Command sheet

Starting a Django project

GitHub setup

1. Create your GitHub repository through GitHub,
2. Include a .gitignore file, use the Python preset, or make your own
3. We want to add a line at the end of the .gitignore for venv/
4. Clone the repository onto your system

Venv setup

You want to make sure that you're in the directory with the local repository.

In there you'll create a new venv environment, I like to use the name venv for the environment as it makes it clear what the directory is.

Once the environment is created, you'll need to activate it on your terminal, you should see the environment name in parentheses once you've activated it. The environment should stay active while working on the Django site.

- Create environment: $ python -m venv <environment-name>
- Activate environment: $ source <environment-name>/bin/activate
- Deactivate environment: $ deactivate

Django install and initial setup

You should have your virtual environment active and you should be in the directory of your local repository. Always be mindful of where your terminal currently is when running these commands, it will not stop you from doing stupid stuff.

- Install Django: $ python -m pip install Django
- Create new website: $ django-admin startproject <project-name>
- Create new app (do this in the website directory): $ python manage.py startapp <app-name>
- Complete initial migrations: $ python manage.py migrate

Updating the models

When you make changes to any model you need to pass those changes onto the sql server.

- Create the migration script: $ python manage.py makemigrations
- Complete the migration: $ python manage.py migrate

The admin panel

- Create new superuser: $ python manage.py createsuperuser

Follow the steps given, if it alerts you that your password is insecure, you can dismiss this

- Start the server: $ python manage.py runserver

Pagination for templates

The data object passed in the context will be referred to as data

- data.has_previous #returns bool for if there is a previous page
- data.has_next #returns bool for if there is a next page
- data.number #property that holds the current page number
- data.paginator.num_pages #property that holds the total number of pages
- data.previous_page_number #property that holds the previous page number
- data.next_page_number #property that holds the next page number