

Information Technologies

Anselm Spoerri PhD (MIT)

SC&I @ Rutgers University

aspoerri@rutgers.edu

anselm.spoerri@gmail.com

Lecture 7 - Overview

Dynamic Web

- JavaScript | PHP | MySQL | Apache

Programming Concepts

JavaScript and Document Object Model (DOM)

- Data Types, Variables, Operators, Statements
- Function and Control Structures
- Document Object Model (DOM)
- Image Rollovers

Lectures – Week 7 Content

<http://comminfo.rutgers.edu/~aspoerri/Teaching/InfoTech/Lectures.html#week7>

Web Pages Created So Far

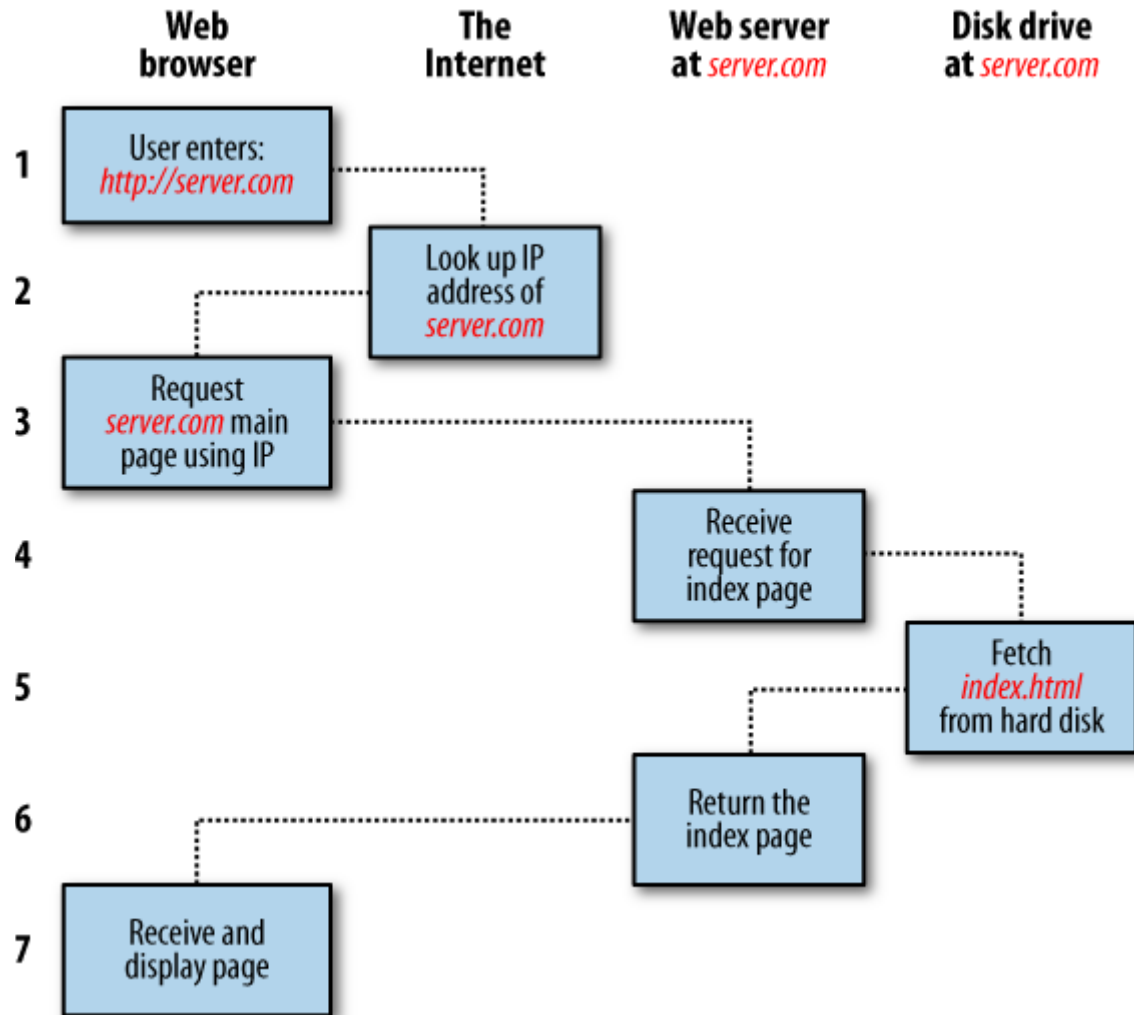
Static HTML Pages

- Same each time

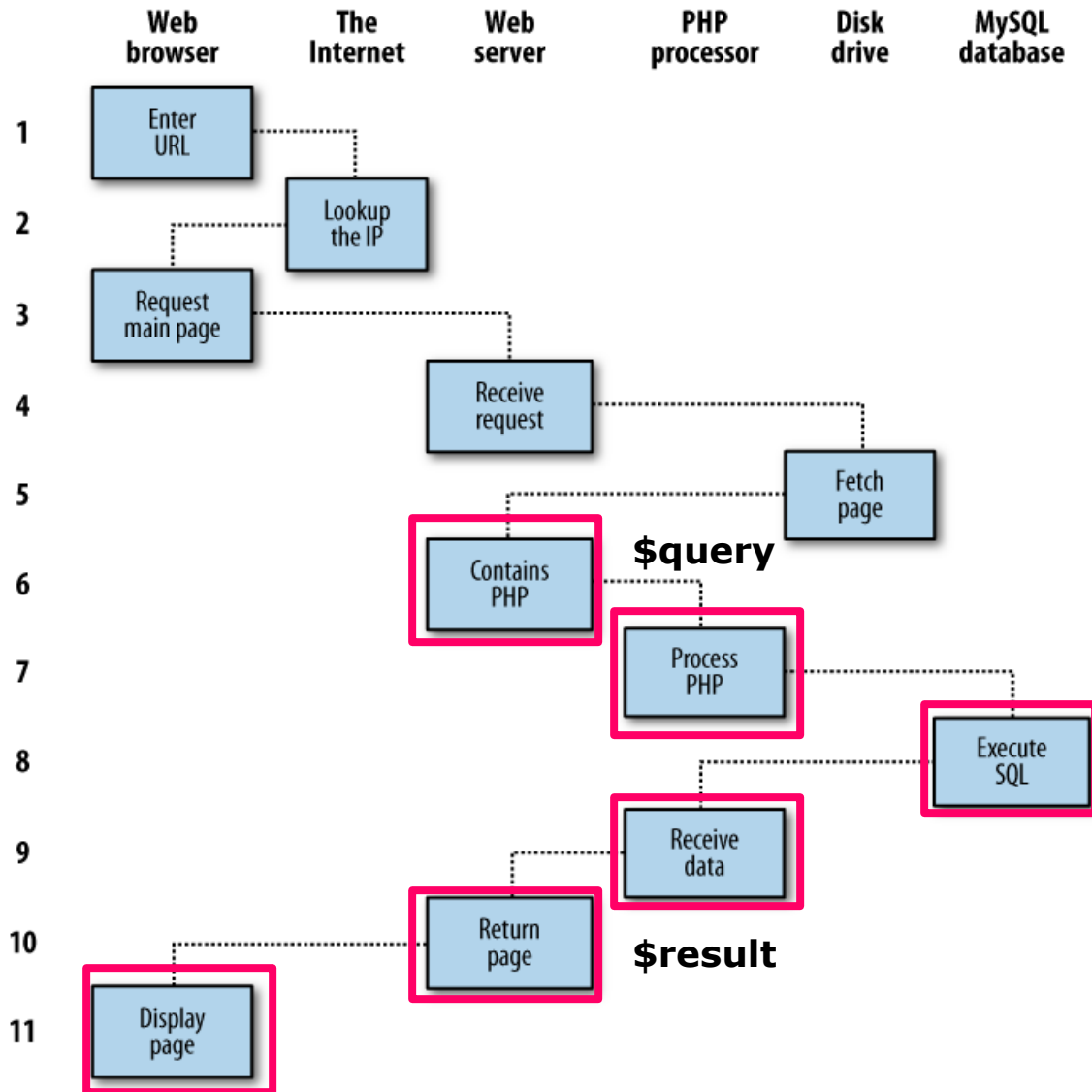
Want Interactive or Dynamic Web Pages

- Depending on browser, time of day or user behavior or ...
different appearance
- Processing user inputs data via **HTML forms**
- Need some sort of **Web programming** (JavaScript, PHP)

Basic Client/Server Request/Response Sequence



Dynamic Client/Server Request/Response Sequence



Dynamic Web

1. JavaScript

- **Client-side scripting** access to elements of HTML document
- Perform data handling and web server requests in background

2. PHP

- **Server-side scripting**
- Open source, simplicity and built-in links to MySQL database

3. MySQL

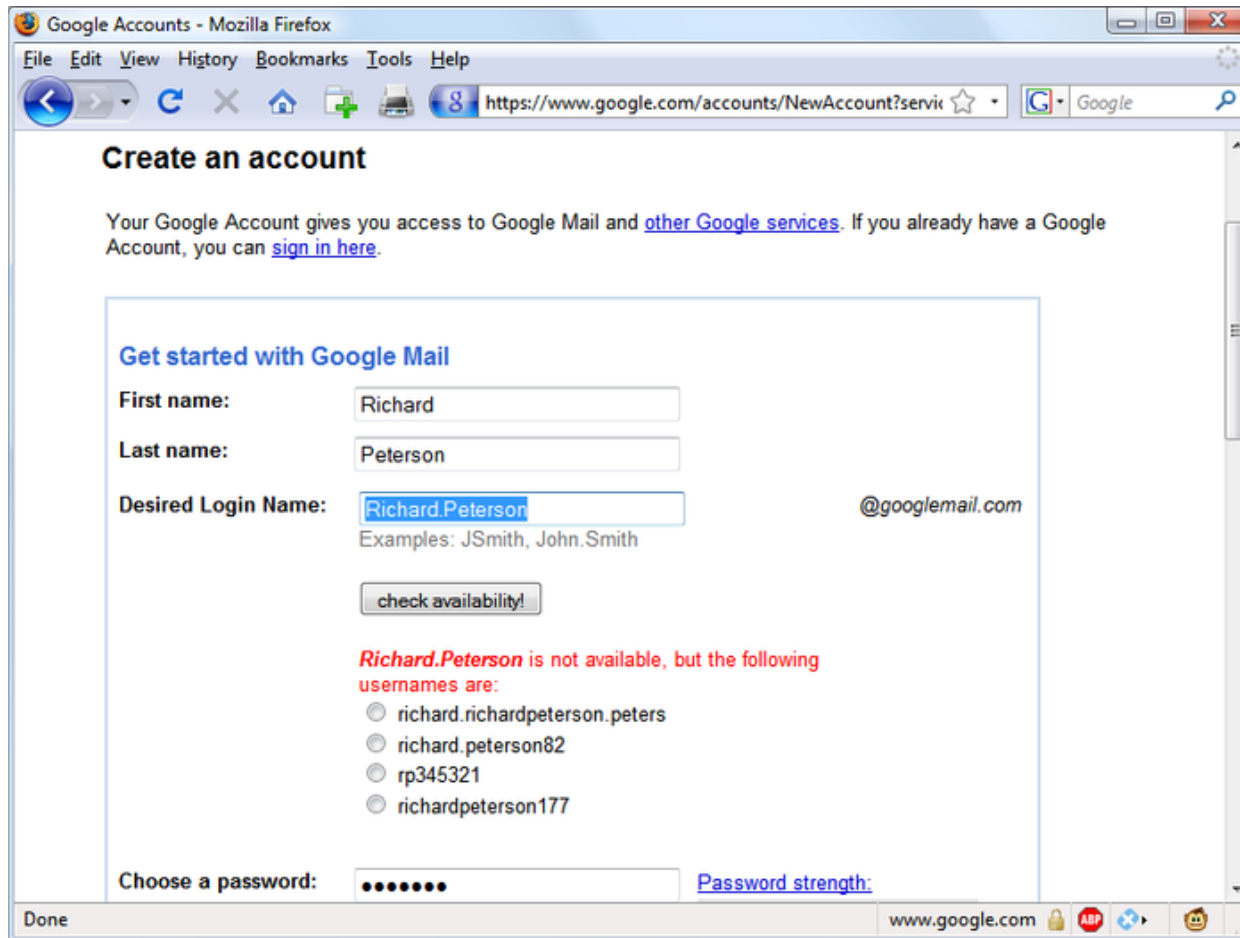
- Open-source **Relational Databases** that supports **structured queries** and is free to use and install on web servers

4. Apache Server

- Open source server software that serves up HTML, media files etc

PHP handles main work on web server, MySQL manages data, and JavaScript looks after web page presentation. JavaScript can also talk with your PHP code on the web server whenever it needs to update something (either on the server or on the web page).

Dynamic Web Page – Gmail Signup Page



<http://mail.google.com/mail/signup>

Dynamic Web, AJAX and Web 2.0

Dynamic Web

- **JavaScript** specifies web page **interactivity** and presentation
- **PHP** handles main work on web server and retrieves data
- **MySQL** manages and stores data
- **JavaScript** performs data handling & **server requests** in background
- **JavaScript** can also talk with **PHP** code on the web server to update data or presentation (either on server or web page)

JavaScript used for **AJAX** and **Web 2.0**

“Asynchronous JavaScript and XML”

- Ajax is the main process behind what is now known as Web 2.0
- Process of accessing the web server in the background

Setups used for Developing Dynamic Web Services

WAMP = Windows, Apache, MySQL, and PHP

MAMP = Mac, Apache, MySQL, and PHP

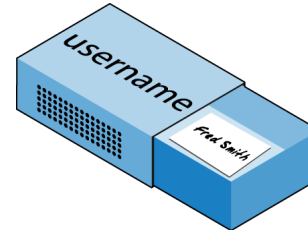
LAMP = Linux, Apache, MySQL, and PHP

Key Programming Concepts

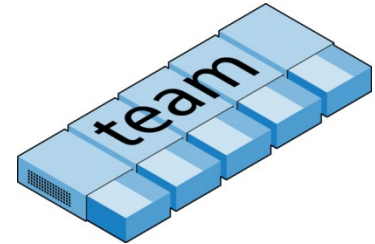
Key Programming Concepts

data types numbers, text, boolean (true, false)

variable = container of data



array = collection of data organized in a sequence



statement = combination of variables, operators and data

→ group of statements

→ **function** = collection of statements to perform specific task

perform test: true or false

if (test) then

for loop: where do we start? when are done? how to continue?

Programming Concepts – Overview

Data Types: such as numbers, text, boolean (true, false)

- Integer | Boolean | Character | String | Floating-Point
- **Array:** [5, 4, 7] (use FOR loop to examine each array item)

Variable: is way to symbolically to refer to data content

- **scores** = [5, 4, 7];

Operators: can be used to perform built-in operations

- **5 +5;** (where + is addition operator)

Statement: combination of variables, operators and data

- **total = 12 * monthly_average;**

Programming Concepts – Overview

Function: collection of statements to perform specific task

```
function add_scores (score1, score2)
{
    return (score1 + score2);
}

total = add_scores (3, 4);
```

Conditional Programming used so that operations are performed only if certain condition is met:

- **IF test** (only do something if certain condition is true)
- **FOR loop** (keep doing something only if certain condition is true)
- **WHILE loop** (keep doing something until certain condition not true anymore)

Debugging code is essential part of programming ...
chasing typos, missing parentheses :)

JavaScript

JavaScript

- **Client-side** scripting language
- **Programs** embedded as **plain text** in HTML of Web page,
`<script type="text/javascript"> ... </script>`
- Since text, **viewable** to the world and for **security reasons**, scripts are **limited** in what they can do
- **Browser executes program** as it loads page, integrating dynamic output of program with static content of HTML
- Used to access to **elements of HTML document**.
- **Validate** form data
- **Create dynamic user interaction** such as checking email address validity in input forms and interactive page elements ...
- **Major implementation differences** between different browsers.

JavaScript – Data Types, Variables, Operators, Statements

Data Types – only three primitive data types

Boolean: true, false **Number:** 5, 7.45 **String:** “Hello World”

Variables = holds value of specific data type

- sequences of letters, digits, and underscores
- start with a letter or underscore
- variables names are **case-sensitive**

sci_550 = true; create **variable sci_550** and set it to true

var count = 100; create **variable count** and set it to 100

Operators

count = 15 + 8 - 7 count += 5 n++ count = 5 * temp
“Hello” + “World”

Statements

Instructions end with **semicolon** ... If missing, automatically inserted

Assignment : total = earn1 – spend1

Return value from function: return total

JavaScript – Arrays

Array stores a sequence of items, accessible via an **index**

```
items = new Array(10); // allocates space for 10 items
```

```
items = new Array(); // if no size given, will adjust dynamically
```

```
items = [0,0,0,0,0,0,0,0,0,0]; // can assign size & values []
```

```
items.length // the number of items in the array
```

To **access array element**, use **[index]**

```
for (i = 0; i < items.length; i++) {
```

```
    document.write(items[i] + "<br>"); // displays elements
```

```
}
```

JavaScript : Function

Functions **function** name (para1, para2, ...) { ... }

Reusable code that consists of **Collection of "statements"**

Takes one or more values as **input "parameters"**

Returns "result" value

```
function convertToCelsius(f) {  
    var celsius = 5/9 * (f-32);  
    return celsius;  
}  
var f = 50;  
c = convertToCelsius(f);
```

JavaScript : Control Structures

Control Structures

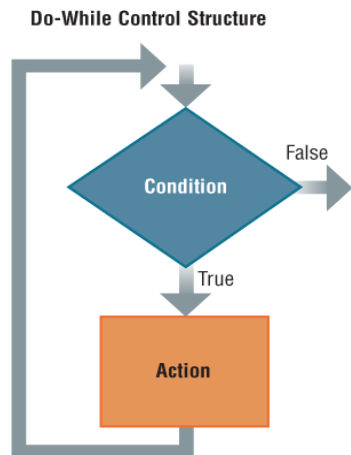
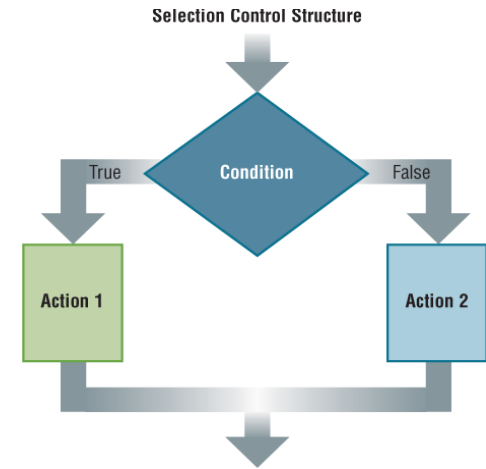
Sequential top to bottom

Conditional **if (test) { ... } else { ... }**

Repetition **for (i=0; i<100; i++) { ... }**

Boolean Operators

x == y true if x and y are **equal**
x != y true if x and y are **not equal**
x > y true if x is **greater** than y
x <= y true if x is **smaller** than or equal to y
x && y true if both x **and** y are true
x || y true if either x **or** y is true
!x true if x is false



JavaScript – Stored Where

Location of JavaScript

External file

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

External .js file: must **not include** any `<script>` or `</script>` tags

<head>

```
<script type="text/javascript">script code</script>
```

Place code here to be available when **page loads**

<body>

```
<script type="text/javascript">
```

```
document.write("<h2>Using JavaScript</h2>");
```

```
document.write("Hello World. Today is " + Date() );
```

```
</script>
```

Places h2 and "Hello World. Today is [current date]" in page where JavaScript is located

JavaScript Example

```
<html>
  <head><title>Hello World</title></head>
  <body>
    <script type="text/javascript">
      document.write("Hello World") ;
    </script>
    <noscript>
      Your browser doesn't support or has disabled JavaScript.
    </noscript>
  </body>
</html>
```

Tasks

- Change Text and apply h2 tag
- Add "+ Date()" to text being displayed
- Add paragraph before and after where JavaScript is inserted

<http://comminfo.rutgers.edu/~aspoerri/Teaching/InfoTech/Lectures/Lec7/Steps/HelloWorld.html>

JavaScript and Document Object Model

Document Object Model

http://en.wikipedia.org/wiki/Document_Object_Model

Platform- and language-neutral interface that allows programs and scripts to **dynamically access** and update the **content**, **structure** and **style** of **documents**.

Examples

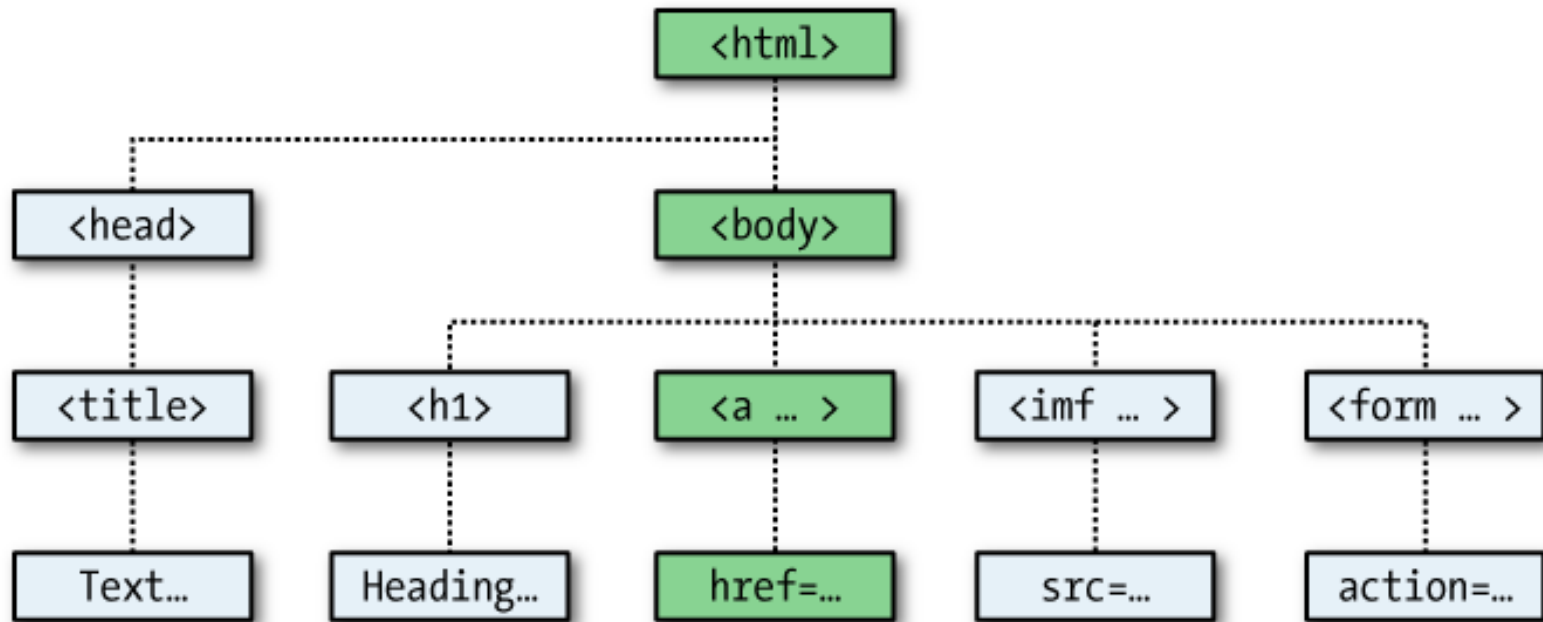
- **document.write()**
- **document.images[0]**
- **document.links.linkname.href**
- **document.getElementById("id")**

Reference, Tutorial and Examples

<http://www.w3schools.com/jsref/default.asp>

See Lectures: Week 8 for more resources.

Document Object Model



url = document.links.linkname.href

Access HREF using JavaScript and DOM

```
<html>
  <head><title>Link Test</title></head>
  <body>
    <a id="mylink" href="http://comminfo.rutgers.edu">SCI</a>
    <br />
    <script>
      url = document.links.mylink.href ;           // variable //
      document.write('The URL is ' + url);         // access DOM //
    </script>
  </body>
</html>
```

<http://comminfo.rutgers.edu/~aspoerri/Teaching/InfoTech/Lectures/Lec7/Steps/linkJavaScript.html>

JavaScript – Detect Visitor's Browser

```
<body>  
  <script type="text/javascript">  
    document.write(navigator.appName);  
    document.write("<br />");  
    document.write(navigator.appVersion);  
    document.write("<br />");  
    document.write(navigator.userAgent);  
    document.write("<br />");  
  </script>  
</body>
```

<http://comminfo.rutgers.edu/~aspoerri/Teaching/InfoTech/Lectures/Lec7/Steps/BrowserDetect.html>

More Info: <http://www.javascriptkit.com/javatutors/navigator.shtml>

Image Rollover JavaScript

```
<head>
  <script type="text/javascript" language="JavaScript">
    littlecat = new Image(143,83)
    littlecat.src = "real.jpg"
    bigcat = new Image(143,83)
    bigcat.src = "dream.jpg"
  </script>
</head>
<body>
  
</body>
```

<http://comminfo.rutgers.edu/~aspoerri/Teaching/InfoTech/Lectures/Lec7/Steps/imagerollover.html>