# DJANGO VIEWS – COMPLETE HANDWRITTEN STYLE NOTES

## 1. What is a View?

A Django View is a Python function or class that:
- Receives an HTTP Request
- Processes logic (database, forms, permissions)
- Returns an HTTP Response (HTML, JSON, Redirect, etc.)

Flow:
URL → View → Logic → Response → Browser

## 2. Function-Based Views (FBV)

Used when logic is simple and you want full control.

Syntax:

```
def home(request):
return HttpResponse("Hello")
```

Use FBV when:
- View is small/simple
- You want clarity
- Minimal abstraction needed

## 3. Class-Based Views (CBV)

Used when you need structure and reuse.

Syntax:

```
class MyView(View):
def get(self, request):
return HttpResponse("GET")

def post(self, request):
return HttpResponse("POST")
```

URL Mapping:

```
path("my/", MyView.as_view())
```

Use CBV when:
- Same route handles GET & POST
- Reusability required
- Cleaner structure needed

## 4. Generic Class-Based Views (CRUD)

TemplateView – Static pages
ListView – Display list of objects
DetailView – Display single object
CreateView – Create object (form + save)
UpdateView – Update object
DeleteView – Delete object

Example:

```
class ProductListView(ListView):
model = Product
template_name = "products/list.html"
paginate_by = 10
```

Use Generic Views when:
- Building CRUD functionality
- Want minimal repeated code

## 5. API Views (JSON)

Simple JSON:

```
def status(request):
return JsonResponse({"ok": True})
```

Django REST Framework:
APIView – Full control
GenericAPIView – Standard CRUD
ModelViewSet – Complete REST automatically

Use DRF when:
- Building frontend (React/Next/Mobile)
- Need authentication & permissions
- Want RESTful APIs

## FINAL MEMORY TRICK

Django Views = Request In → Response Out

FBV → Simple logic
CBV → Structured logic
Generic CBV → CRUD shortcut
DRF → APIs

Choose based on complexity and project size.