

Introduction to Django

What is Django?

Django is a web framework written in the Python programming language aimed at building web applications quickly. Django is free, opened source, and maintained by Django Software Foundation.

Opinionated Frameworks

Opinionated frameworks recommend specific ways of solving problems to streaming productivity.

Starting a Development Server

A project installation comes with administrative tools to deploy a development server immediately after establishing a project.

```
python3 manage.py runserver
```

View Functions

URLs are mapped to a view function, which processes a user's web request and sends a response back.

```
def home(request):  
    return HttpResponse("This is  
a response from the server.")
```

Projects and Apps

A project is a collection of configurations and individual apps for a particular website. Each app should have a clearly defined purpose.

```
└─ myproject/  
    └─ myproject/  
        └─ myapp/
```

django-admin Utility

A project is started using the `startproject` command.

```
django-admin startproject myproject
```

manage.py

manage.py is a file that includes commands to administer a Django app.

settings.py

The **settings.py** file contains the project settings for a Django project.

Starting a Development Server

A project installation comes with administrative tools to deploy a development server immediately after establishing a project.

```
python3 manage.py runserver
```

Stopping a Django Server

A Django server can be stopped by pressing **ctrl** +

c.

Development Server Port

The development server runs locally on port 8000 by default. An alternative port can be provided as an option.

```
python3 manage.py runserver  
<optional_port_number>
```

Viewing the Project

When the development server is started on the default port, the project can be viewed in the browser at the URL: `http://localhost:8000`.

Hot Reload

The development server will reload whenever there are changes to the application code.

Default Apps

New Django projects have several apps installed by default such as the admin panel and authentication.

```
INSTALLED_APPS = [  
# ... Other pre-installed apps  
    "myapp.apps.MyappConfig"  
]
```

Creating a Django App

New apps are made with the `startapp` command.

```
python3 manage.py startapp  
<name_of_app>
```

Installed Apps

New Django apps must be added to **settings.py** in the `INSTALLED_APPS` list in order for the project to recognize them.

```
INSTALLED_APPS = [  
    # ... Other pre-installed apps  
    "myapp.apps.MyappConfig"  
]
```

URLconfig Files

URLs are configured in URLconfig file, **urls.py**.

```
urlpatterns = [  
    ("", include("app.urls"))  
]
```

Mapping View Functions

View functions are mapped to URLs inside **urls.py**.

```
urlpatterns = [  
    path('', views.home, name="home")  
]
```

Migrations

The `migrate` command will update database models for projects installed in `INSTALLED_APPS`.

```
python3 manage.py migrate
```

urlpatterns

URLs are created using the `path()` function which takes three arguments— the route, the view function, and an optional name.

```
path("", views.home, name="home")
```

Template Namespace

Templates should be namespaced by creating a nested folder structure to avoid name conflicts between apps.

```
└─ myapp/  
  └─ templates/  
    └─ myapp/  
      └─ mytemplate.html
```

```
mysite/  
├─ manage.py  
└─ mysite  
  ├─ __init__.py  
  ├─ settings.py  
  └─ urls.py
```

HTTP Responses

A response to a web request is given in a `HttpResponse` object, which can be anything from HTML to an image.

```
def myView(request):  
    return HttpResponse("Here is my  
response")
```