

Browser Sensing Basics: Navigator Object

- **Navigator** object provides basic information about browser such as name and version
- Much of this information comes via the user agent string which is transmitted during an HTTP request

Browser Sensing Basics:

Navigator Object

- **Common properties**

- `appCodeName` [Mozilla]
- `appMinorVersion` [;SP1;] *IE only*
- `appName` [Microsoft Internet Explorer]
- `appVersion` [5.0 (Windows;en-US)]
- `userAgent` [Mozilla/5.0 (Windows)]
- `vendor` [Netscape6] *NS6
> only*
- `vendorSub` [6.01] *NS6
> only*

Browser Detection: Example 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Browser Detect Example</title>
</head>
<body>
<script language="JavaScript" type="text/javascript">
<!--
var browserName = navigator.appName;
var browserVersion = parseFloat(navigator.appVersion);
document.write("Your browser is ", browserName, " ", browserVersion,
".");
// -->
</script>
<noscript>
    Sorry, I can't detect your browser without JavaScript.
</noscript>
</body>
</html>
```

Browser Detection: Example 2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
  Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Browser Detect Example 2</title></head>
<body>
<h1>Something Special for Your Browser</h1><hr>
<script language="JavaScript" type="text/javascript">
<!--
var browserName = navigator.appName;

if (browserName == "Microsoft Internet Explorer")
  document.write("<marquee>Some IE specific
markup!</marquee>");
else if (browserName == "Netscape")
  document.write("<blink>Netscape specific code!</blink>");
else
  document.write("<b>Browser Not Known: Just a bold
element!</b>");
// --></script>
<hr></body></html>
```

What to Detect

- Useful info detect comes in four categories
 1. Technical issues
 2. Visual issues
 3. Delivery issues
 4. User issues
- Technology detection
 - Markup, CSS, scripting, Java, objects and plug-ins

Technology Detection

- HTML and CSS is a little hard to detect for
 - DOCTYPE switch
 - Quirk mode vs. standards mode
 - CSS is possible using a measurement idea
- JavaScript detection
 - If script runs then it is on!
 - Using the **language** attribute on **<script>** to determine version
 - **if** statements for object detection

JavaScript Detection

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
  Transitional//EN"  
    "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
<title>JS Check</title>  
<noscript>  
<meta http-equiv="Refresh" CONTENT="0; URL=noscript.htm">  
</noscript>  
</head>  
<body>  
<script language="JavaScript" type="text/javascript">  
<!--  
  document.write("This page has JavaScript!");  
  // -->  
</script>  
</body>  
</html>
```

JavaScript Detection

- ```
<script language="JavaScript" type="text/javascript">
<!--
 window.location="scripton.htm";
 // -->
</script>
```
- ```
<script language="JavaScript" type="text/javascript">  
  // JS 1.0 features  
</script>
```

```
<script language="JavaScript1.1" type="text/javascript">  
  // JS 1.1 features  
</script>
```

```
<script language="JavaScript1.2" type="text/javascript">  
  // JS 1.2 features  
</script>
```


Object Detection

```
<script language="JavaScript" type="text/javascript">  
  if (document.images)  
    alert("Rollovers would probably work");  
  else  
    alert("Sorry no rollovers");  
</script>
```

- Careful!

```
var ie = (document.all) ? true : false;
```

- Better

```
var allObject = (document.all) ? true : false;  
var getByld = (document.getElementById) ? true : false;
```

More Technology Detection

- Java Detection

```
if (navigator.javaEnabled())  
    // do Java stuff or write out <applet> tag  
else  
    alert("Sorry no Java");
```

- Plug-in Detection

```
if (navigator.plugins["Shockwave Flash"])  
    alert("You have Flash!");  
else  
    alert("Sorry no Flash");
```

Visual Detection

- Starting with 4.x generation browsers you have **Screen** object for detecting visual characteristics of the accessing browser
- ```
<script language="JavaScript1.2" type="text/javascript">
<!--
if (window.screen)
{
 document.write("Height: "+screen.height+"
");
 document.write("Width:"+screen.width+"
");
 document.write("Available Height: "+screen.availHeight+"
");
 document.write("Available Width: "+screen.availWidth+"
");
 document.write("Color Depth: "+screen.colorDepth+"bit
");
}
else
 document.write("No Screen object support");
// -->
</script>
```

# Available Region Checking Example

```
<h2>Set to a new size then reload page</h2><hr>
<script language="JavaScript1.2" type="text/javascript">
 <!--
 var winWidth = 0;
 var winHeight = 0;

 if (window.innerWidth)
 winWidth = window.innerWidth;
 else if ((document.body) && (document.body.clientWidth))
 winWidth = document.body.clientWidth;

 if (window.innerHeight)
 winHeight = window.innerHeight;
 else if ((document.body) && (document.body.clientHeight))
 winHeight = document.body.clientHeight;

 document.write("Available height = "+winHeight+"
");
 document.write("Available width = "+winWidth+"
");
 //-->
</script>
```

# Using Screen Detection

- It would be nice to be able to resize fonts and other layout components based upon screen size
- Some browsers might be able to be more dynamic in resizing
  - Try this in IE 5
  - `<h1 style="font-family: verdana; text-align: center; font-size: expression(document.body.clientWidth / 13)">Internet Explorer Font Sizing!</h1>`

# More on Screen Detection

- We could use if statements to redirect to certain layout pages or insert hi or low color images

- ```
<script language="JavaScript" type="text/javascript">
<!--
  if (window.screen)
  {
    // Sense the bit depth...
    if (screen.colorDepth > 8)
      document.writeln('');
    else
      document.writeln('');
  }
  else
    document.writeln('');
// -->
</script>
<!-- Deal with the script off or non-JS aware browsers -->
<noscript>  </noscript>
```

Language Detection

- We can detect the browser's default language

```
var lang = "en-us";  
if (window.navigator.language)  
    lang = window.navigator.language  
else if (window.navigator.userLanguage)  
    lang = window.navigator.userLanguage  
if (lang == "es")  
    document.write("Hola amigo!");  
else  
    document.write("Hi friend!");
```

Advanced Detection

- Server-side detection
 - Issue to keep track of database of user agents
 - BrowserHawk
- Client capabilities in IE
- You can actually detect for lots of things including cursor position if you want to get really tricky!

Browser Control

- **Location** object
 - Child of the **Window** object
 - Allows you to read and set the current browser page location
 - `window.location = 'http://www.pint.com';`
 - You can pull the various pieces out of the Location string URL such as hostname, protocol, etc.
 - `alert(window.location.hostname);`

Browser Control Contd.

- **History** Object
 - Read-only array of URL strings showing where the user has been
 - Movement is allowed with **History** but examination of the history would be a security violation
 - Examples using **History** movement methods

```
<a href="javascript: window.history.forward()">Forward</a>  
  <a href="javascript: window.history.back()">Back</a>  
<a href="javascript: window.history.go(-2)">Back two times</a>  
  <a href="javascript: window.history.go(3)">Forward 3 times</a>  
<a href="javascript: window.history.go(window.history.length-1)) ">Last  
  Item</a>
```

Status Bar

- You can set the status bar at the bottom of the browser window using `window.status` and `window.defaultStatus`
- In instances with HTML event handlers you need to return true to cause the status to change
- ```
<a href="http://www.yahoo.com"
 onmouseover="window.status='Don\'t Leave Me!'; return true;"
 onmouseout="window.status=''; return true">
Go to Yahoo!
```
- ```
<script language="JavaScript" type="text/javascript">
<!--
defaultStatus='JavaScript is fun!';
//-->
</script>
```

Timeouts and Intervals

- Window object supports methods for setting timers
 - `setTimeout()` and `clearTimeout()`
 - `timerId = setTimeout(script-to-execute, milliseconds);`
- Timeouts happen once you can also use intervals which occur over and over until stopped
 - Similar syntax `setInterval()` and `clearInterval()`

Timeout Example

- ```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>5,4,3,2,1...BOOM</title></head>
<body>
<h1 align="center">Browser Self-Destruct</h1>
<hr><div align="center">
<form>
 <input type="button"
 value="Start Auto-destruct"
 onclick="timer = setTimeout('window.close()', 5000);
 alert('Destruction in 5 seconds'); return true">
 <input type="button"
 value="Stop Auto-destruct"
 onclick="clearTimeout(timer); alert('Aborted!'); return true">
</form>
</div>
</body>
</html>
```

# Applied Timeouts

- Common uses of timeouts include
  - Scrollers
  - Timed advertisements
  - Auto-run slide shows
  - Animation
- Besides controlling what a browser does a certain time you can even simulate button presses and in some cases “slam” the user’s preferences

# Button Press Example

- ```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head><title>Browser Button Simulator</title></head>
<body>
<h1 align="center">Button Simulator</h1><hr>
<form>
<input type="button" value="PRINT" onclick="if (window.print) window.print()">
<br><br>
<input type="button" value="FORWARD" onclick="if (window.forward)
window.forward()">
<br><br>
<input type="button" value="BACK" onclick="if (window.back) window.back()">
<br><br>
<input type="button" value="HOME" onclick="if (window.home) window.home()">
<br><br>
<input type="button" value="STOP" onclick="if (window.stop) window.stop()">
</form>
</body>
</html>
```

The Dark Side

- Of course this leads to folks pursuing the dark corners of JavaScript including bugs to do things the user may not want
- Home Page Slamming Example
 - `<a href="#"
 onclick="HomePage = 'http://www.mm214.com';
 this.style.behavior='url(#default#homepage)';this.setH
 omePage(HomePage);return false">Set mm214 to your
 home page`