



SETUP & PROJECT

```
pip install django
django-admin startproject mysite
python manage.py startapp blog
python manage.py runserver
python manage.py migrate
INSTALLED_APPS must include new apps
```

MODELS

```
from django.db import models
class Post(models.Model):
    title = models.CharField(max_length=200)
    body = models.TextField()
    created =
models.DateTimeField(auto_now_add=True)
author = models.ForeignKey(User,
on_delete=models.CASCADE)
class Meta: ordering = ['-created']
```

VIEWS

```
from django.views import View
class PostListView(View):
    def get(self, request):
        posts = Post.objects.all()
        return
render(request, 'list.html', {'posts': posts})
# Function-based:
def post_list(request):
    return JsonResponse({'posts':
list(...)}, safe=False)
```

URLS & ROUTING

```
# urls.py
urlpatterns = [
    path('posts/',
views.PostListView.as_view()),
    path('posts/<int:pk>/',
views.PostDetailView.as_view()),
    path('api/', include('api.urls')),
]
re_path(r'^old/(?P<slug>[-w]+)/$', view)
```

ORM QUERYSETS

```
Post.objects.all()
Post.objects.filter(author=user,
published=True)
Post.objects.exclude(status='draft')
Post.objects.get(pk=1) # raises
DoesNotExist
Post.objects.order_by('-created')[:10]
Post.objects.select_related('author')
Post.objects.prefetch_related('tags')
```

MIGRATIONS

```
python manage.py makemigrations
python manage.py migrate
python manage.py showmigrations
python manage.py migrate blog 0002
python manage.py sqlmigrate blog 0001
Never delete migrations in
production
```

TEMPLATES

```
{{ variable }} {# comment #}
{% if user.is_authenticated %}...{%
endif %}
{% for post in posts %}...{% endfor %}
{% extends 'base.html' %}
{% block content %}...{% endblock %}
{% url 'post-detail' pk=post.pk %}
```

ADMIN

```
from django.contrib import admin
@admin.register(Post)
class PostAdmin(admin.ModelAdmin):
    list_display =
['title', 'author', 'created']
search_fields = ['title', 'body']
list_filter = ['status', 'created']
python manage.py createsuperuser
```

AUTHENTICATION

```
from django.contrib.auth.decorators
import login_required
@login_required(login_url='/login/')
def dashboard(request): ...
LoginView / LogoutView /
PasswordResetView
request.user.is_authenticated
authenticate(username=u, password=p)
login(request, user) / logout(request)
```

MIDDLEWARE

```
MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'myapp.middleware.CustomMiddleware',
]
Custom: define
__init__(get_response) and
__call__(request)
```

FORMS & VALIDATION

```
class PostForm(forms.ModelForm):
class Meta:
    model = Post
    fields = ['title', 'body']
form = PostForm(request.POST or None)
if form.is_valid(): form.save()
def clean_title(self): # custom
validation
```

COMMON GOTCHAS

N+1: always use
select_related/prefetch_related
CSRF token required in all POST
forms
DEBUG=True exposes full tracebacks
disable in prod
SECRET_KEY must be secret use env
vars
Migrations in prod: always backup
before migrate