

# Using Handlebars

Dr. Charles Severance

[www.wa4e.com](http://www.wa4e.com)

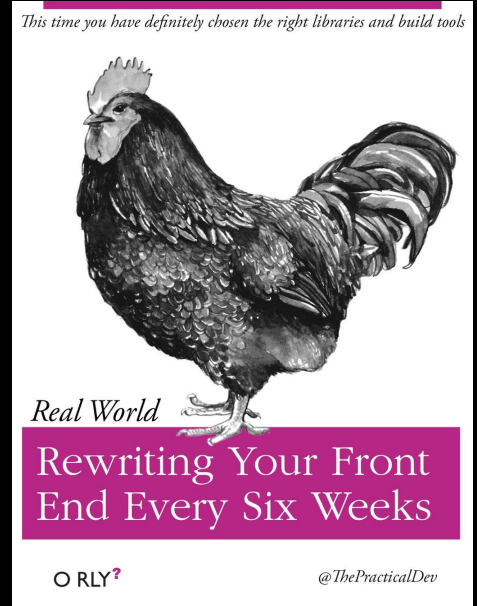
<http://www.wa4e.com/code/handlebars.zip>



# Rendering in the Browser

Applications are starting to use JSON “back end” services and construct the View HTML in JavaScript

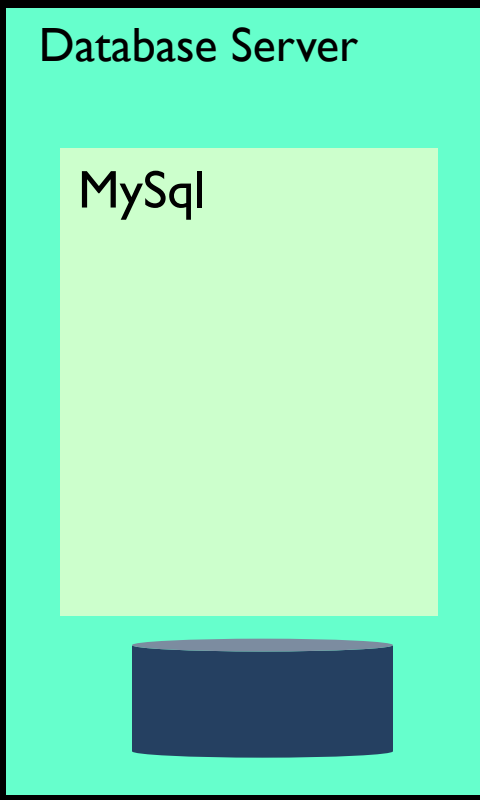
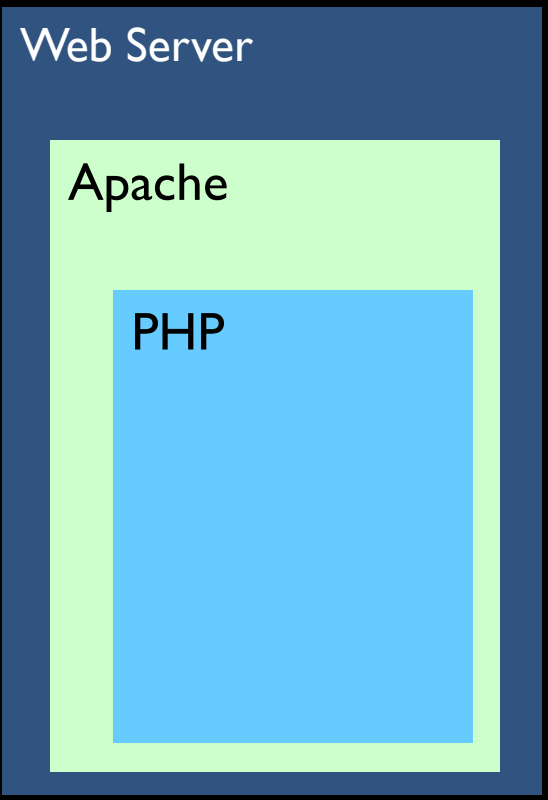
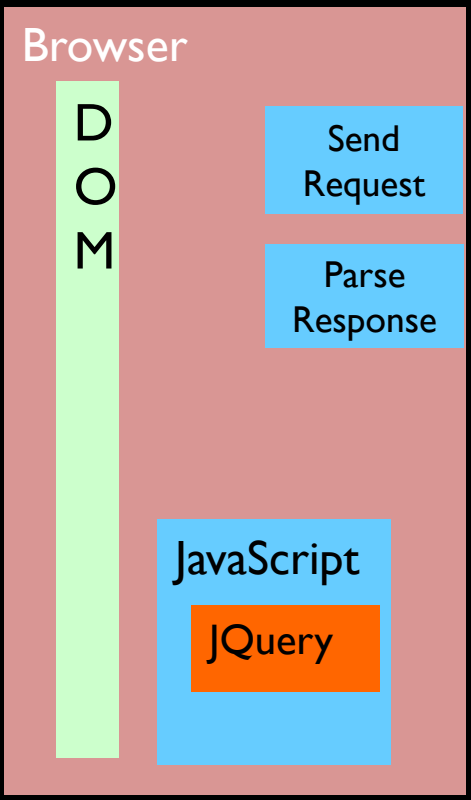
- [www.backbonejs.org](http://www.backbonejs.org)
- [www.emberjs.com](http://www.emberjs.com)
- [www.angular.net](http://www.angular.net)
- [www.reactjs.net](http://www.reactjs.net)
- ...



The only correct way to write JavaScript is whatever you were not doing last week.

@ThePracticalDev

Time

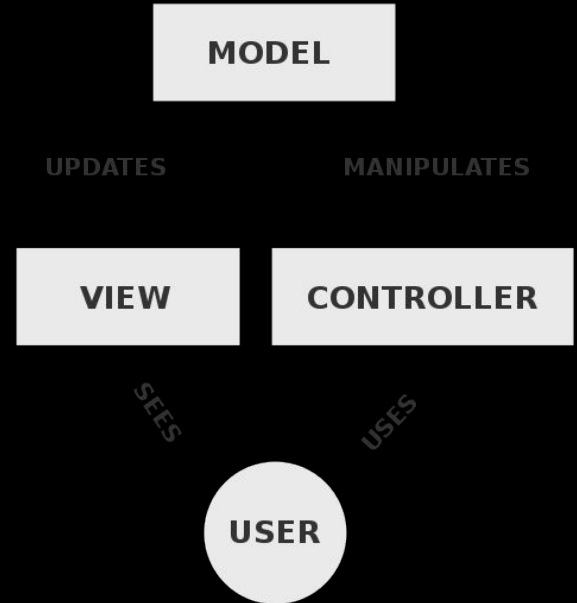


<http://www.wa4e.com/code/rrc/>

# Model-View-Controller

A model that defines the elements of a web application and how they interact

- View – Produce output
- Model – Handle data
- Controller – Orchestration / Routing



<https://en.wikipedia.org/wiki/Model-view-controller>

Model

```
<?php
    $oldguess = '';
    $message = false;
    if ( isset($_POST['guess']) ) {
        // Trick for integer / numeric parameters
        $oldguess = $_POST['guess'] + 0;
        if ( $oldguess == 42 ) {
            $message = "Great job!";
        } else if ( $oldguess < 42 ) {
            $message = "Too low";
        } else {
            $message = "Too high...";
        }
    }
?>
<html>
<head>
    <title>A Guessing game</title>
</head>
<body style="font-family: sans-serif;">
<p>Guessing game...</p>
<?php
    if ( $message !== false ) {
        echo("<p>$message</p>\n");
    }
?>
<form method="post">
    <p><label for="guess">Input Guess</label>
    <input type="text" name="guess" id="guess" size="40"
        value="<?= htmlentities($oldguess) ?>" /></p>
    <input type="submit" />
</form>
</body>
```

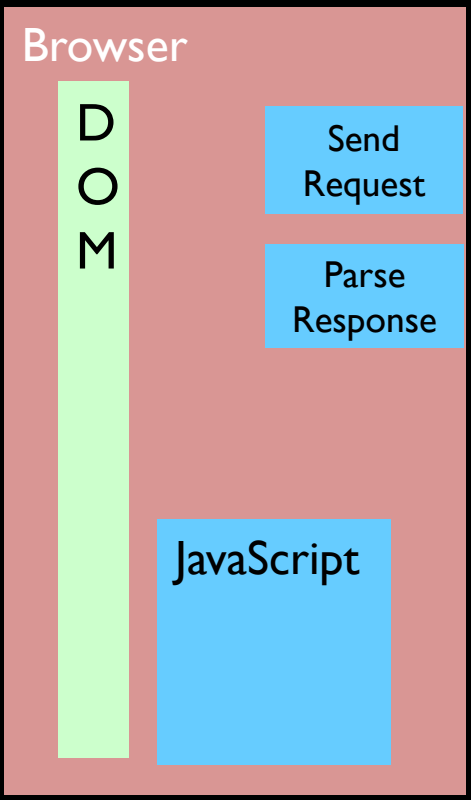
Context

Controller

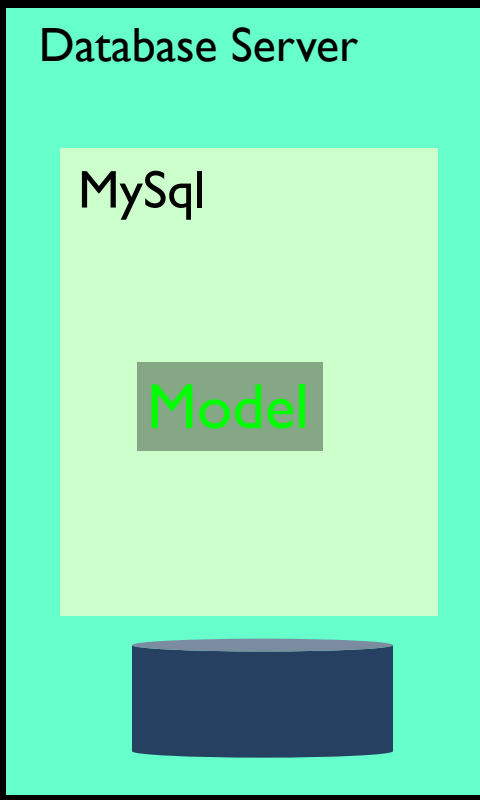
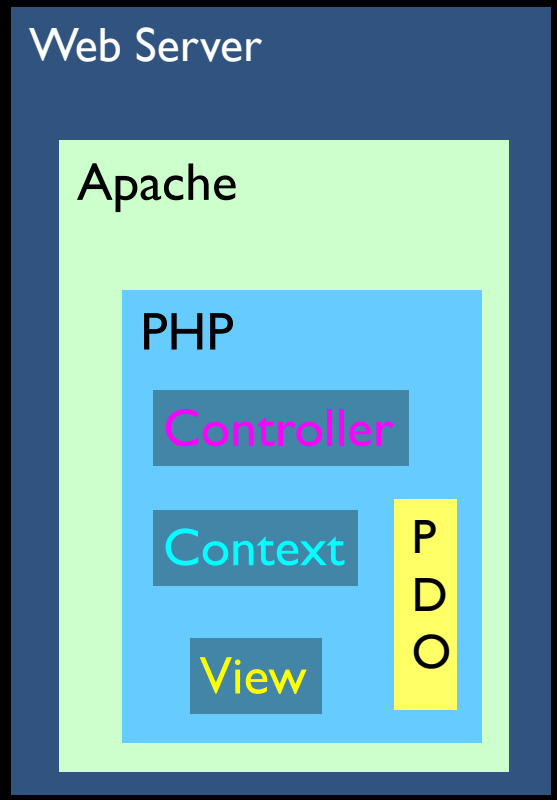
View

guess.php

Time



MVC in PHP



# handlebars.js

- Templates with curly braces
  - Adapted from Django / AppEngine
- Hot spots accept data from the **context**

Context

+

```
<div class="entry">
  <h1>{{title}}</h1>
  <div class="body">
    {{body}}
  </div>
</div>
```

=

```
<div class="entry">
  <h1>Equations</h1>
  <div class="body">
    guess &lt; 42
  </div>
</div>
```

```
<div id="stuff"></div>
```

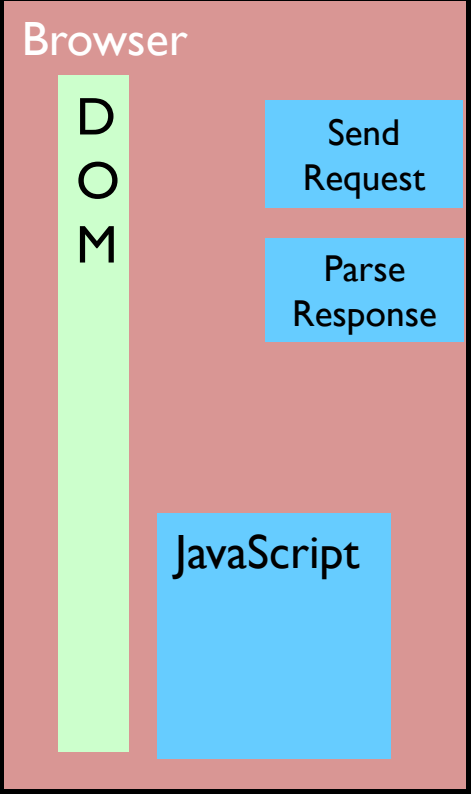
```
<script id="entry-template" type="text/x-handlebars-template">  
  <div class="entry">  
    <h1>{{title}}</h1>  
    <div class="body">  
      {{body}}  
    </div>  
  </div>  
</script>
```

```
<script>  
var raw_template = $("#entry-template").html();  
var template = Handlebars.compile(raw_template);  
var context = {title: "Equations", body: "guess < 42"};  
var rendered = template(context);  
alert('Check out the sweet spinner..:~:');  
$('#stuff').html(rendered);  
</script>
```

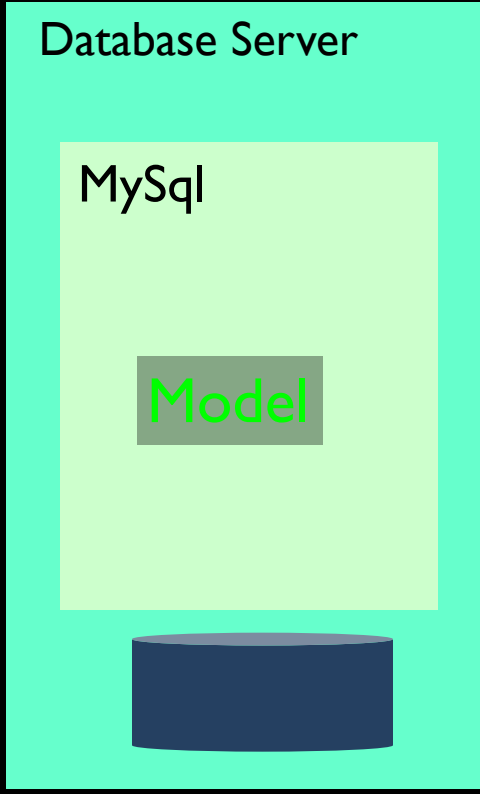
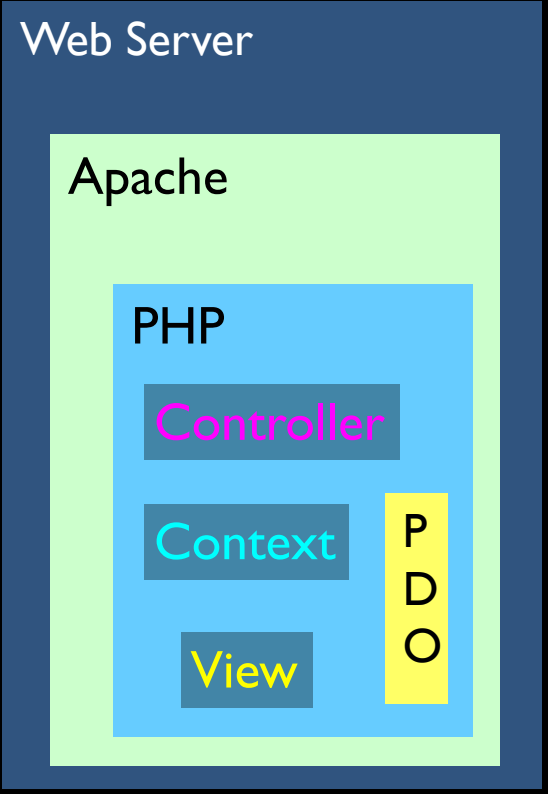
hand-01.htm



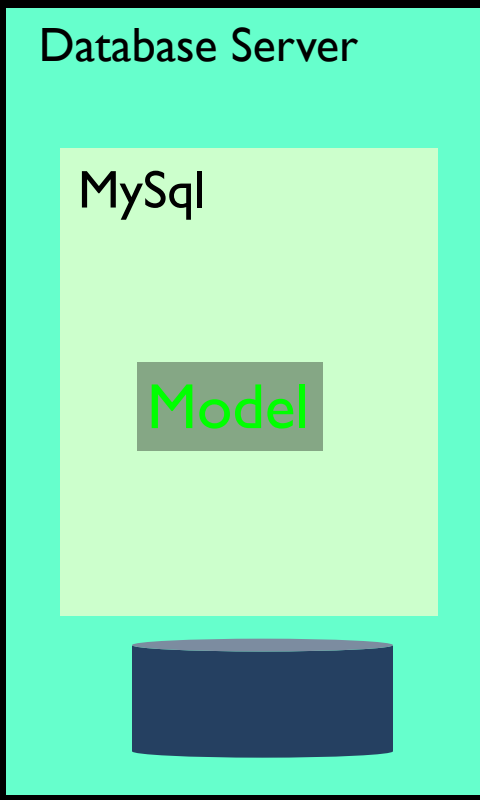
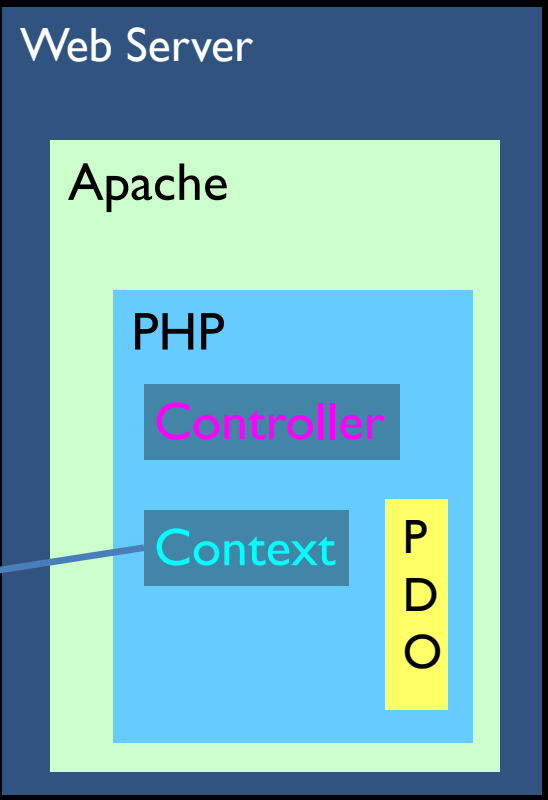
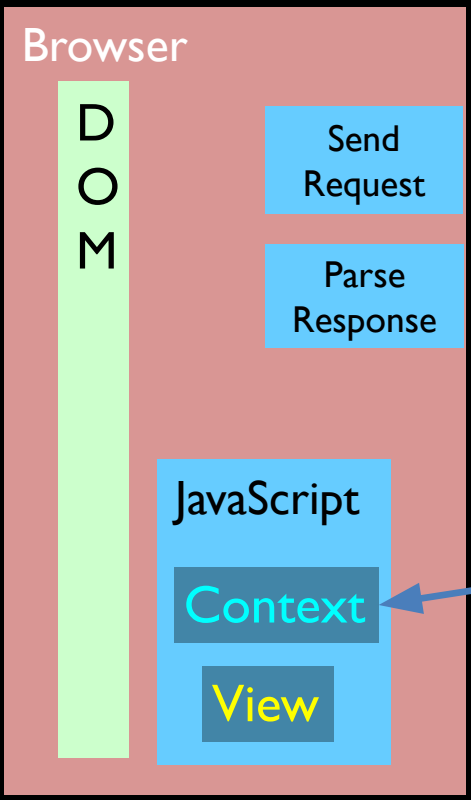
Time



MVC in PHP



Time



View in JavaScript

```
<div id="stuff"></div>
```

```
<script id="entry-template" type="text/x-handlebars-template">
  <div class="entry">
    <h1>{{title}}</h1>
    <div class="body">
      {{body}}
    </div>
  </div>
</script>
```

hand-02.htm

```
<?php
  header('Content-Type: ...');
  $stuff = array(
    'title' => 'Mathematics',
    'body' => 'guess > 42');
  echo(json_encode($stuff));
```

```
<script>
var raw_template = $("#entry-template").html();
var template = Handlebars.compile(raw_template);

$.getJSON('hand-02-json.php', function (context) {
  var rendered = template(context);
  $('#stuff').replaceWith(rendered);
});
</script>
```

```
{ "title": "Mathematics",
  "body": "guess > 42" }
```

# Handlebars Application

```
<script id="list-template" type="text/x-handlebars-template">
  {{#if profiles.length}}
    <p><table border="1">
      <tr><th>Name</th><th>Headline</th>
      {{#if loggedin}}<th>Action</th>{{/if}}</tr>
      {{#each profiles}}
        <tr><td><a href="view.php?profile_id={{profile_id}}">
          {{first_name}} {{last_name}}</a>
        </td><td>{{headline}}</td>
        {{#if ../loggedin}}
          <td>
            <a href="form.php?profile_id={{profile_id}}">Edit</a>
            <a href="delete.php?profile_id={{profile_id}}">Delete</a>
          </td>
        {{/if}}
      </tr>
    {{/each}}
  </table></p>
  {{/if}}
</script>
```

```
<div id="list-area"></div>
```

```
....
```

```
<script>
$(document).ready(function() {
    $.getJSON('profiles.php', function(profiles) {
        window.console && console.log(profiles);
        var source = $("#list-template").html();
        var template = Handlebars.compile(source);
        var context = {};
        context.loggedin =
            <? = isset($_SESSION['user_id']) ? 'true' : 'false' ?>;
        context.profiles = profiles;
        $('#list-area').replaceWith(template(context));
    }).fail(function() { alert('getJSON fail'); });
});
</script>
```

```
<?php
// This script works even if you are not logged in
require_once 'pdo.php';
header("Content-type: application/json; charset=utf-8");

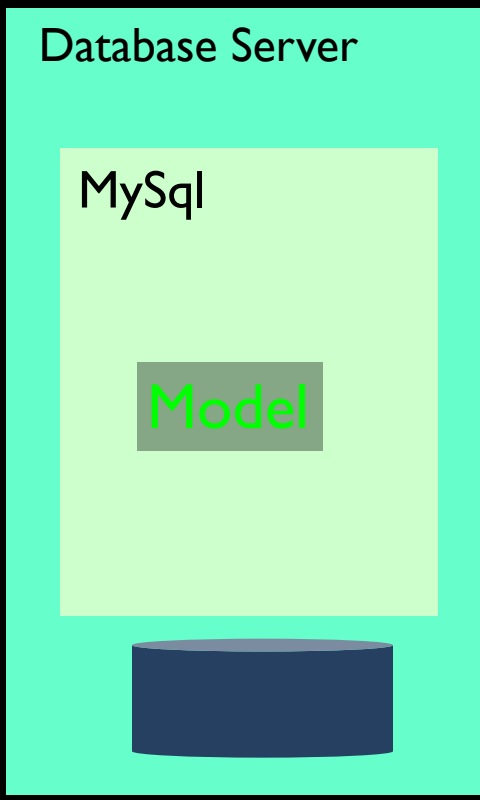
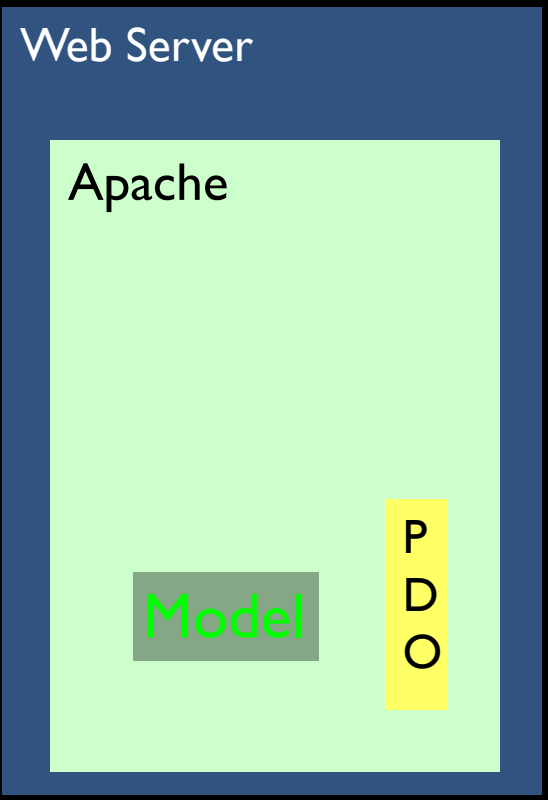
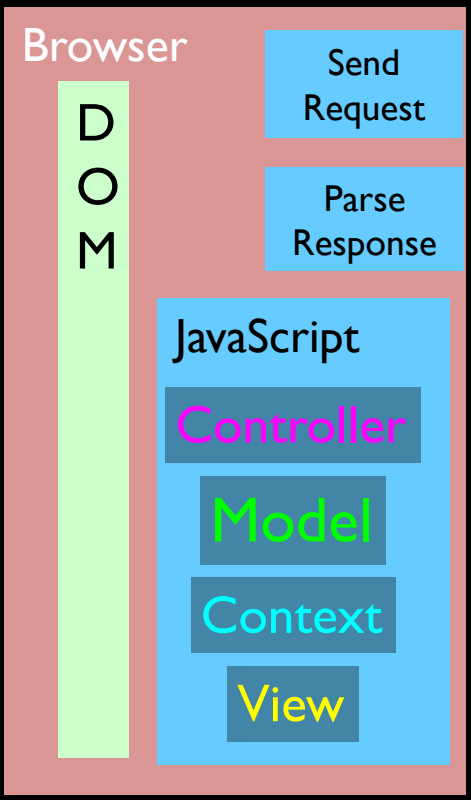
$stmt = $pdo->query('SELECT * FROM Profile');
$profiles = $stmt->fetchAll(PDO::FETCH_ASSOC);
echo(json_encode($profiles, JSON_PRETTY_PRINT));
```

## profiles.php

```
[
  {
    "profile_id": "5",
    "user_id": "1",
    "first_name": "Chuck",
    "last_name": "Severance",
    "email": "csev@example.com",
    "headline": "Python Rulez",
    "summary": "Python",
    "updated_at": "2016-04-04 21:15:27"
  },
  {
    "profile_id": "6",
    "user_id": "1",
    "first_name": "Colleen",
    "last_name": "van Lent",
    "email": "colleen@example.com",
    "headline": "Responsive Design Rulez",
    "summary": "HTML5!",
    "updated_at": "2016-04-04 21:16:12"
  }
]
```



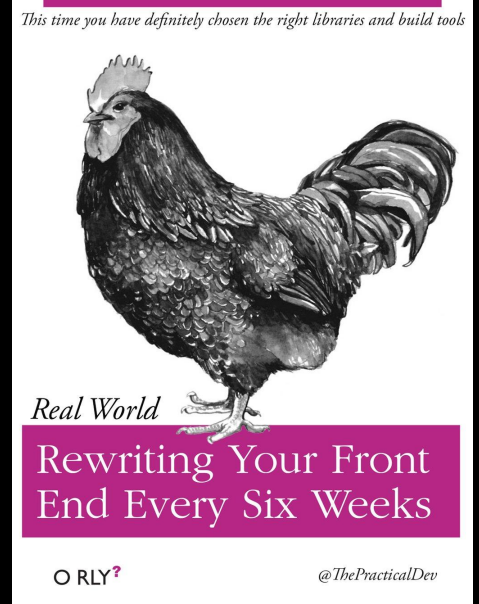
Time



Single Page Application - Angular

# Summary

- With JavaScript ES6 on the way and significant browser improvements on the horizon, it is likely that the “best practice” both on the server and the client will continue to evolve.
- It will continue to be important to understand how web applications work “all the way down” so you can work with these new innovations.



The only correct way to write JavaScript is whatever you were not doing last week.

@ThePracticalDev

# Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) as part of www.wa4e.com and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here