

# Django

\* Django installation instructions given in week 7 slide.

- pip install Django.
- python -m Django --version  
          ↑  
     module

- create empty folder and type 'cmd' in explorer bar.

3-2-6 → Sir uses this

creating django project :- ✓

- creating django project :-
- `django-admin startproject facebook2` name of project.

- Django divides the features into apps.

folder structure:

Facebook 2 (Fol)

facebook2 (Fol)

manage.py (py)

## Facebook 2

--init--.py

$$-asyi - py$$

- settings.py

---

urls.py

↳ wsgl.py

reconfig webserver  
(talk to webserver)

we will use this

~~By~~ default django uses sqlite3 DB

### Running:

cd to facebook2  
folder

type 'attrib'

## Starting Server

python manage.py

γυνή, γυνή

DB

• in settings change the timezone to Dhaka.

• " ' there is 'DEFAULT\_AUTO\_FIELD = 'django.db.models  
• BigAutoField'

what the  
type of default primary  
key will be.

STATIC\_URL = '/static/'

how urls work:

127.0.0.1:8000/url

↑  
admin

urls.py

path('admin', admin.site.urls),

Don't put '/' in the  
beginning! if put then  
127.0.0.1/admin

Creating views in django:

create views.py in facebook2 folder where all other .py  
included.

## Views.py

```
from django.http import HttpResponse
```

```
def home(request):
```

```
    return HttpResponse('<h1>Hello</h1>')
```

also containing  
header so wraps  
the body inside  
HttpResponse.

## urls.py

```
from . import views
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', views.home), ]
```

use include()  
function to  
import from other  
apps.

## investigating the request.

```
def home(request):
```

```
    return HttpResponse('-----' + str(dir(request)) + '-----')
```

conversion  
list down  
attributes

Output

### Welcome to our homepage

```
Request [COOKIES: {}, GET: META: POST: class: delattr: dict: dir:
doc: eq: format: ge: getattr: get: hash: init:
int_subclass: iter: le: lt: module: ne: new: reduce:
reduce_ex: repr: setattr: sizeof: str: subclasshook: weakref:
current_scheme_host: encoding: get_full_path: get_post: get_raw_host: get_scheme:
initialize_handlers: load_post_and_files: mark_post_parse_error: messages: read_started:
set_content_type_params: set_post: stream: upload_handlers: accepted_types: accepts: body:
build_absolute_uri: close: content_params: content_type: csrf_processing_done: encoding:
environ: get_full_path: get_full_path_info: get_host: get_port: get_raw_uri: get_signed_cookie:
headers: is_ajax: is_secure: method: parse_file_upload: path: path_info: read: readline:
readlines: resolver_match: scheme: session: upload_handlers: user]
```

`str(request.get_host())` returns `127.0.0.1:8000`

`str(request.headers)` returns headers.

using parameters in urls: `path('article/<id>', views.article)`

in `views.py` function

```
def article(request, id):  
    return HttpResponse(id)
```

it want only number `<int:id>`

Multi parameter urls and how they work.

### Example

```
from . import views  
  
urlpatterns = [  
    path('articles/2003/', views.special_case_2003),  
    path('articles/<int:year>', views.year_archive),  
    path('articles/<int:year>/<int:month>', views.month_archive),  
    path('articles/<int:year>/<int:month>/<slug:slug>', views.article_detail),  
]
```

### Notes:

- To capture a value from the URL, use angle brackets.
- Captured values can optionally include a converter type. For example, use `<int:name>` to capture an integer parameter. If a converter isn't included, any string, excluding a `/` character, is matched.
- There's no need to add a leading slash, because every URL has that. For example, it's `articles`, not `/articles`.

→ put special cases first.

like switch

case

Default values

`def ... (req, id = 1)`

if not given  
this used