

# Web Browsers & Compatibility Problems

## Web Browsers

If you are developing a web application using a client-side scripting language then you will need to ensure cross compatibility amongst browser. A browser is where the script will run. As browser are constantly being updated , so too are the version of scripting language that is run inside them.

Common scripting languages that run inside browsers are **Javascript** and **Vbscript**.

- Javascript is available across a variety of browsers.
- VBscript is developed by Microsoft and is available only on the Internet Explorer browsers.

The choice in browser you are developing for is entirely dependent on the choice of language you will be developing your application in. The Javascript language syntax is very similar to Java, however they not the same.

Therefore if you already know Java then you can easily learn Javascript. The VBScript language is very similar to Visual Basic, and therefore if you already know Visual Basic then VBScript is easy to learn.

## Differences between Client-side and Server-side Scripting

### Client-side Environment

The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the users computer over the internet and run directly in the browser.

The scripting language needs to be **enabled** on the client computer. Sometimes if a user is conscious of **security risks** they may switch the scripting facility off. When this is the case a message usually pops up to alert the user when script is attempting to run.

### Server-side Environment

The **server-side environment** that runs a scripting language is a web server. A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser. It is usually used to provide interactive web sites that interface to databases or other data stores on the server.

This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores.

## Compatibility

Web designers spend a lot of time trying to make their sites work in all the different browsers. Typically the 3 major browsers used are Firefox, IE6 and IE7. Perhaps you thought they would behave the same when rendering CSS in the browser window? **If only life were that simple.**

Although Firefox and IE7 are more similar than ever in rendering CSS, IE6 has always had a mind of its own, adding padding and spacing amongst other things in random locations. However, there is a simple way to cater specifically for the major browsers. **IE6** recognises **underscore lines**, but **IE7 does not**. **IE6 and IE7 recognise period lines**.

For example...

```
#header
{
margin-top: 8px;
.margin-top: 10px;
._margin-top: 4px;
}
#header
{
margin-top: 8px;
.margin-top: 10px;
._margin-top: 4px;
}
```

**Firefox**, and every other **non-IE browser**, will only see and use the **8px**, **IE6** will see the **10px line**, but then the **4px line will stomp on it**, and **only IE7** will see the **10px line**. You can now support all major browsers without having to write a serverside script, clientside script or even a horrible CSS expression :)

Three problems

- 1) we must recognize all browsers have **bugs**, although some more than others. In fact, it's likely you will stumble upon at least one such bug for every site you design.
- 2) **IE** does not fully implement the CSS 2.1 specifications, so it will not understand fancy (but very useful) CSS like:
  - li:hover – might be used for a drop-down menu
  - input:focus – might be used to change the color of a text input when the user is typing in it
  - tr > td:first-child – might be used to change the color of the first column of a table
- 3) **IE** uses a different box model than all the other browsers. The padding of an element is not added to the width, but is instead part of it. It also does other nasty things. For instance, it won't center a division if you do this `div.your_class {margin:0 auto;}` (every other browser does).

You can design the most beautiful CSS compliant web, all validated to W3C standards, only to find it's almost completely unreadable or unusable in another browser. Font sizes are microscopic, menus don't appear as they may now be hidden under a layer, and the layout is all wrong. Worse, it just doesn't look as if it's been professionally designed, and if you have already published it, all you can do is cringe before rolling up your sleeves and correcting it.

Browsers have different defaults. Internet Explorer seems to have more than others, so if you design and test using only IE your pages may not display properly in other browsers. As a rule you should test your site with other browsers before publishing it.

## **Think About It**

The best way to minimize these problems is to pay attention to browser compatibility when building your Web page. Avoid using HTML extensions and be careful about using cutting-edge features of the language that may not yet be supported by all the major browsers.

### **Different Browser Versions**

The major difference between two versions of the same browser is their support for newer portions of the HTML language. A new browser is generally better at displaying Web pages than an old one.

### **Different Computer Types**

The Macintosh is still used by 12% of computer users, and has a very loyal following among graphic designers and publishers. In theory, if you view your page on both a PC and a Mac using the same version of the same browser, it should display the same, right?

### **Different Screen Sizes**

If you don't test your pages using different screen resolutions, your page may be stretched to fit a large screen, or be cropped to fit a small screen.

### **Different Font Sizes**

Most browsers allow users to customize their default font size. Many users who work on computers all day do this to reduce eye strain. As a result, user preferences may cause the typeface that you used to design your Web page to increase as much as 50% larger in a user's browser. This increase in font size can hurt many carefully-planned page designs.

### **HTML Errors**

Whether you're an HTML coder who builds Web pages by working with the raw HTML tags, or a designer using a WYSIWYG editor like FrontPage or Dreamweaver, the odds are your finished Web page will contain HTML errors.

### **Browser Bugs**

As you've probably seen by now, building a Web page that displays well on all browsers isn't easy. To make matters worse, sometimes you've done everything right, and your page still doesn't display correctly under one specific browser. It's not your fault - you've just encountered a browser bug.

## **What You Can Do**

If only it were easy to cure browser display problems, but fixing them takes time. Read the following steps to learn what you can do to make it easier.

**The Web is hip**; it's hot and exciting; it's radical. So many Web Designers feel they have to build cutting-edge features into their Web page. That's a **bad idea**, because cutting-edge features are rife with browser compatibility problems, not to mention the impact they have on your page load time.

**HTML errors** are the leading cause of browser display problems. Making sure your Web pages are error free is one of the most important steps you can take to solve browser display problems.

Next to HTML errors, **compatibility problems** are the leading cause of browser display errors. We've already warned you about including cutting-edge features in your site, but compatibility dangers extend to all aspects of HTML.

### **Why do you care about Browser Compatibility?**

Web sites reflect the company's professional image. If your site renders improperly or not at all, your company's reputation can be tarnished. If your site has browser display problems, visitors and potential customers will leave your site and not look back. In contrast, a professional looking site will make visitors feel more comfortable, stay longer and browse more pages. And because of this increased credibility, they are more likely to purchase the products and services that they are looking for from you.

### **Exercise**

---

- 1) Name two scripting languages used in web browsers
- 2) name the two main browsers in use today.
- 3) identify 3 compatibility problems between browsers
- 4) Identify three methods which could be used to minimize these problems.