

# Project based Python Class – Website Full Stack Development (G8-G14)

# Class 1 Requirement for class

- Install python, pycharm : <https://blog.lemonthy.com/?p=52>
- Django Cheat sheet link : <https://lemonthy.ca/python/django/django%20cheat%20sheet.pdf>
- Django for beginner :
  - <https://lemonthy.ca/python/django/django1.pdf>
- Minimum python basic knowledge for django:
  - <https://lemonthy.ca/python/django/django2.pdf>
- Django doc <https://docs.djangoproject.com/en/3.1/> pdf
- Download <https://lemonthy.ca/class/html/> with <https://lemonthy.ca/python/game/14mario/0001download.py>
- Option: Download atom and install
- Homework: <https://lemonthy.ca/class/> finish
- Homework: <https://lemonthy.com/> crawler must finish level2
- Homework: register github user <https://github.com/>
- Homework: register pythonanywhere <https://www.pythonanywhere.com/>
- Option: download and install sourcetree <https://www.sourcetreeapp.com/>

# Class 2 git Django static

- Create project
  - `django-admin startproject dj02`
  - `cd dj02`
  - `python manage.py startapp app01`
- edit `setting.py` add `app01`,
- edit url with `views.home`,
- Edit `views.py` `home()` ,
- edit `templates/home.html`
- Install git, <https://git-scm.com/downloads>
  - `git init` (only run 1 time in begin of project)
  - `git status` (repeat for each step to see current file status)
  - `git add .` (when you have new file added or changed, run this step, before and after this step- git status file color will change from red to green)
  - `git commit -m "initial setup"` (commit your code with any description for every new change)
  - `git config --global user.email you@example.com` (if you get error of later git push, you may need setup your email and username for commit, run one time is enough)
  - `git config --global user.name "Your Name"`
- register github user <https://github.com/> create repository
  - `git remote add origin https://github.com/seanxuncaoj/dj02.git`
  - `git branch -M main`
  - `git push -u origin main` (later step push new change , use git push is enough)
  - `git config --global credential.helper store` (if you want save your username and password that you don't want input each time push to private github, you can do this)
- `.gitignore` -- Google [gitignore.io](https://gitignore.io) with Django
- [pythonanywhere.com help/Deploying an existing Django project on PythonAnywhere](https://pythonanywhere.com/help/Deploying-an-existing-Django-project-on-PythonAnywhere)
- Create static folder with file
- Homework: [lemonthy.com](https://lemonthy.com)
- Homework: html 12 projects

# git cheat sheet

- Install git, <https://git-scm.com/downloads>
  - git init (only run 1 time in begin of project)
  - git status (repeat for each step to see current file status)
  - git add . (when you have new file added or changed, run this step, before and after this step- **git status** file color will change from red to green)
  - git commit -m "initial setup" (commit your code with any description for every new change, **git status** will change from green color to empty)
  - git config --global user.email [you@example.com](mailto:you@example.com) (if you get error of later git push, you may need setup your email and username for commit, run one time is enough)
  - git config --global user.name "Your Name"
- register github user <https://github.com/> create repository
  - git remote add origin <https://github.com/seanxuncao/dj02.git>
  - git branch -M main
  - git push -u origin main (later step push new change , use git push is enough)
  - git config --global credential.helper store (if you want save your username and password that you don't want input each time push to private github, you can do this)
- .gitignore -- Google gitignore.io with Django
- git push / git pull / git clone
  - git clone <https://github.com/seanxuncao/dj02.git> (For new project download use git clone)
  - git pull (For pull new changes from git website)
  - git push (for new change after git commit , use git push to push to github)
- Google git cheat sheet <https://lemonthy.ca/python/django/git%20cheat%20sheet.pdf>

# virtualenv cheat sheet

- Install virtualenv
  - `pip install virtualenv`
  - If you are under linux box, your virtualenv tools is different base on linux server. E.g.
  - Pythonanywhere: use `mkvirtualenv` which is default
  - Ubuntu linux: `sudo easy_install virtualenv pip`
- Create a virtual environment
  - `virtualenv yourenv`
  - Pythonanywhere: `mkvirtualenv --python=/usr/bin/python3.7 mysite-virtualenv`
- Activate the virtual environment
  - `yourenv\Scripts\activate` (In windows)
  - `source yourenv/bin/activate` (In linux)
- Deactivate the virtual environment
  - `deactivate`
- Useful pip
  - `pip freeze > requirements.txt`
  - `pip install -r requirements.txt`

# Class 3 wordcount

- Create project
  - `django-admin startproject wordcount`
  - `cd wordcount`
- edit `setting.py` add templates
- edit url with `views.home`, `wordcount`,
- Edit `views.py` `home`, `countword`,
- edit `templates/word.html` , `count.html`
- Git init, add, commit, push, gitignore
- `pythonanywhere.com`
- Create static folder with file
- Homework: `lemonthy.com`
- Homework: html 12 projects

# Class 4 portfolio

- Create project
  - `django-admin startproject portfolio`
  - `cd portfolio`
- Startapp
  - `python manage.py startapp jobs`
  - `python manage.py startapp blog`
- edit setting.py add jobs, blog,
- edit urls with include jobs.urls, blog.urls. → edit jobs.urls.py with views.home, edit blog.urls.py with views.allblogs,
- Edit jobs.views.py home() , blog.views.py allblogs()
- edit jobs/templates/jobs/home.html , blog/templates/blog/allblog.html
- Google [bootstrap example](#) , update home.html , search navbar under docs, auto-indent / then allblog.html
- Git init, add, commit, push, gitignore
- Google [Django model fields](#) , update models.py
- pip install Pillow
- `python manage.py makemigrations` / `python manage.py migrate`
- Edit setting.py with MEDIA\_ROOT, STATIC\_ROOT
- `python manage.py collectstatic` (create folder static/static-in , static/media) , download pic in <https://lemonthy.ca/pic/py/gm12/>
- pythonanywhere.com
- Register user on [digitalocean.com](#) get \$100 credit for 60days
- [How To Set Up Django with Postgres, Nginx, and Gunicorn on Ubuntu 18.04](#) (windows download putty)
- ssh-keygen (need install git, open git bash, then run ssh-keygen)
- [download postgresql](#) (pip install psycopg2) (pip install psycopg2-binary)

# Postgresql cheat sheet

- [download postgresql](#) , if you have antivirus program, make sure it is not block postgres server install
- psql - SQL shell in windows: server localhost, Database postgres, port:5432, username:postgres , Password during installation
- Option run command in terminal: psql --username=postgres
- \u # list user
- \password # change password
- \l # list all databases
- DROP DATABASE if exists portfoliodb;
- CREATE DATABASE portfoliodb;
- \dS # show objects in databases
- \dt # show tables in databases
- \d # show tables and views in databases
- CREATE USER reddit WITH PASSWORD 'django1234';
- ALTER ROLE reddit SET client\_encoding TO 'utf8';
- ALTER ROLE reddit SET default\_transaction\_isolation TO 'read committed';
- ALTER ROLE reddit SET timezone TO 'UTC';
- GRANT ALL PRIVILEGES ON DATABASE portfoliodb TO reddit;
- \q
- pip install psycopg2 (pip uninstall psycopg2)
- pip install psycopg2-binary