Web programming 2

Material taken from:

CS215 Web Oriented Programming @ University of Regina

and

Web Programming Step by Step by Stepp, Miller & Kirst

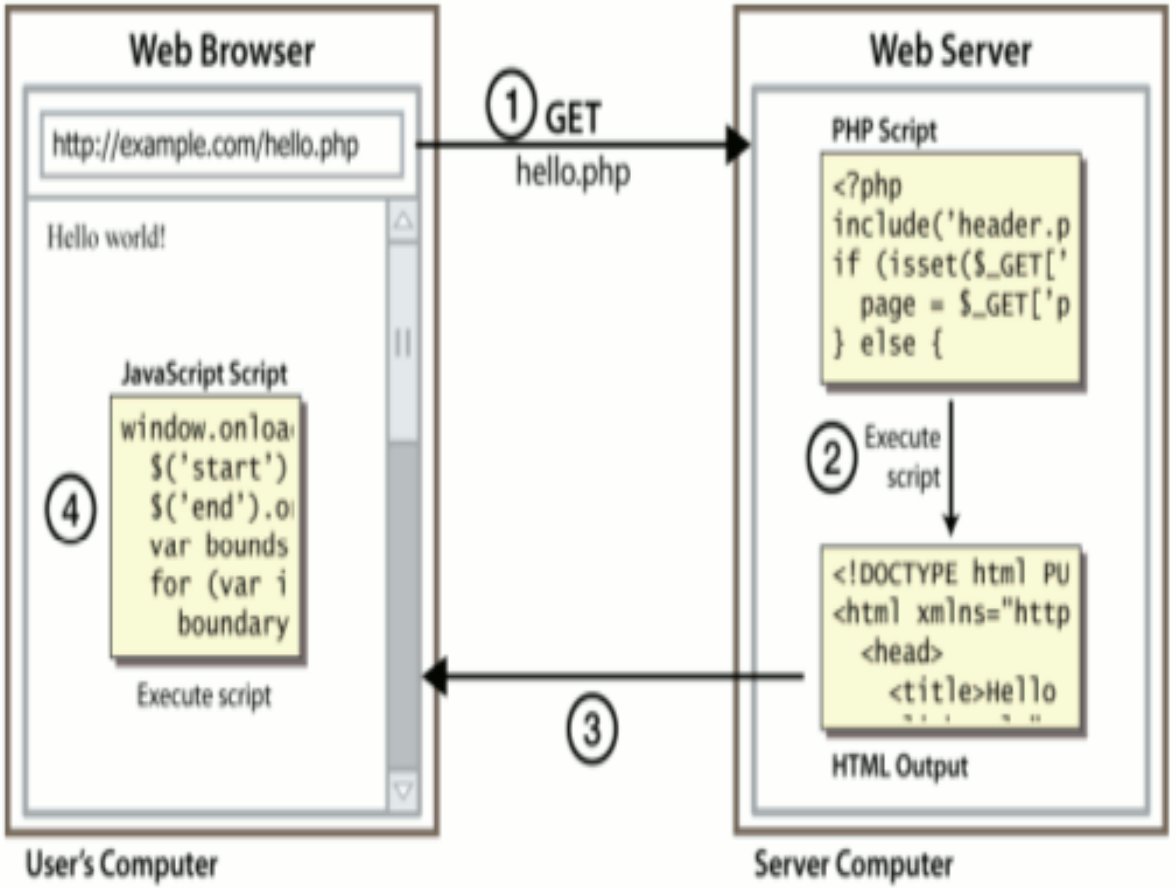
Basic Javascript

Output

Events

The DOM

Client-side evaluation



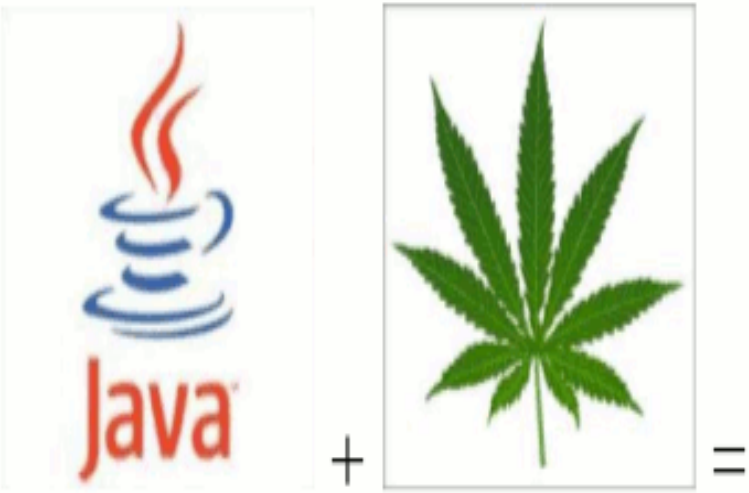
- **client-side script**: code runs in browser after page is sent back from server
  - often this code manipulates the page or responds to user actions

What is JavaScript?

- a lightweight programming language ("scripting language")
- used to make web pages interactive
  - insert dynamic text into HTML (ex: user name)
  - react to events (ex: page load user click)
  - get information about a user's computer (ex: browser type)
  - perform calculations on user's computer (ex: form validation)
- a web standard (but not supported identically by all browsers)
- NOT related to Java other than by name and some syntactic similarities

JavaScript vs. Java

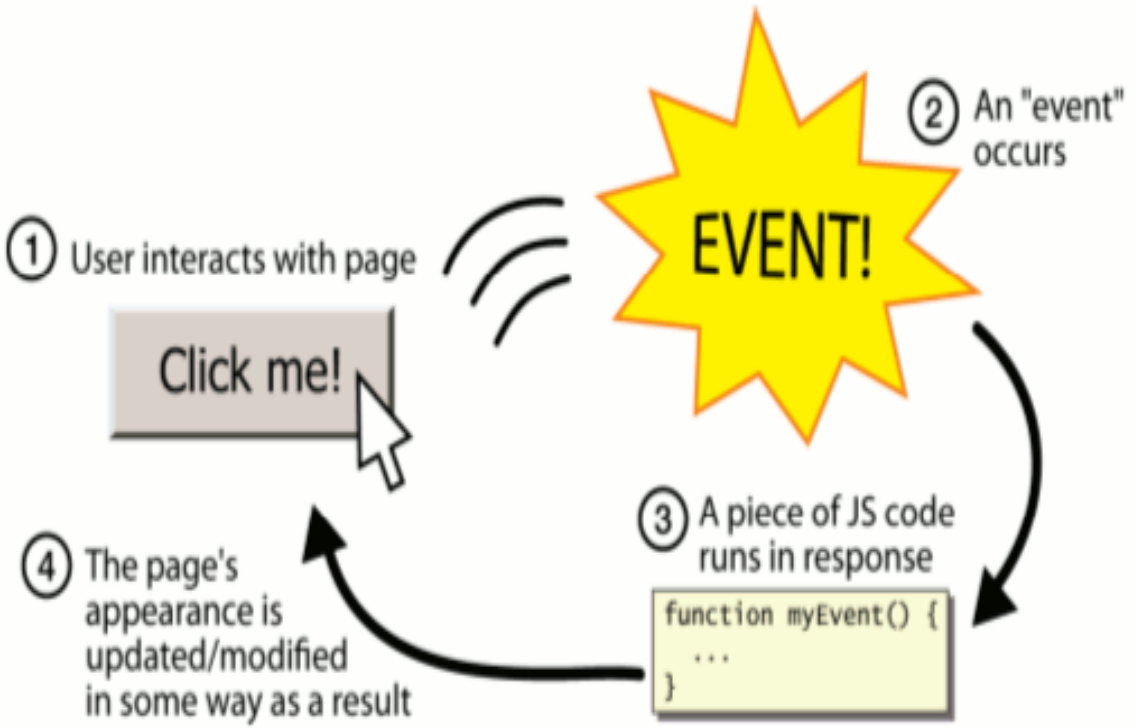
- **interpreted**, not compiled
- more relaxed syntax and rules
  - fewer and "looser" data types
  - variables don't need to be declared
  - errors often silent (few exceptions)
- key construct is the **function** rather than the class
  - "first-class" functions are used in many situations
- contained within a web page and integrates with its HTML/CSS content



Javascript Basics

- Javascript basics
- Demo programs
  - To unpack: ***tar xvfz JSdemos.tar.gz***

Event-driven programming



- you are used to programs start with a `main` method (or implicit `main` like in PHP)
- JavaScript programs instead wait for user actions called **events** and respond to them
- **event-driven programming**: writing programs driven by user events
- Let's write a page with a clickable button that pops up a "Hello, World" window...

Event handlers

```HTML
<element attributes onclick="function();">...
```

```HTML
<button onclick="myFunction();">Click me!</button>
```

Click me!

output

- JavaScript functions can be set as **event handlers**
  - when you interact with the element, the function will execute
- `onclick` is just one of many event HTML attributes we'll use

- but popping up an `alert` window is disruptive and annoying
  - A better user experience would be to have the message appear on the page...

- Soooo many events
- On-line example programs

Linking to a JavaScript file: `script`

```HTML
<script src="filename" type="text/javascript"></script>
```

```HTML
<script src="example.js" type="text/javascript"></script>
```

- `script` tag should be placed in HTML page's `head`
- script code is stored in a separate `.js` file
- JS code can be placed directly in the HTML file's `body` or `head` (like CSS)
  - but this is bad style (should separate content, presentation, and behavior)

More on Output

- Four ways to output information:
  - Use alert function
  - Write to the document object
  - Manipulate HTML elements
  - Write to the console
- On-line examples

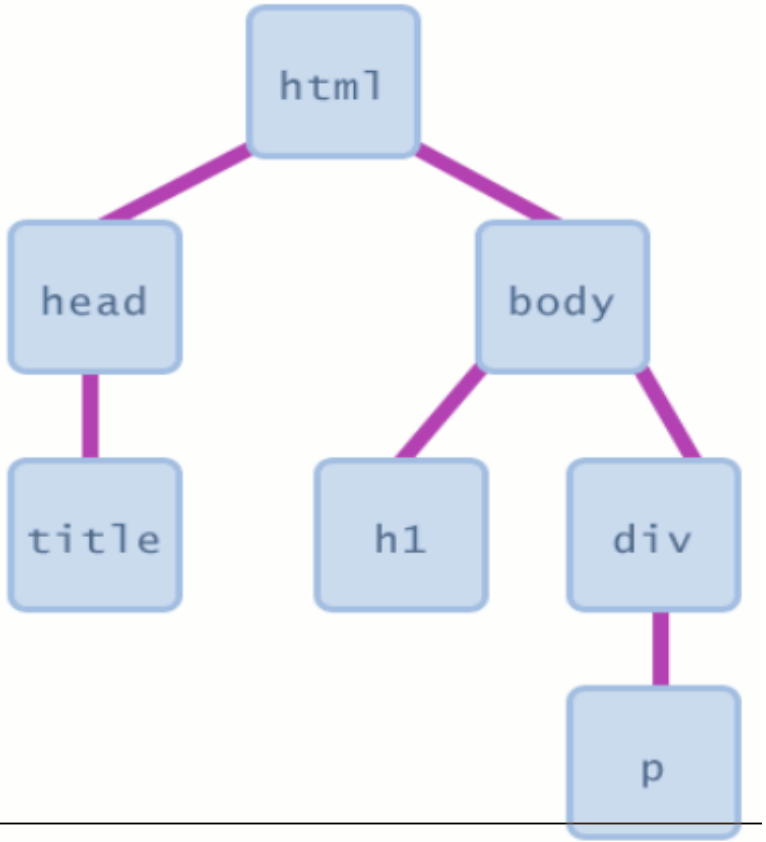
The DOM

- Harvard video

Document Object Model (DOM) (7.1.4)

a set of JavaScript objects that represent each element on the page

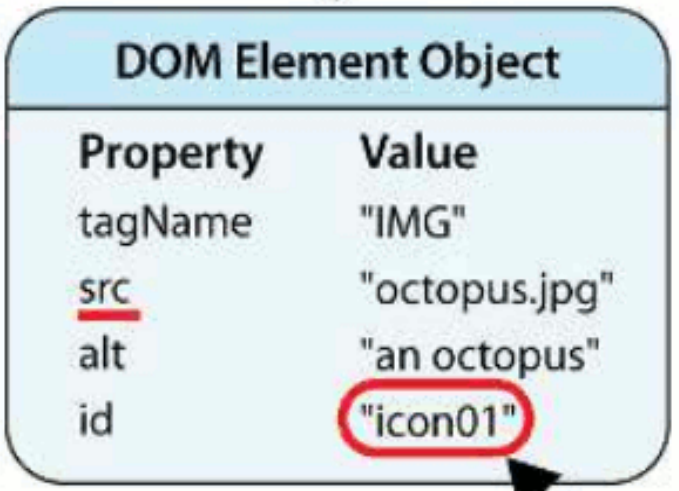
- most JS code manipulates elements on an HTML page
- we can examine elements' state
  - e.g. see whether a box is checked
- we can change state
  - e.g. insert some new text into a `div`
- we can change styles
  - e.g. make a paragraph red



DOM element objects

HTML

```
<p>
  Look at this octopus:
  <img src="octopus.jpg" alt="an octopus" id="icon01" />
  Cute, huh?
</p>
```



JavaScript

```
var icon = document.getElementById("icon01");
icon.src = "kitty.gif";
```

- every element on the page has a corresponding DOM object
- access/modify the attributes of the DOM object with *objectName . attributeName*

DOM Changing Examples

- On-line examples
- A collection of examples from *Web Programming Step by Step*
  - Unpack with ***tar xvfz DOMexamples.tar.gz***

The $ function

```
$("id")
```
JS

- returns the DOM object representing the element with the given `id`
- short for `document.getElementById("id")`
- often used to write more concise DOM code:

```
$("footer").innerHTML = $("username").value.toUpperCase();
```
JS

In-class Exercise

- Work the firebug tutorial from USC

  Show me your work before you leave class