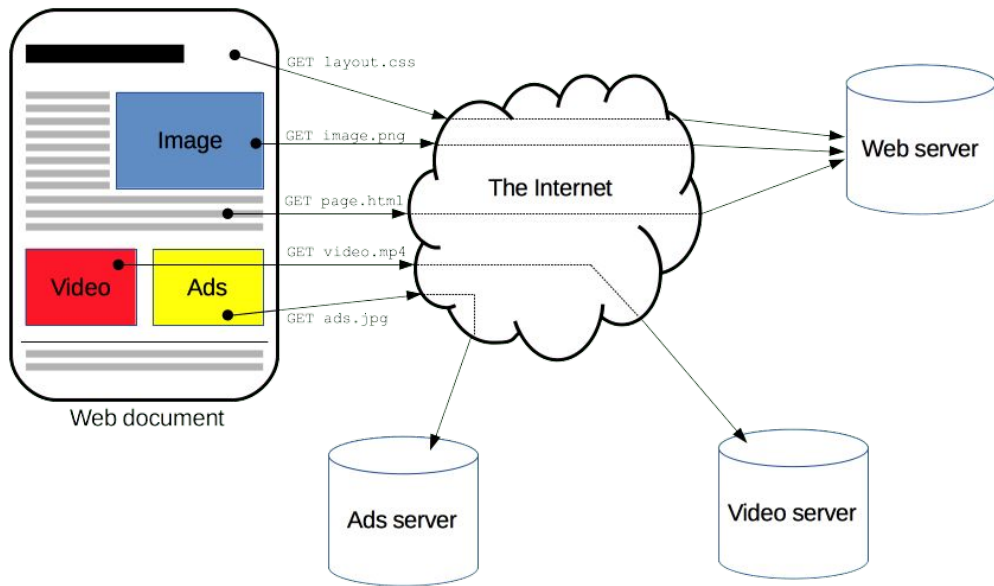
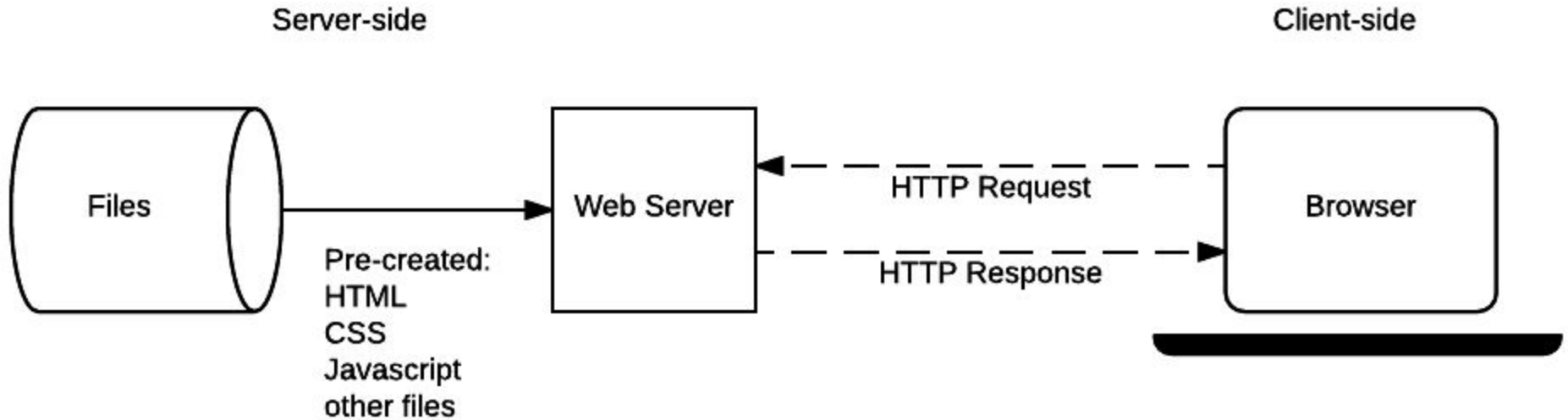

Server-side website programming (Django)

What is a website?



What is a website?



Request

```
GET /en-US/search?
q=client+server+overview&topic=apps&topic=html&topic=css&topic=js&topic=a
pi&topic=webdev HTTP/1.1
Host: developer.mozilla.org
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8
Referer: https://developer.mozilla.org/en-US/
Accept-Encoding: gzip, deflate, sdch, br
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7
Accept-Language: en-US,en;q=0.8,es;q=0.6
Cookie: sessionId=6ynxs23n521lu21b1t136rhbv7ezngie;
csrftoken=zIPUJsAZv6pcgCBJSCj1zU6pQZbfMUAT; dwf_section_edit=False;
dwf_sg_task_completion=False; _gat=1; _ga=GA1.2.1688886003.1471911953;
ffo=true
```

Response

HTTP/1.1 200 OK

Server: Apache

X-Backend-Server: developer1.webapp.scl3.mozilla.com

Vary: Accept, Cookie, Accept-Encoding

Content-Type: text/html; charset=utf-8

Date: Wed, 07 Sep 2016 00:11:31 GMT

Keep-Alive: timeout=5, max=999

Connection: Keep-Alive

X-Frame-Options: DENY

Allow: GET

X-Cache-Info: caching

Content-Length: 41823

<!DOCTYPE html>

<html lang="en-US" dir="ltr" class="redesign no-js" data-ffo-opensanslight=false data-ffo-opensans=false >

<head prefix="og: http://ogp.me/ns#">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=Edge">

<script>(function(d) { d.className = d.className.replace(/\\bno-js/,
''); })(document.documentElement);</script>

...

What does a developer do?

→ Read the request

When a client make a request , a developer reads the request and method.

→ Process data

After getting request, developers process data with the help of database by programming language.

→ Return Response

After processing data, developers send the data as response in a template or JSON format.

Django

→ Complete

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box" (i.e. authentication, permission, template rendering, email sending)

→ Versatile

→ Secure

→ Scalable

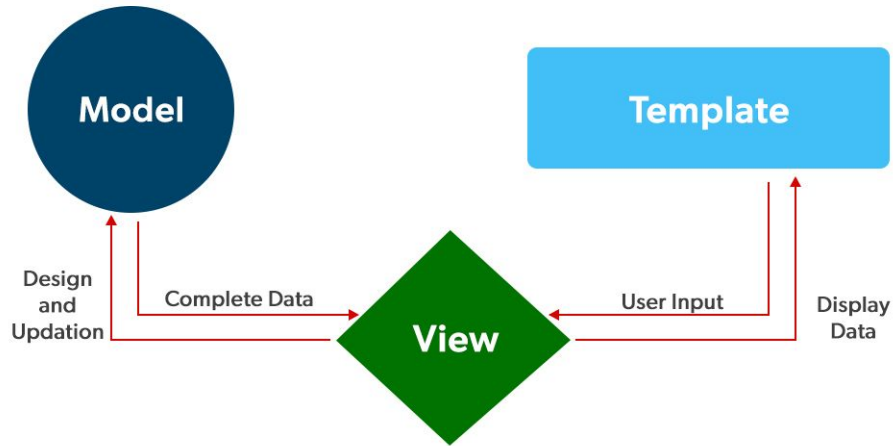
→ Maintainable

→ Portable

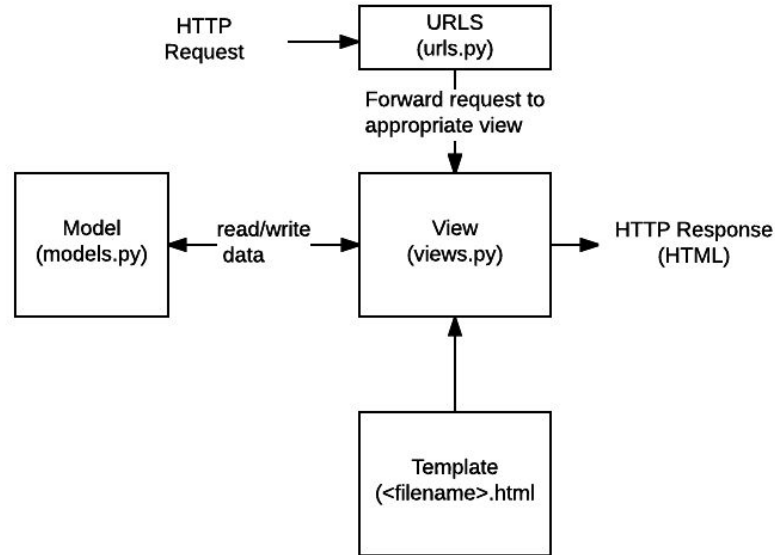
Who are using Django?



Django Architecture



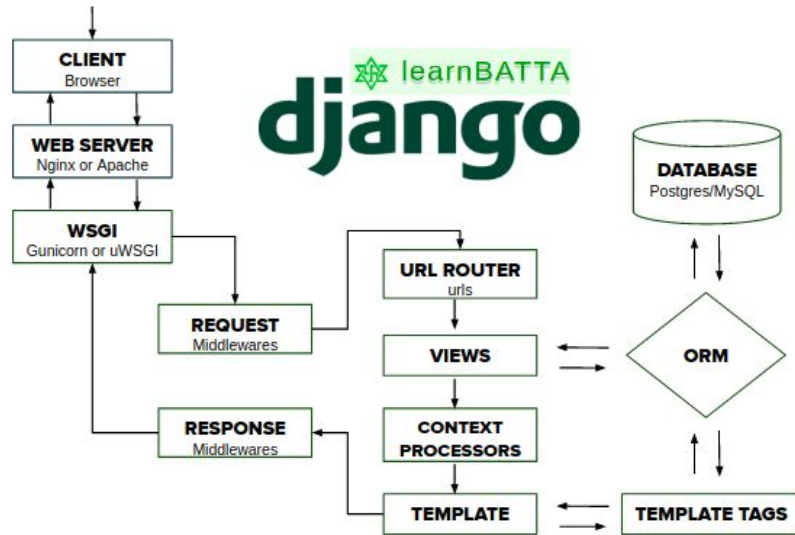
Django Architecture



Prerequisites

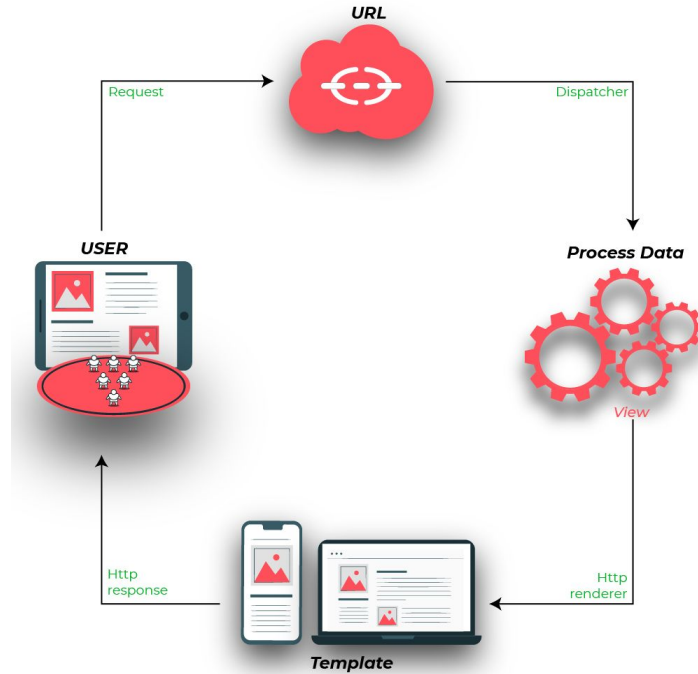
- Python [data structure, virtual environment]
- Client server interaction
- HTML
- CSS
- VScode

Django Lifecycle



request-response lifecycle in Django

Django Lifecycle



Django Models

polls/models.py



```
from django.db import models
```

```
class Question(models.Model):
```

```
    question_text = models.CharField(max_length=200)
```

```
    pub_date = models.DateTimeField('date published')
```

```
class Choice(models.Model):
```

```
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
```

```
    choice_text = models.CharField(max_length=200)
```

```
    votes = models.IntegerField(default=0)
```

Django Views

polls/views.py



```
from django.http import HttpResponse

from .models import Question

def index(request):
    latest_question_list = Question.objects.order_by('-pub_date')[:5]
    output = ', '.join([q.question_text for q in latest_question_list])
    return HttpResponse(output)

# Leave the rest of the views (detail, results, vote) unchanged
```

Django urls

polls/urls.py



```
from django.urls import path

from . import views

urlpatterns = [
    # ex: /polls/
    path('', views.index, name='index'),
    # ex: /polls/5/
    path('<int:question_id>/', views.detail, name='detail'),
    # ex: /polls/5/results/
    path('<int:question_id>/results/', views.results, name='results'),
    # ex: /polls/5/vote/
    path('<int:question_id>/vote/', views.vote, name='vote'),
]
```