

Django Web Framework Cheatsheet

Author: Placements Simplified Editorial Team

Introduction: Batteries Included Web Framework

Django is a high-level, Python-based web framework that encourages rapid development, clean design, and pragmatic code. It follows the Model-View-Template (MVT) architecture pattern and comes with built-in Object-Relational Mapping (ORM), user authentication, admin panels, and security layers.

1. Installation & Setup

Install Django framework package:

```
pip install django
```

Check framework version:

```
django-admin --version
```

Create a new project directory:

```
django-admin startproject projectName  
cd projectName
```

Start local development server (defaults to port 8000):

```
python manage.py runserver  
# Specify custom port:  
python manage.py runserver 8080
```

2. Django Model (Database Table Blueprint)

Models define the structural database table schemas. Example Product model definition:

```
from django.db import models

class Product(models.Model):
    product_id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=100)
    price = models.FloatField()
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name
```

3. View and Template Rendering

Standard HTTP Response view (views.py):

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Welcome student!")
```

Render HTML template with context arguments:

```
from django.shortcuts import render

def index(request):
    return render(request, "index.html", {"title": "Welcome"})
```

4. URL Routing

Example route mappings (urls.py):

```
from django.contrib import admin
from django.urls import path, include
from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name='index'),
    path('about/', views.about, name='about'),
    # Modular inclusion:
    path('blog/', include('blog.urls')),
]
```

5. Database Migrations

Generates migration instructions based on changes made in models.py:

```
python manage.py makemigrations
```

Apply modifications to the physical database:

```
python manage.py migrate
```

Review actual SQL queries generated for dry runs:

```
python manage.py sqlmigrate appName 0001
```

6. Django Admin & Shell Commands

Create superuser credential for administrative panel Access (/admin):

```
python manage.py createsuperuser
```

Launch interactive python terminal linked to models:

```
python manage.py shell
```