Essential of Modern Web Application Programming

Nuri Çilengir

$ whoami

- Nuri Çilengir
    - Security & Software Development and Free Software Enthusiast
    - Freelance Developer @ Anywhere
    - Translator & Writer @ Arka Kapı Cyber Security Magazine

- Social Media
    - Twitter : der3ctus
    - Github : derectus

What is the Web ?

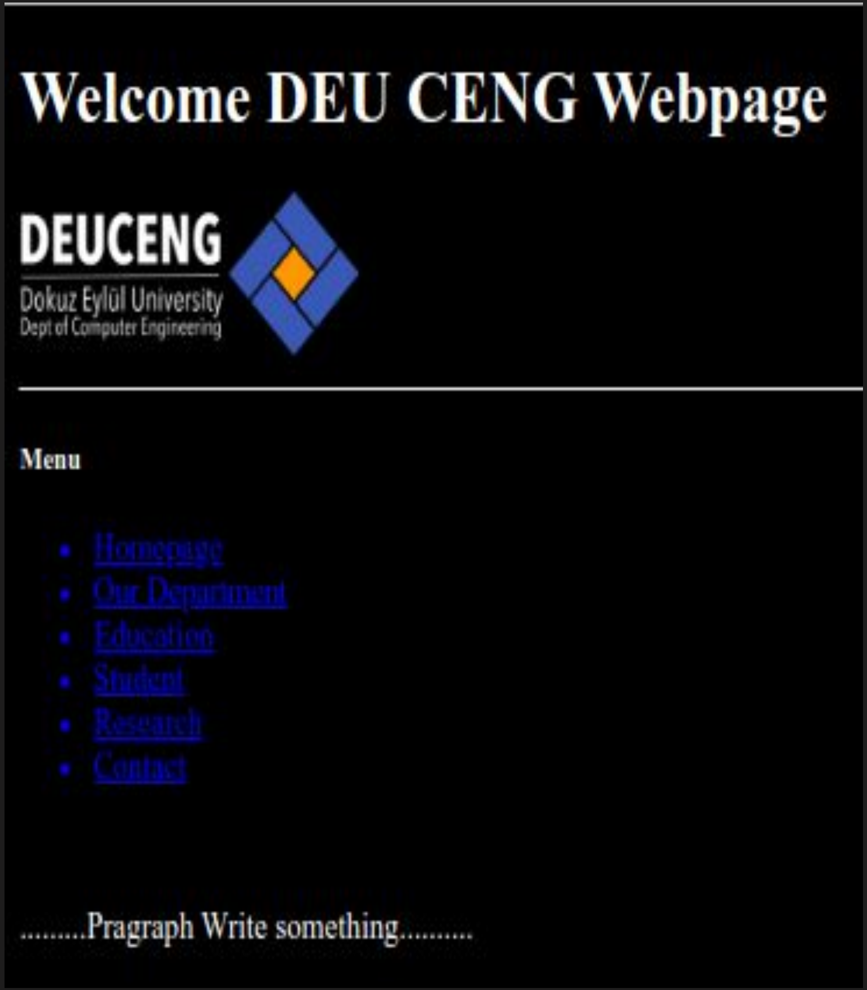
- Web is NOT internet.
    - The Web is just one of the ways that information is shared over the Internet; others include email, instant messaging and File Transfer Protocol (FTP).

Tim Berners Lee, Creator of WWW

- A brief history of WEB!

HTML - HyperText Markup Language

- HTML (**H**yper**T**ext **M**arkup **L**anguage) is the code that is used to structure a web page and its content.
  - For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.
  - So, HTML is just MARKUP LANGUAGE.
- Okeyy but why did we need HTML?

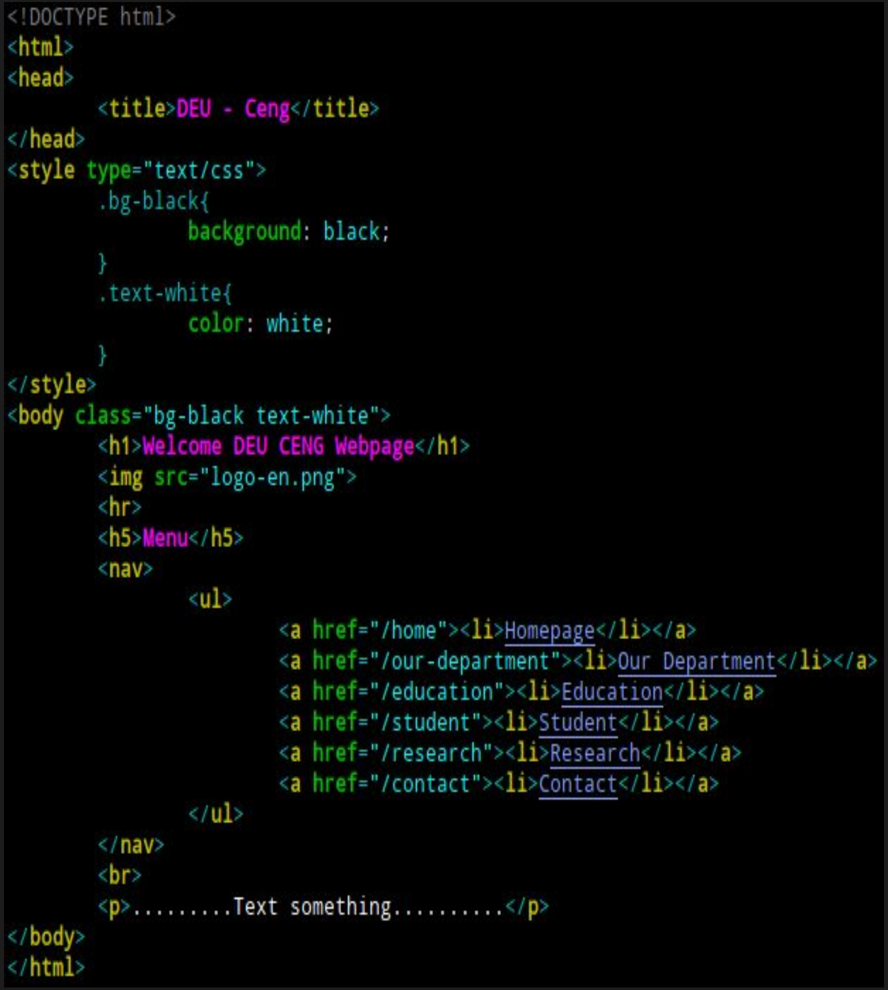


Welcome DEU CENG Webpage

Menu

- Homepage
- Our Department
- Education
- Student
- Research
- Contact

.........Pragraph Write something..........



```
<!DOCTYPE html>
<html>
<head>
        <title>DEU - Ceng</title>
</head>
<style type="text/css">
        .bg-black{
                background: black;
        }
        .text-white{
                color: white;
        }
</style>
<body class="bg-black text-white">
        <h1>Welcome DEU CENG Webpage</h1>
        <img src="logo-en.png">
        <hr>
        <h5>Menu</h5>
        <nav>
                <ul>
                        <a href="/home"><li>Homepage</li></a>
                        <a href="/our-department"><li>Our Department</li></a>
                        <a href="/education"><li>Education</li></a>
                        <a href="/student"><li>Student</li></a>
                        <a href="/research"><li>Research</li></a>
                        <a href="/contact"><li>Contact</li></a>
                </ul>
        </nav>
        <br>
        <p>.........Text something..........</p>
</body>
</html>
```

● HTTP - HyperText Transfer Protocol

- The main mission of HTTP is to communicate our requests in a way that the server understands and to return the response it receives to us.

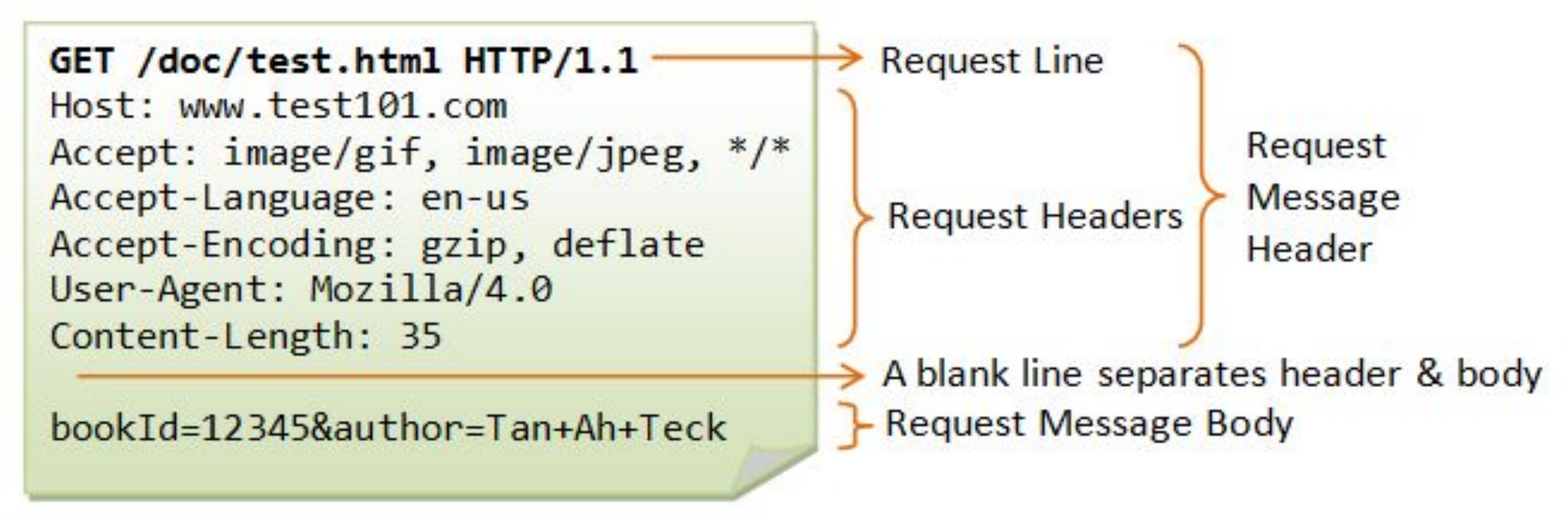
- So, if you want to share any information to world, you need a Transfer Protocol is called HTTP!

- It have some versions;
  - HTTP/0.9
  - HTTP/1.0
  - HTTP/1.1
  - HTTP/2.0

- Check out the RFC 2616 and RFC 7540 for more!

What is the content of HTTP request?

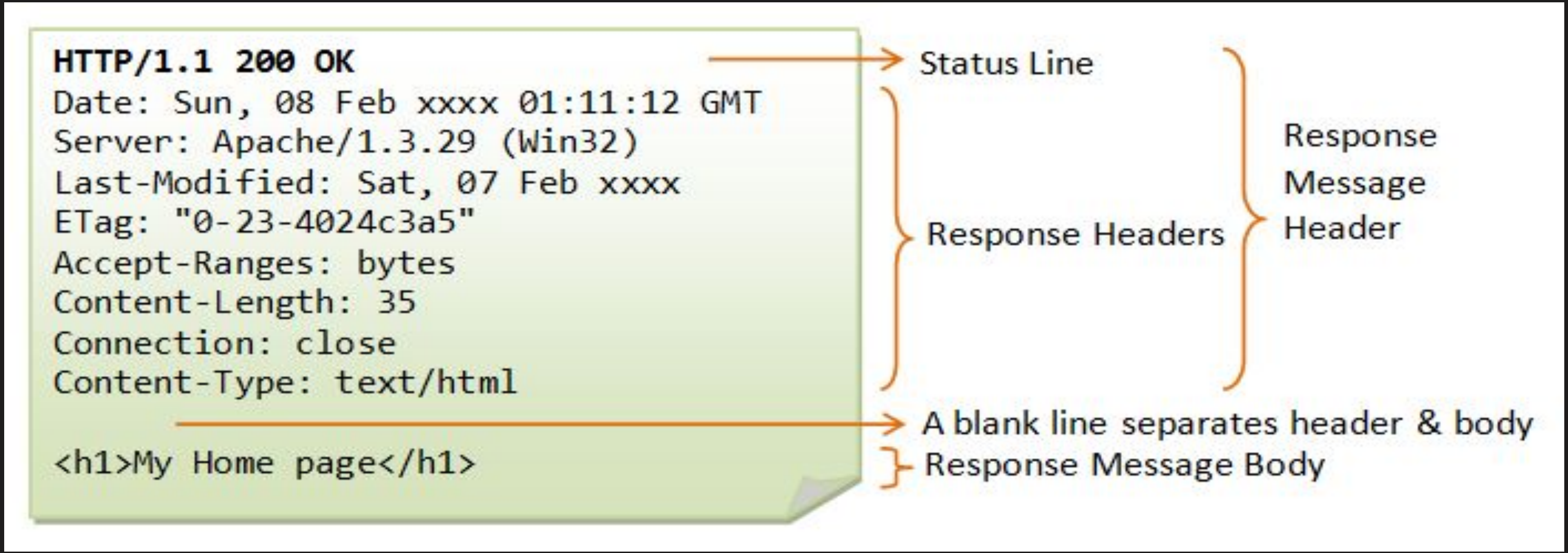
- Example : HTTP request message



```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

And.. Response...

- Example : HTTP response



```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

HTTP Methods

- **GET**
  - The GET method requests a representation of the specified resource. Requests using GET should only receive data.
- **POST**
  - The POST method is used to send an entity to the specified resource, which usually causes a state change or side effect on the server.
- PUT
  - The PUT method replaces all current representations of the target resource with the uploaded content.
- DELETE
  - The DELETE method deletes the specified resource.
- CONNECT
  - The CONNECT method creates a tunnel to the server defined by the source on the target.
- OPTIONS
  - The OPTIONS method is used to define the communication options of the source on the target.
- PATCH
  - The PATCH method is used to apply partial changes to a resource.

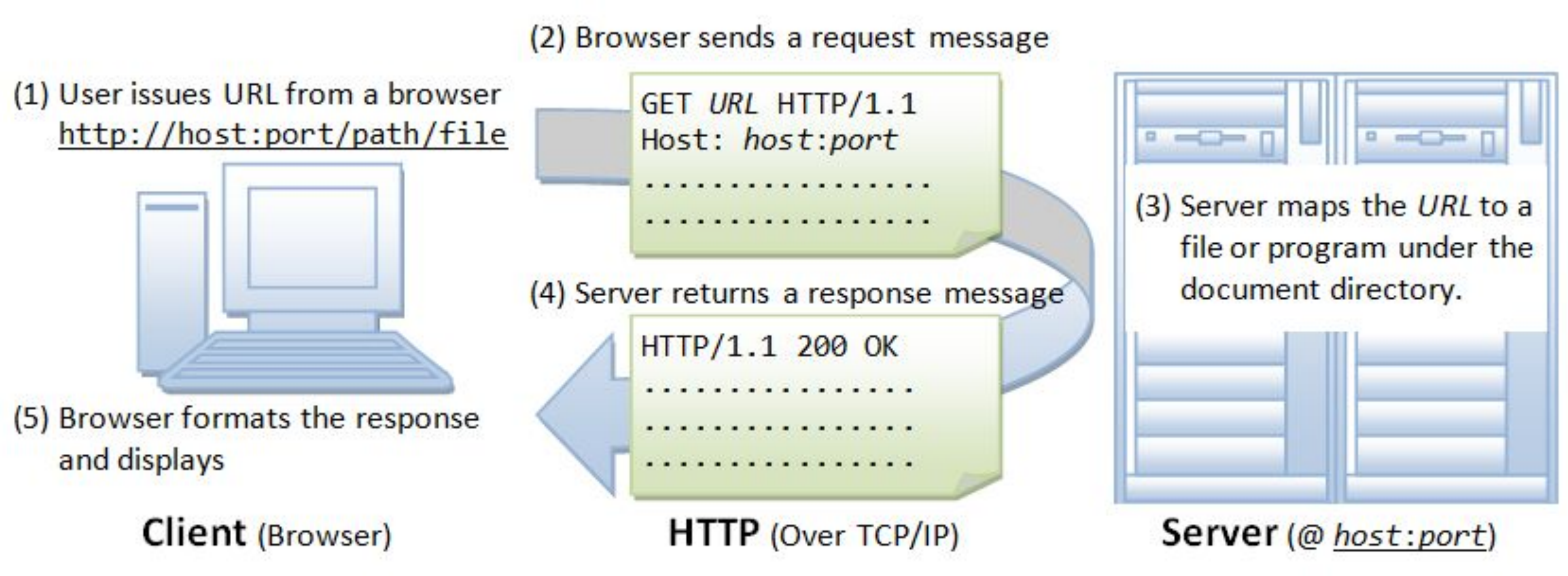
HTTP Status Codes

- 1xx: Informational
- 2xx: Success
  - ex: 200 OK
- 3xx: Redirection
  - ex: 301 Moved Permanently
- 4xx: Client Error
  - ex: 404 Not Found <3
- 5xx: Server Error
  - ex: 500 Internal Server Error

- # Browsers!!

  - Now, we created HTML to represent the information. So, the information we wanted to share became, in part, orderly. We used HTTP to share this but something is missing…
    - Browsers are interpreter!!

- What happens when we want to send any request?
  - Client - Server Model



(1) User issues URL from a browser http://host:port/path/file

(2) Browser sends a request message

```
GET URL HTTP/1.1
Host: host:port
..................
..................
```

(3) Server maps the *URL* to a file or program under the document directory.

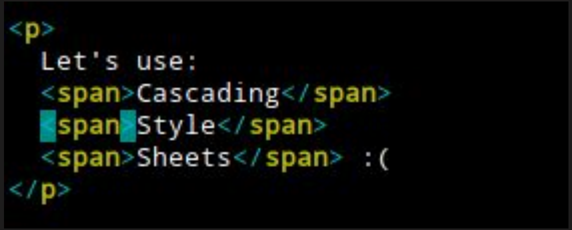
(4) Server returns a response message

```
HTTP/1.1 200 OK
................
................
................
```

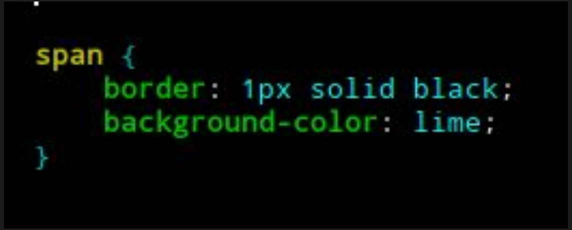
(5) Browser formats the response and displays

CSS - Cascading Style Sheets

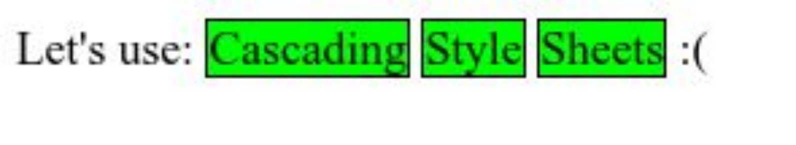
- CSS is a language for specifying how documents are presented to users.



```
<p>
  Let's use:
  <span>Cascading</span>
  <span>Style</span>
  <span>Sheets</span>  :(
</p>
```



```
span {
    border: 1px solid black;
    background-color: lime;
}
```



Let's use: Cascading Style Sheets :(

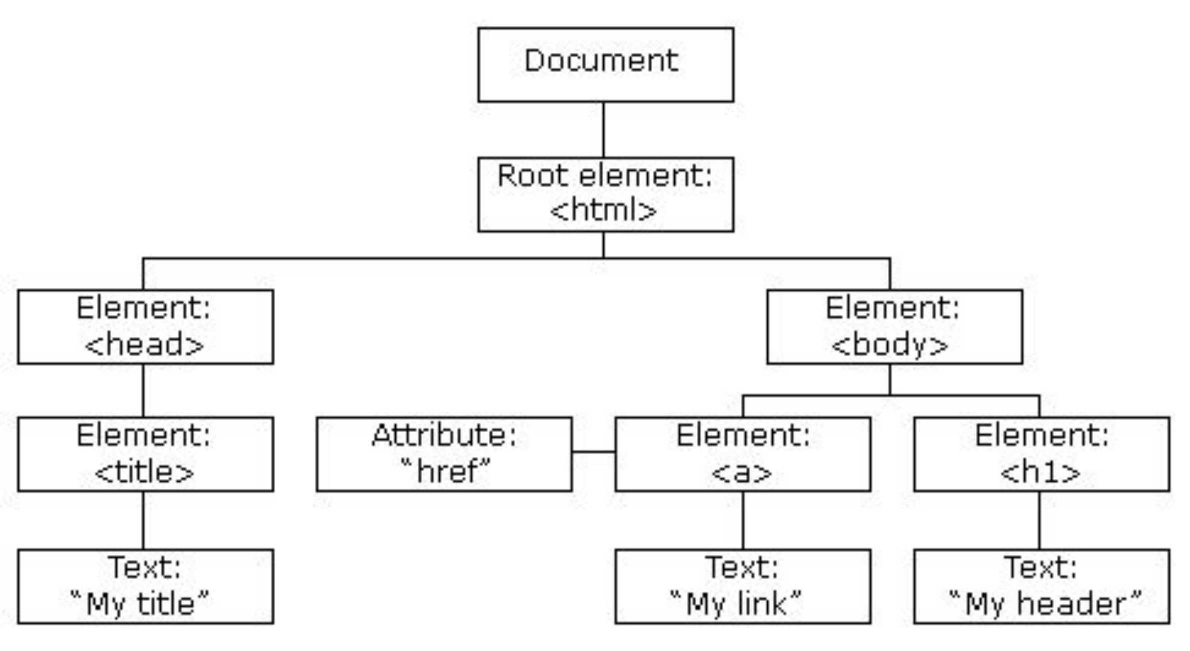
JS - JavaScript

- JavaScript ("JS" for short) is a full-fledged dynamic programming language that, when applied to an HTML document, can provide dynamic interactivity on websites.

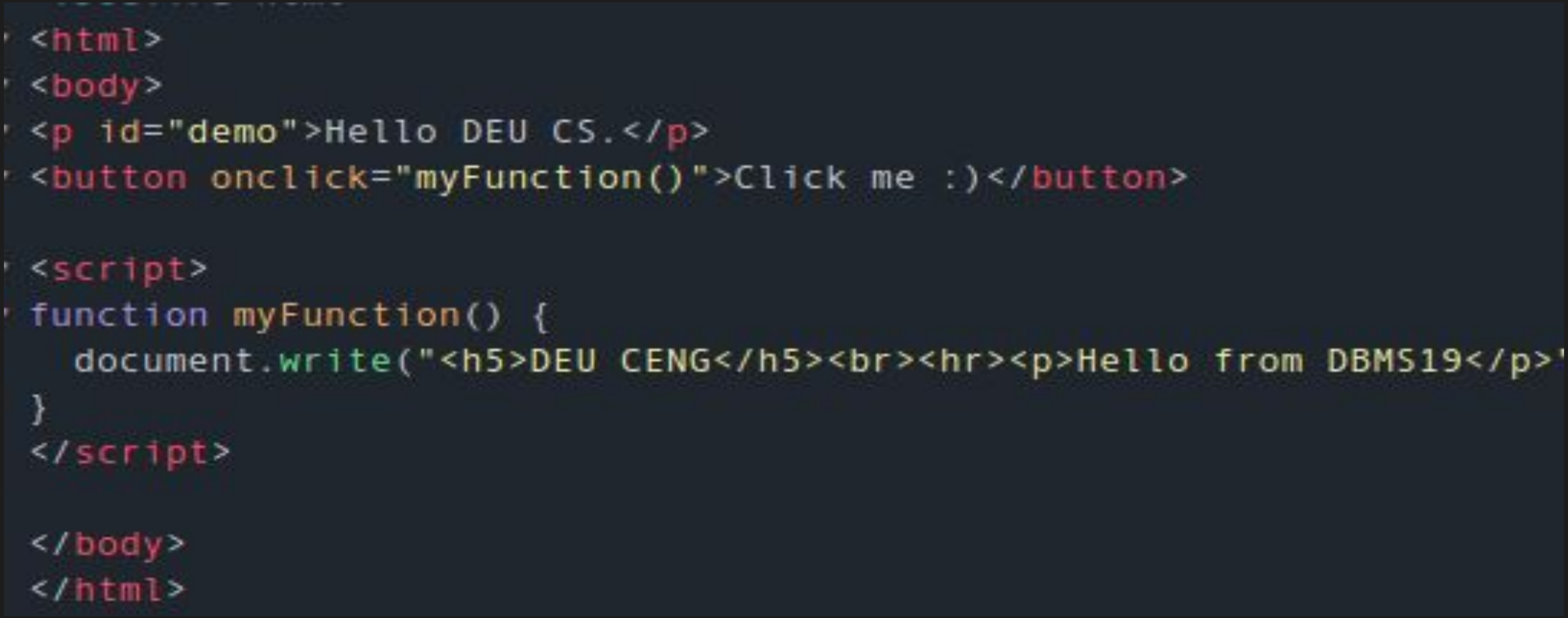
- object-oriented, imperative, functional

- JS using both client-side and server-side
    - How works on client-side ??
    - How works on server-side ???

- DOM - Data Object Model

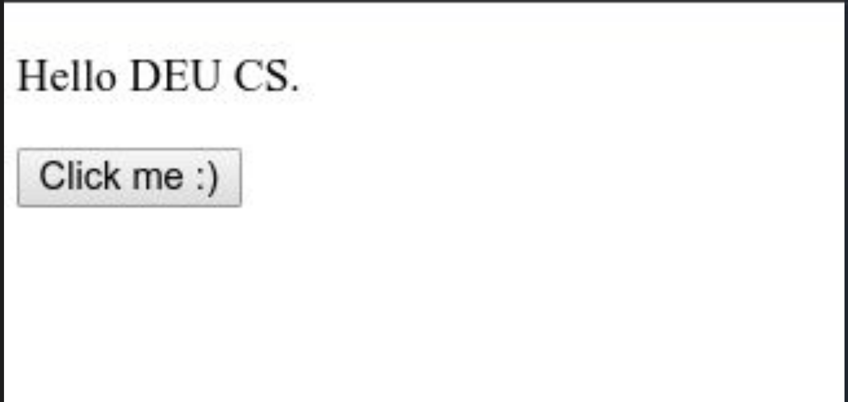


- ## JS Exp: DOM

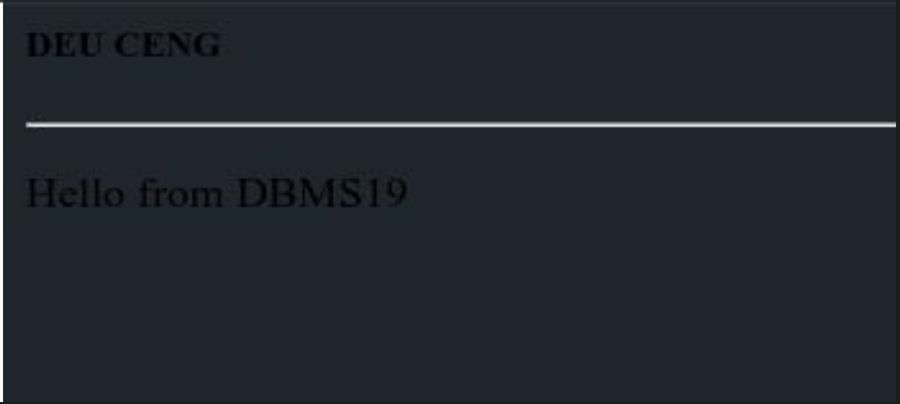


```
<html>
<body>
<p id="demo">Hello DEU CS.</p>
<button onclick="myFunction()">Click me :)</button>

<script>
function myFunction() {
  document.write("<h5>DEU CENG</h5><br><hr><p>Hello from DBMS19</p>"
}
</script>

</body>
</html>
```

- JS Exp: DOM



Hello DEU CS.

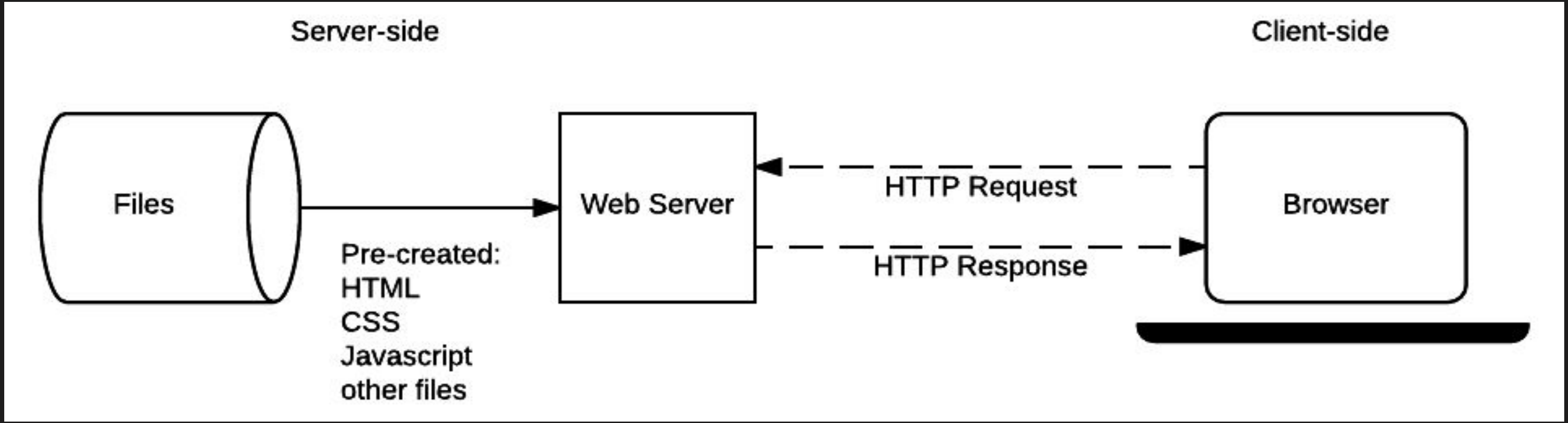


DEU CENG

Hello from DBMS19

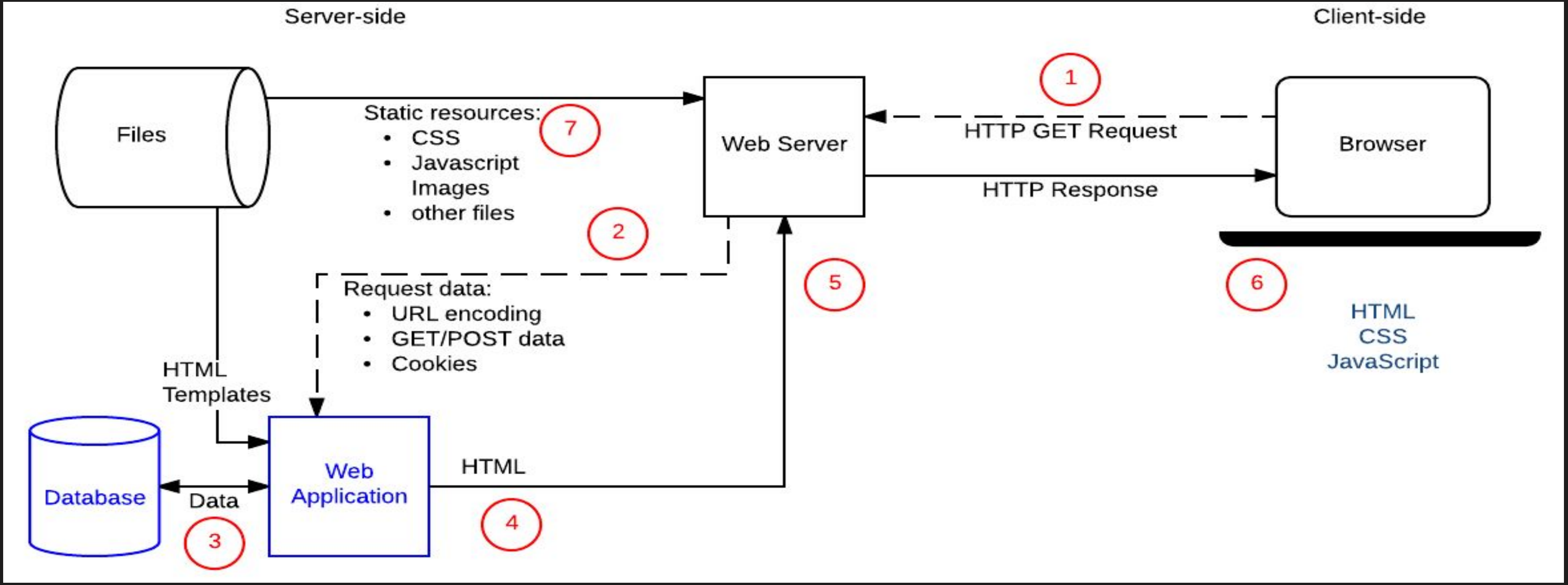
Static Sites

- It is the program that runs on the client machine (browser) and deals with the user interface/display. It can be interact with temporary storage, sending request for data to server, interact with local storage, work as an interface between server and user.



Dynamic sites

- A dynamic website is one where some of the response content is generated dynamically only when needed. Dynamic websites that access information from a database are also called database-driven websites.

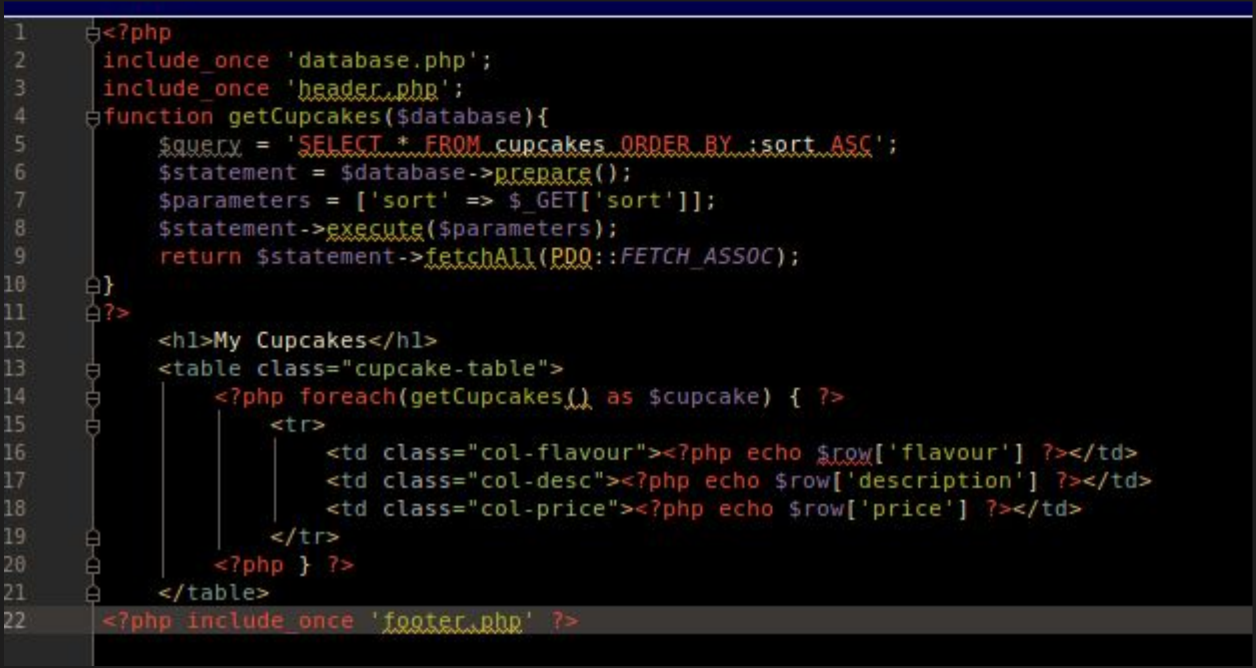


Server Side Programming with PHP

- In short, PHP is a programming language made for the web, built up from the C programming language, and which uses idiosyncratic HTML-like tags (or sigils) to contain its code.

- **What can PHP do?**
  - **What is PHP used for? Basically, anything that you want to do on a web server, you can do with PHP.**
    - **Server-side scripting**
    - **Command line scripting**
    - **Writing desktop application (YES YOU CAN :))**
- So with PHP, you have the freedom of choosing an operating system and a web server.

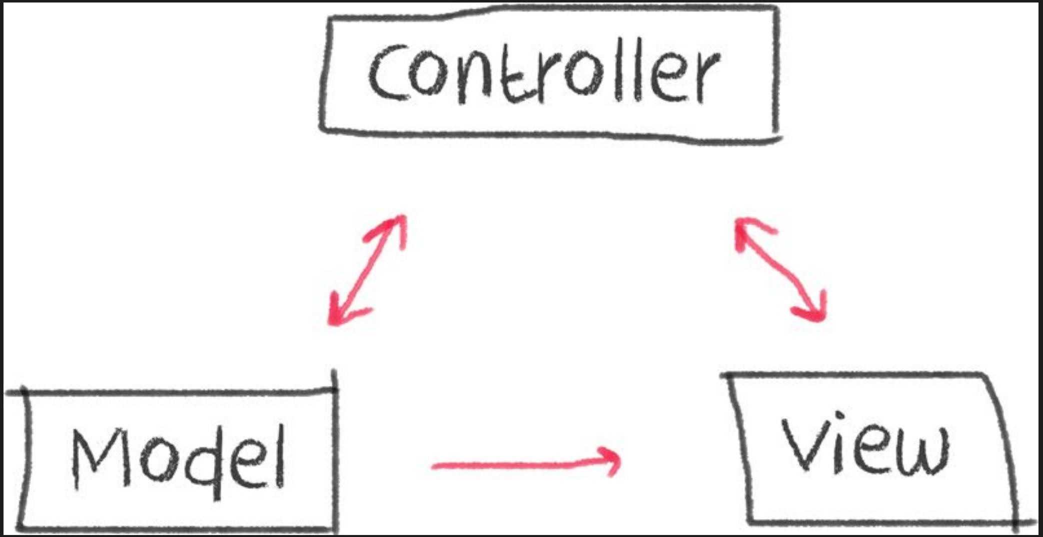
- Example PHP coding



```php
<?php
include_once 'database.php';
include_once 'header.php';
function getCupcakes($database){
    $query = 'SELECT * FROM cupcakes ORDER BY :sort ASC';
    $statement = $database->prepare();
    $parameters = ['sort' => $_GET['sort']];
    $statement->execute($parameters);
    return $statement->fetchAll(PDO::FETCH_ASSOC);
}
?>
    <h1>My Cupcakes</h1>
    <table class="cupcake-table">
        <?php foreach(getCupcakes() as $cupcake) { ?>
            <tr>
                <td class="col-flavour"><?php echo $row['flavour'] ?></td>
                <td class="col-desc"><?php echo $row['description'] ?></td>
                <td class="col-price"><?php echo $row['price'] ?></td>
            </tr>
        <?php } ?>
    </table>
<?php include_once 'footer.php' ?>
```

MVC Architecture

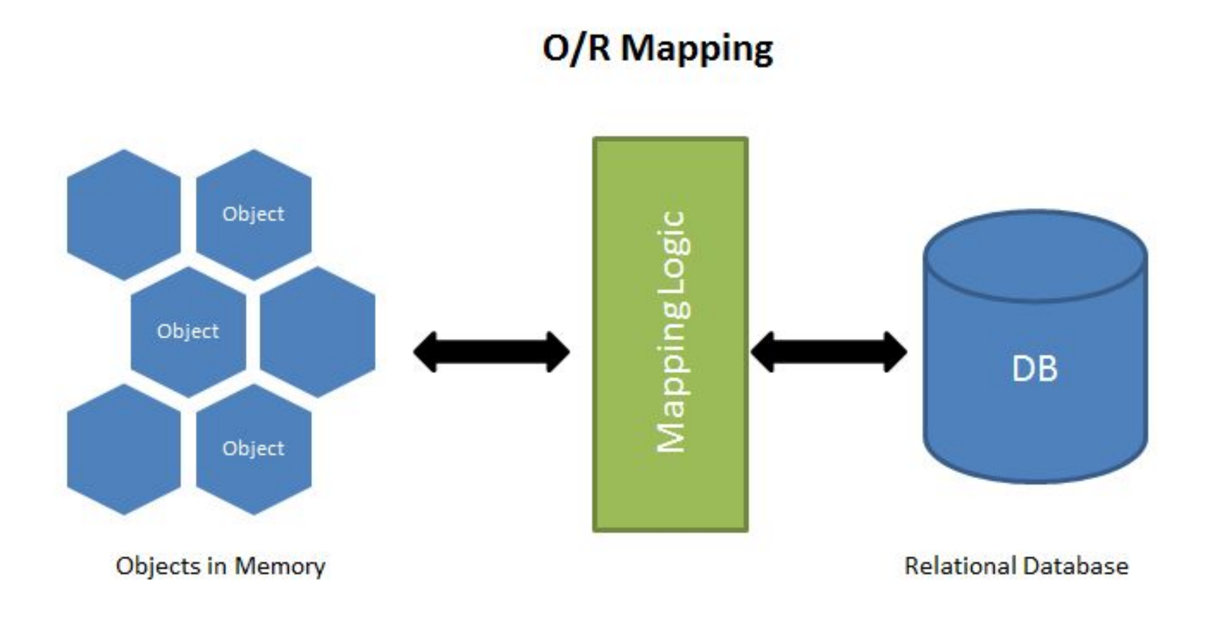
- Along with web applications, software architectures such as MVC entered our lives. One of the common features of this type of architecture is the ability to write backend and frontend codes in separate layers for ease of code readability, web service and other web applications.



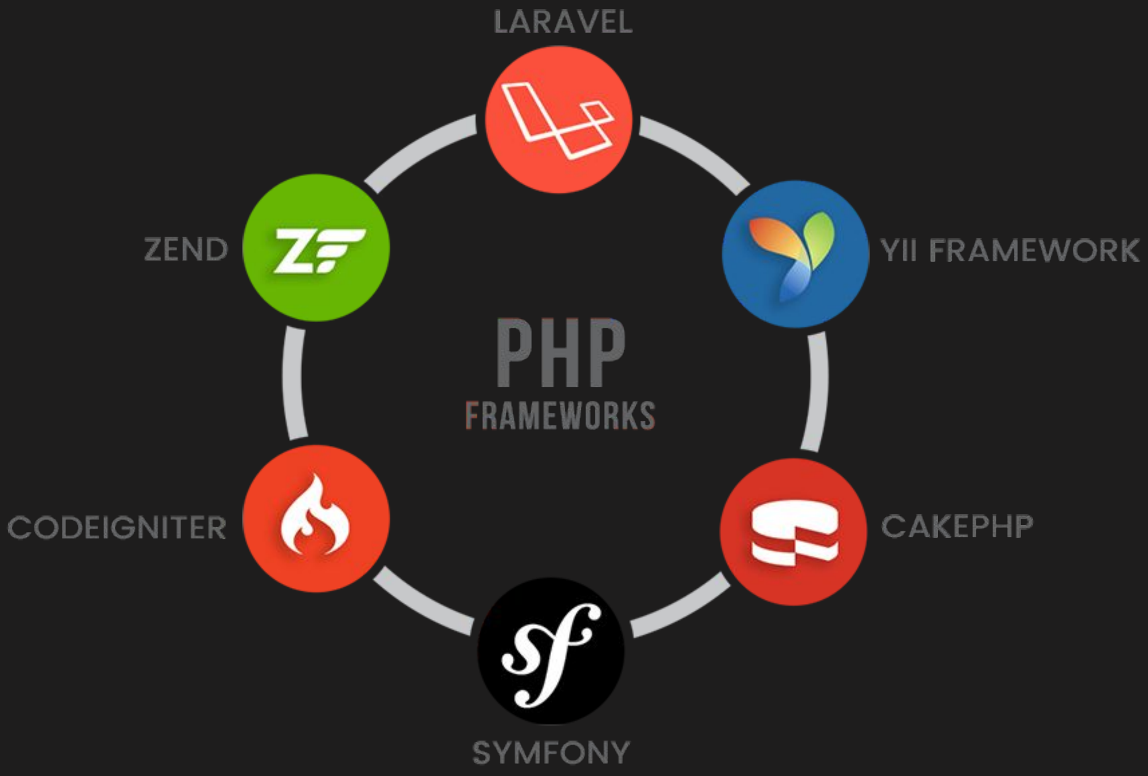
How works MVC?

- The user tells the Controller layer what he wants by means of defined routes.
- Controller layer; if there is any data transaction between the user's requests, he takes the necessary actions to the Model layer and gets the answer.
- Than, It transmits the response to the View layer.
- The View layer prepares the answer that the user needs to see and transmits it to the user.

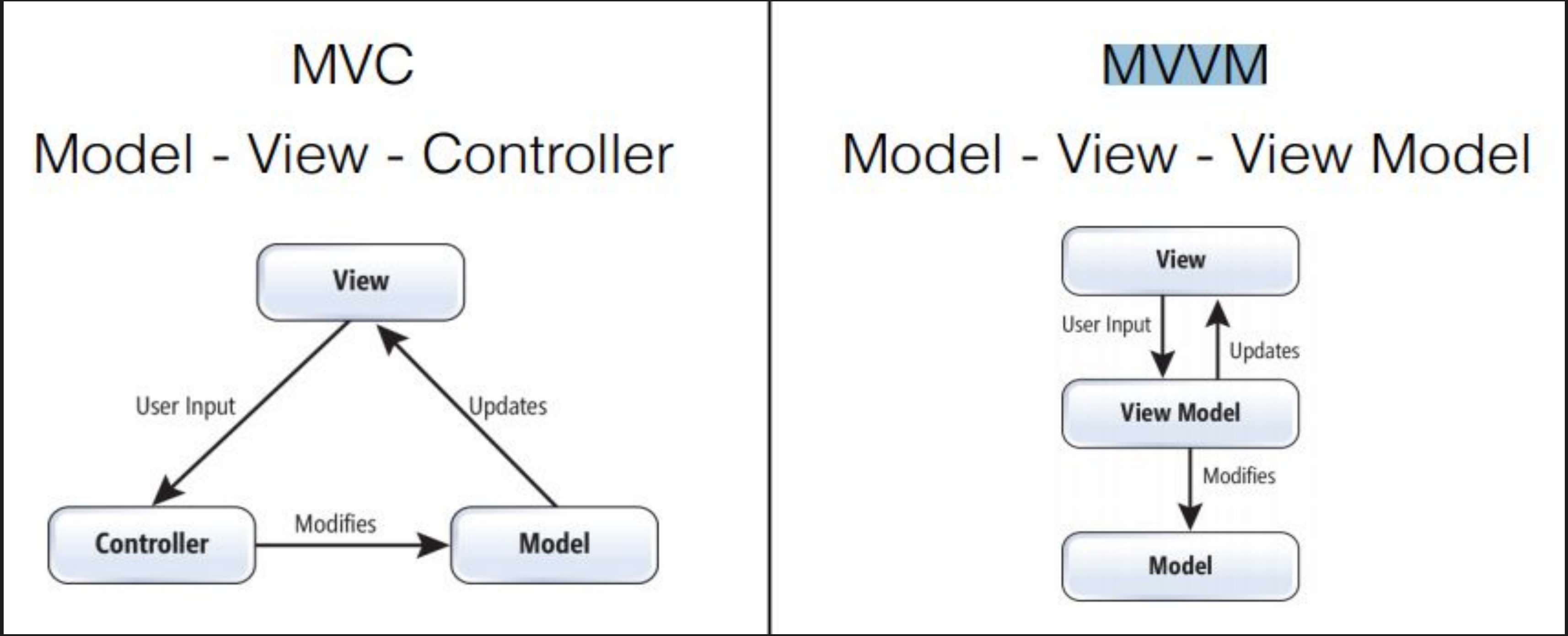
- ORM - Object Relational Mapping



- PHP Frameworks



- MVVM - Model - View - View Model



- Frameworks

Angular
React
Vue.js
ember
Ember.js

DEMO

- Our demo application will be done on Laravel Framework and use Vue.js as the frontend component.

- Tasks
  - MVC Project Structure
  - Database Connection
  - Route Structure
  - Model Structure
  - Controller Structure
  - View Components
  - Also, CRUD Operations