

### setup

**creating a project** `django -admin startproject roject mysite`

**changing settings** `edit mysite/settings.py`

**database setup** `python3 manage.py migrate`

**run the server** `python3 manage.py runserver`

**and in browser:** `http://127.0.0.1:8000/`

**test shit in an interactive console** `python3 manage.py shell`

### apps

**creating an application** `python3 manage.py startapp blog`

**add the app name to INSTALLED\_APPS** `'blog.apps.BlogConfig'`

**models are kept in models.py**

**add, edit and delete model in admin.py**

### django queriesets

**getting the model classes** `from app_name.models import model_name`

**getting the user model** `from django.contrib.auth.models import User`

**see all objects of a certain model** `Class_name.objects.all()` (is iterable)

**create an object** `Class_name.objects.create(atribute)` don't have to save for it to be reflected

**accessing the object via attribute** `Class_name.objects.get(atribute="value")`

**filtering (can return multiple, iterable)** `Class_name.objects.filter(atribute="value")`

**filtering (contains)** `Class_name.objects.filter(title__contains="phrase")`

`published_date__lte=timezone.now()`

### django queriesets (cont)

**ordering** `Class_name.objects.order_by("attribute")` does the reverse

**you can combine ordering and filtering** `Class_name.objects.filter(conditions).order_by("attribute")`

**delete** `obj.delete()`

### forms

**import** `from django import forms`

`from .models import model_name`

### security

**login decorator** `from django.contrib.auth.decorators import login_required`

`@login_required`

### model attributes

model attribute	description	args
<code>models.CharField</code>	text with a limited number of characters	<code>max_length</code>
<code>models.TextField</code>	text field with unlimited content	
<code>models.DateTimeField</code>	datetime field	<code>default, blank, null, field</code>

### databases

**creating a database** `python3 manage.py migrate`

**update changes to database** `python3 manage.py makemigrations`

**migrate** `python3 manage.py migrate app_name`

**register database in admin** `admin.site.register(Model_name, app_name/admin.py)`

### admin

register a model `admin.site.register(Post)`

create superuser `python3 manage.py create_superuser`

### pythonanywhere deployment

1. create the api token

2. new bash console

3. in bash: `pip3 install --user python any where`

4. `python3 manage.py collectstatic`

5. to update website, go to your directory and pull code

6. reload at web

7. work on your directory

8. `python3 manage.py collectstatic`

### basic styling

get bootstrap `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">`

`<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap-theme.min.css">`

add static files make a *static* folder with css files inside *app\_name* folder

`{% load static %}`

`<link rel="stylesheet" href="{% static 'css/blog.css' %}">`

create a block `{% block content %}`

block

`{% endblock %}`

block content `{% block content %}`

content

`{% endblock %}`

connect a block to the base `{% extends 'blog/base.html' %}`

a block

to the

base

link a url within a site `a href="{% url 'url_name' pk=post.pk %}"`

within a

site

where pk stands for primary key

to use the link `path('whatever/<int:pk>', views.view_name, name='url_name')`

the link

