Web Programming

with PHP and Javascript

Jess Fortier
Portland State University

January 23, 2013

THE INTERNET

Files → Server → Network → Browser → User

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
    <head>
        <title>Hello, World!</title>
    </head>
    <body>
        Hello, World!
    </body>
</html>
```

The Internet

Files → Server → Network → Browser → User

- Browser: HTTP Communicator, HTML Interpreter

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
    <head>
        <title>Hello, World!</title>
    </head>
    <body>
        Hello, World!
    </body>
</html>
```

The Internet

Files → Server → Network → Browser → User

- Browser: HTTP Communicator, HTML Interpreter
- HTTP: HyperText Transfer Protocol

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
    <head>
        <title>Hello, World!</title>
    </head>
    <body>
        Hello, World!
    </body>
</html>
```

The Internet

Files → Server → Network → Browser → User

- Browser: HTTP Communicator, HTML Interpreter
- HTTP: HyperText Transfer Protocol
- HTML : HyperText Markup Language

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
    <head>
        <title>Hello, World!</title>
    </head>
    <body>
        Hello, World!
    </body>
</html>
```

HTML & CSS

```
<html>[1]
    <head>
        <title>Bacon Ipsum</title>
    </head>
    <body>
        <p>
            Bacon ipsum dolor sit amet chuck
            sirloin shank andouille.
        </p>
    </body>
</html>
```

HTML & CSS

Attribute-based formatting

```
<html>
    <head>
        <title>Bacon Ipsum</title>
    </head>
    <body bgcolor="black">
        <p color="white" align="center">
            Bacon ipsum dolor sit amet chuck
            sirloin shank andouille.
        </p>
    </body>
</html>
```

HTML & CSS

In-line styles

```
<html>
    <head>
        <title>Bacon Ipsum</title>
    </head>
    <body style="background-color:black;">
        <p style="color:white;
           text-align:center;">
            Bacon ipsum dolor sit amet chuck
            sirloin shank andouille.
        </p>
    </body>
</html>
```

HTML & CSS

Embedded, recycleable styles; separation of content and presentation

```
<html>
    <head>
        <title>Bacon Ipsum</title>
        <style>
            body{background-color:black;}
            p{color:white;}
            .centered{text-align:center;}
        </style>
    </head>
    <body>
        <p class="centered">
            Bacon ipsum dolor sit amet chuck
            sirloin shank andouille.
        </p>
    </body>
</html>
```

HTML & CSS

External, recycleable styles; separation of content and presentation

```
<html>
    <head>
        <title>Bacon Ipsum</title>
        <link rel="stylesheet" type="text/css"
           href="style.css" />
    </head>
    <body>
        <p class="centered">
            Bacon ipsum dolor sit amet chuck
            sirloin shank andouille.
        </p>
    </body>
</html>
```

HTML & CSS

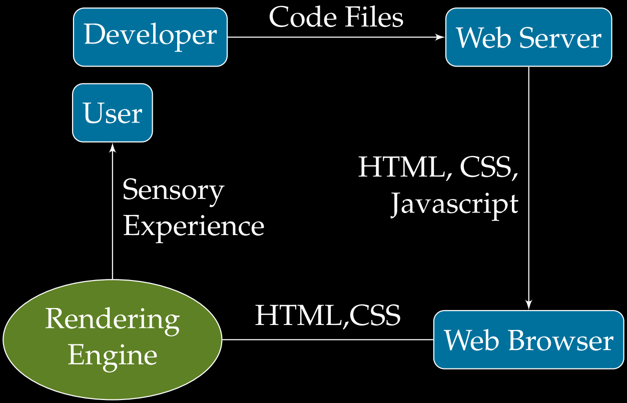
Trends in the Tech:

- maximizing code reuse
- improving semantics and modularity
- increasing interactivity and media support

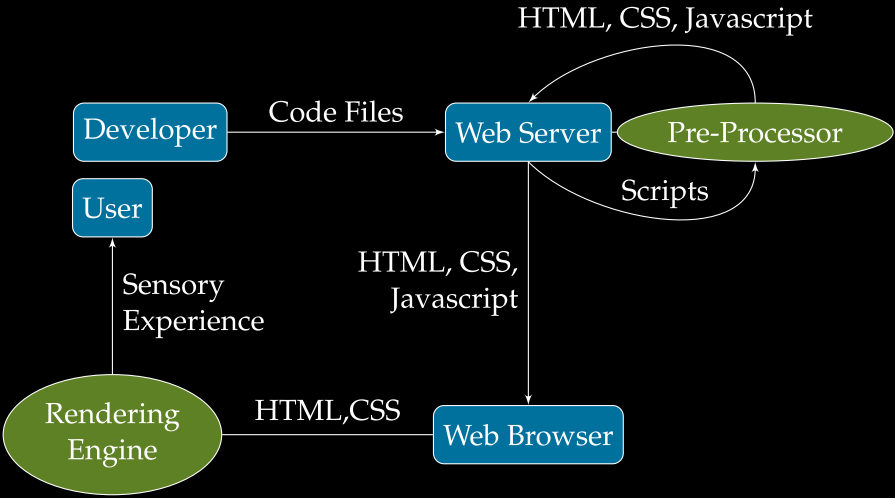
Trends in the Tools:

- generating code automatically
- enforcing MVC design patterns
- augmenting code dynamically

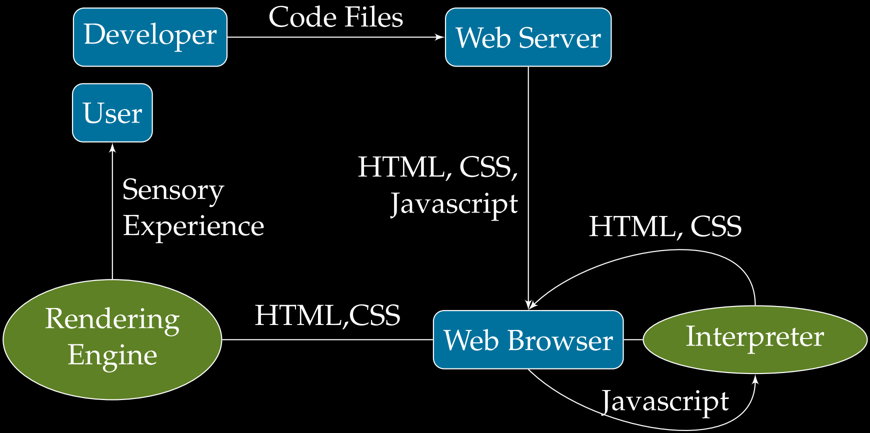
Web Data Flow



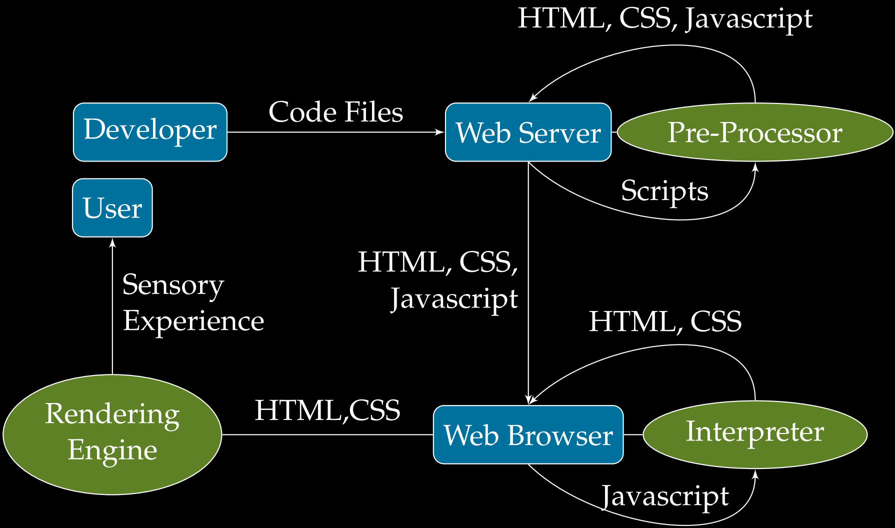
Web Data Flow



Web Data Flow



Web Data Flow



Common Technologies

Server-side Scripting

- PHP
- Javascript
- Ruby
  - Haml
  - SASS/SCSS
- ASP
- JSP

Client-side Scripting

- Javascript
  - CoffeeScript

Fewer Client-Side Options

Adding client-side support for a language requires the browser to bundle an interpreter for that language, or for reliable plugins to be published.

Common Technologies

Server-side Scripting

- PHP
- Javascript
- Ruby
  - Haml
  - SASS/SCSS
- ASP
- JSP

Client-side Scripting

- Javascript[2]
  - CoffeeScript

Fewer Client-Side Options

Adding client-side support for a language requires the browser to bundle an interpreter for that language, or for reliable plugins to be published.

PHP

PHP: Hypertext Preprocessor[3]

```
<?php
    $myVar = 'Hello World!';
    echo $myVar;
?>
```

- Imperative
- Object Oriented (Class-based)
- Loosely typed
- Embeddable
- Good documentation of syntax and libraries
- Differentiates between single quotes and double quotes

JAVASCRIPT

Not related to Java!

- Imperative
- Object Oriented (Prototype-based)
- Functional
- Loosely typed
- Embeddable
- Server- and client-side
- Many robust libraries for out-of-the-box functionality
  - jQuery
  - mooTools
  - Dojo
  - Node.js

Web Programming Tools

- Install an AMP stack (Apache server, MySQL, PHP)
- Get a good, lightweight text editor[4]
- Extend your browser's inspection tools (Optional)

Windows

- WAMP
  XAMPP
- Sublime Text 2
  Notepad++

Mac

- MAMP
  XAMPP
- Sublime Text 2
  TextWrangler

Linux

- BitNami LAMP
  XAMPP
- Sublime Text 2
  VIM

Chrome

- Firebug Lite

Firefox

- Firebug

Safari

- Firebug Lite

A Basic PHP Example

```
<?php
    $employeeArray =
            array(’Tom’,’Jerry’,’Jack’,’Sally’);

    for($i = 0; $i < sizeof($myBasicArray); $i++){
        echo $myBasicArray[$i];
    }
?>
```

A Basic PHP Example

Displayed: Names appear on the same line without spaces

```
<?php
    $employeeArray =
            array(’Tom’,’Jerry’,’Jack’,’Sally’);

    for($i = 0; $i < sizeof($myBasicArray); $i++){
        echo $myBasicArray[$i];
    }
?>
```

A Basic PHP Example

```
<?php
    $employeeArray =
            array(’Tom’,’Jerry’,’Jack’,’Sally’);

    for($i = 0; $i < sizeof($myBasicArray); $i++){
        echo $myBasicArray[$i] . ’\n’;
    }
?>
```

A Basic PHP Example

Displayed: Each name has its own line in source, but not on the web page

```
<?php
    $employeeArray =
            array('Tom','Jerry','Jack','Sally');

    for($i = 0; $i < sizeof($myBasicArray); $i++){
        echo $myBasicArray[$i] . '\n';
    }
?>
```

A Basic PHP Example

```
<?php
    $employeeArray =
            array(’Tom’,’Jerry’,’Jack’,’Sally’);

    for($i = 0; $i < sizeof($myBasicArray); $i++){
        echo $myBasicArray[$i] . ’<br />’;
    }
?>
```

A Basic PHP Example

Displayed: Each name has its own line on the page, but not in source

```
<?php
    $employeeArray =
            array(’Tom’,’Jerry’,’Jack’,’Sally’);

    for($i = 0; $i < sizeof($myBasicArray); $i++){
        echo $myBasicArray[$i] . ’<br />’;
    }
?>
```

An Embedded PHP Example

```
<html>
    <head><title>Embedded PHP Demo</title></head>
    <body><h1>Employees</h1>
        <?php $associativeEmployeeArray = array(
                 ’Tom’ => ’Web Developer’,
                 ’Jerry’ => ’Project Manager’,
                 ’Jack’ => ’Designer’,
                 ’Sally’ => ’Software Engineer’); ?>
        <table border="1" cellpadding="3" cellspacing="0">
            <thead><tr><td>Name</td>
                    <td>Job Title</td></tr></thead>
            <tbody><?php
                foreach ($associativeEmployeeArray as
                  $employee => $title){
                    echo "<tr><td>$employee</td>
                            <td>$title</td></tr>";
                }
         ?></tbody>
        </table>
    </body>
</html>
```

A Javascript Form Example

```
<html>
    <head><title>Javascript Demo</title>
    <script>
        function addEmployee(){
            var name = document.getElementById("employee-name").value;
            var job = document.getElementById("job-title").value;
            var row = "<tr><td>"+name+"</td><td>"+job+"</td></tr>";
            document.getElementById("employees").innerHTML += row;
        }
    </script> </head>
    <body>
        <form name="new-employee">
            Name: <input type="text" id="employee-name"><br />
            Title: <input type="text" id="job-title"><br />
            <input type=’button’ onclick=’addEmployee()’ value=’Submit Emp
        </form>
        <table>
            <thead><tr><td>Name</td><td>Job Title</td></tr></thead>
            <tbody id="employees"></tbody>
        </table>
    </body>
</html>
```

A Keyboard Input Example

```
<style type="text/css">
    #stickman{position:absolute; left:0; top:0; height:100px;}
</style>
<script type="text/javascript">
    function Stickman() { this.x = 0; this.y = 0; }

    Stickman.prototype.moveLeft = function() {
        this.x -= 1; this.draw();
    }; // Also moveRight, moveUp, and moveDown

    Stickman.prototype.draw = function() {
        document.getElementById(’stickman’).style.left = this.x;
        document.getElementById(’stickman’).style.top = this.y;
    } var mrStick = new Stickman();

    window.addEventListener(’keydown’, function(event) {
        switch (event.keyCode) {
            case 37: mrStick.moveLeft(); break; // Also 38, 39, 40
    } }, false);
</script>

<body> <img src="stickman.png" id="stickman" /> </body>
```

AN ANIMATION EXAMPLE

```
<script type="text/javascript">
    function Stickman() { this.x = 0; this.y = 0; this.dir = ’R’; }

    Stickman.prototype.changeDir = function(dir) { this.dir = dir; }

    Stickman.prototype.draw = function() {
        document.getElementById(’stickman’).style.left = this.x;
        document.getElementById(’stickman’).style.top = this.y;
    }

    Stickman.prototype.move = function() {
        switch (this.dir) {
            case ’L’: this.x -= 1; break; // Also ’U’, ’R’, and ’D’
        } this.draw();
    } var mrStick = new Stickman();

    window.addEventListener(’keydown’, function(event) {
        switch (event.keyCode) {
            case 37: mrStick.changeDir(’L’); break; // Also 38, 39, an
    } }, false);

    setInterval(’mrStick.move()’, 20);
</script>
```

Snake: Requirements

- Behavior should be consistent with traditional gameplay[5]
  - The game animates itself
  - The user directs the head of the snake
  - Colliding with goals lengthens the snake
  - Colliding with walls or the tail ends the game
- Main game controls should be the keyboard arrow keys
- Minimum game parameters should be set via PHP form:
  - Board size
  - Snake pace
  - Number of simultaneous goals
- Score should be tracked and displayed during play
- An external stylesheet should be present to skin the app
- There should be no javascript errors when the app runs

Snake: Extra Credit Ideas

- Implement WASD controls
  - Let single players choose which set to use
  - Allow for two concurrent players to compete
- Provide additional game options accessible through the UI
  - Game color scheme (snake, goals, background)
  - Specialty board selection (e.g. - internal walls)
- Implement an optional leveling system which might
  - Also increase speed when goals are consumed
  - Change the color scheme when goals are consumed
  - Introduce timed bonus rounds where collisions don't count
- Connect to a database to store high scores
- Validate form elements
- Persist form elements if validation fails

End notes

[1] Check out w3schools to read up on HTML.

[2] Check out Tizag's Javascript tutorials

[3] PHP is a recursive acronym

[4] For our purposes, heavy duty editors like ”Eclipse for PHP Developers” will be cumbersome and unnecessary. If given the choice between your computer's basic text editing environment and a bulky project management editor, choose the former.

[5] Play some Snake online to get the hang of it