework301 which can be used to customize similar commands in mature projects.

<https://github.com/PacktPublishing/>

<https://docs.python.org/3/howto/regex.html>

conda install -c conda-forge django-taggit

<https://docs.djangoproject.com/en/3.0/ref/templates/builtins/>

<https://docs.djangoproject.com/en/3.0/ref/templates/builtins/#built-in-filter-reference>