CSS SPRITES vs JAVA SCRIPT ROLLOVER

an alternate event for rollovers

An article by Sven Lennartz, **Smashing Magazine**, April 27th, 2009
http://coding.smashingmagazine.com/2009/04/27/the-mystery-of-css-sprites-techniques-tools-and-tutorials/

What Are CSS Sprites?

- The term “sprite” (similar to “spirit,” “goblin,” or “elf”) has its origins in computer graphics, in which it described a graphic object blended with a 2-D or 3-D scene through graphics hardware.

- Because the complexity of video games has continually increased, there was a need for smart techniques that could deal with detailed graphic objects while keeping game-play flowing.

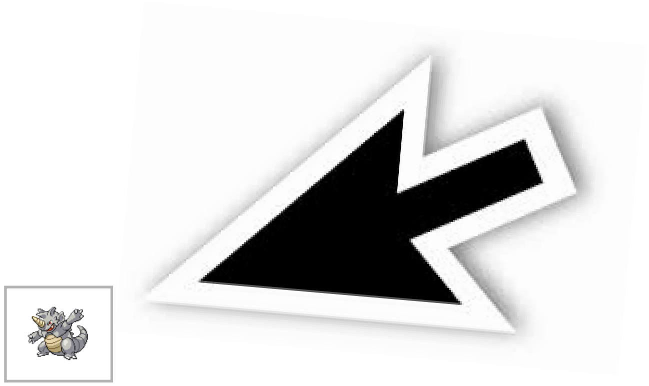
What Are CSS Sprites?

- One of the techniques developed saw sprites being plugged into a master grid, then later pulled out as needed by code that mapped the position of each individual graphic and selectively painted it on the screen.

http://coding.smashingmagazine.com/2009/04/27/the-mystery-of-css-sprites-techniques-tools-and-tutorials/

What Are CSS Sprites?

- Sprites were displayed over a static or dynamic background image, and the positioning of the sprite was controlled simply by the hardware controllers. The term was coined because the sprites seemed to "haunt" the display and didn't really exist in the graphic memory.



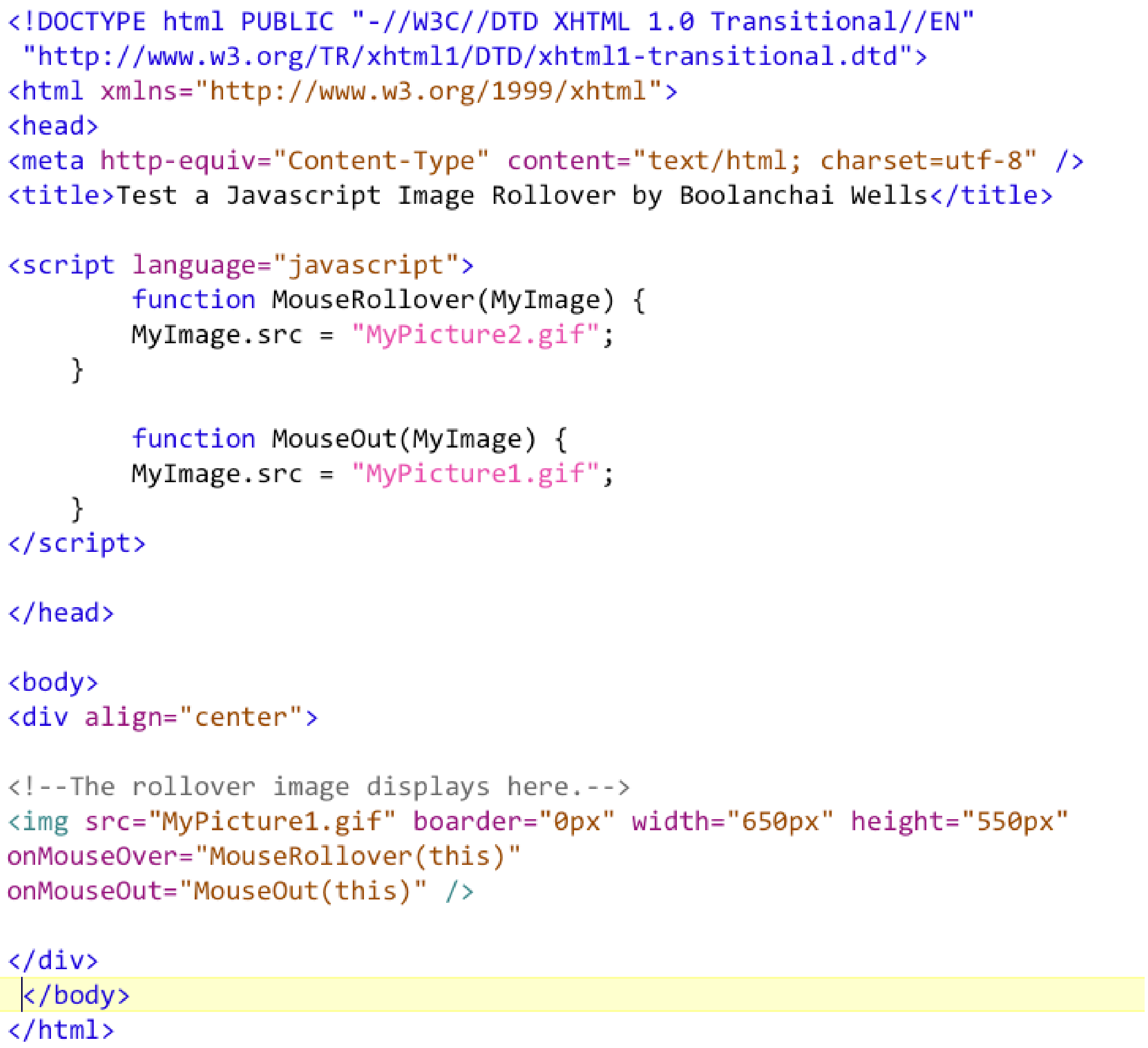
JS Rollover (replaces image)

- Time passed, and at the beginning of the 2000s, when progressive Web designers started to seek alternatives to JavaScript-based rollover menus (with onMouseOver and onMouseOut effects), sprites saw a renaissance in Web development. With CSS, the simple implementation of sprites

```
<script language=”javascript”>
       function MouseRollover(MyImage) {
       MyImage.src = “MyPicture2.jpg”;
    }
       function MouseOut(MyImage) {
       MyImage.src = “MyPicture1.jpg”;
    }
</script>
```

http://www.wikihow.com/Make-a-JavaScript-Image-Rollover

JS Rollover (replaces image)

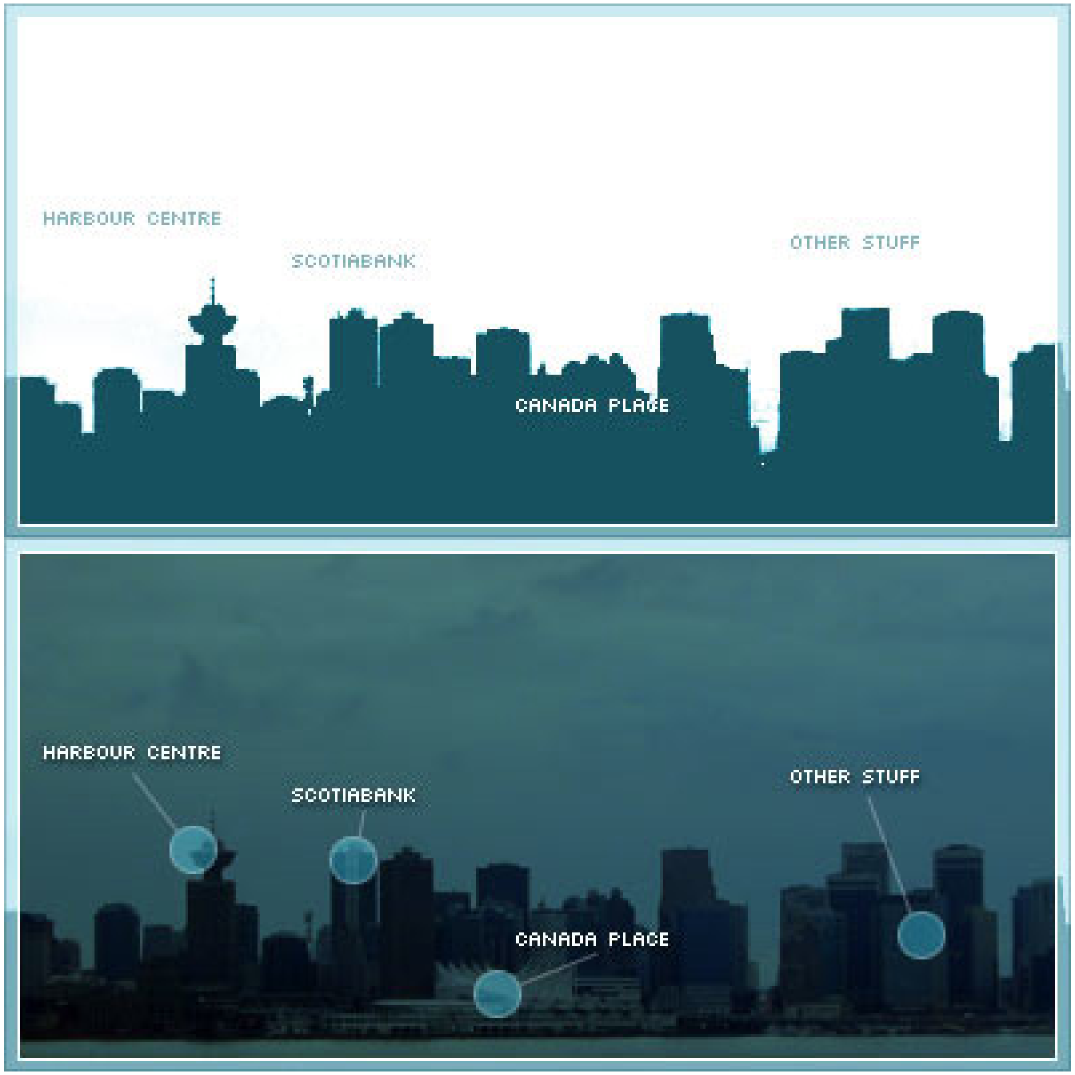


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Test a Javascript Image Rollover by Boolanchai Wells</title>

<script language="javascript">
        function MouseRollover(MyImage) {
        MyImage.src = "MyPicture2.gif";
    }

        function MouseOut(MyImage) {
        MyImage.src = "MyPicture1.gif";
    }
</script>

</head>

<body>
<div align="center">

<!--The rollover image displays here.-->
<img src="MyPicture1.gif" boarder="0px" width="650px" height="550px"
onMouseOver="MouseRollover(this)"
onMouseOut="MouseOut(this)" />

</div>
 </body>
</html>
```

http://www.1stwebdesigner.com/inspiration/30-menu-creative-javascript-navigation-menu-examples/

David Shea

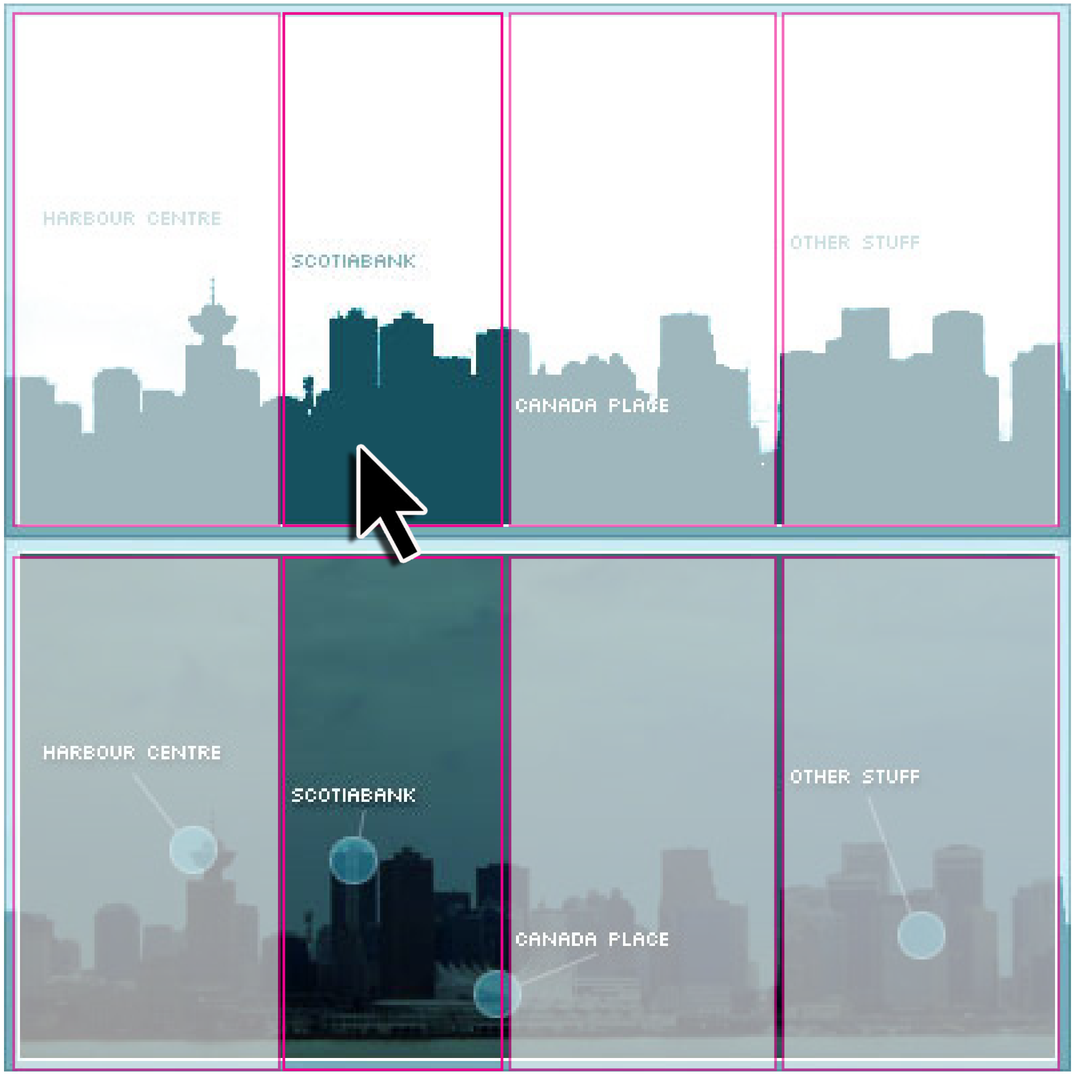
- In 2004, Dave Shea suggested a simple CSS-based approach to CSS sprites based on the practice established by those legendary video games.
- In this case, multiple images used throughout a website would be combined into the so-called "master image."



http://alistapart.com/article/sprites

David Shea

- To display a single image from the master image, one would use the background-position property in CSS, defining the exact position of the image to be displayed.

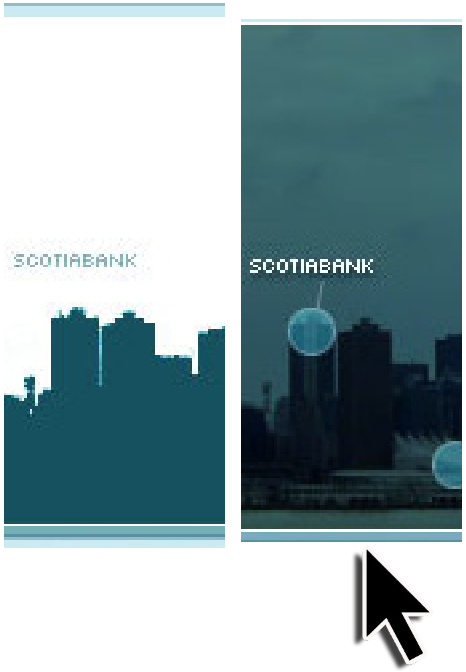


http://alistapart.com/article/sprites

David Shea

- Any hover, active or focus effects would be implemented using the simple definition of the background-position property for the displayed element was possible

- It was now much easier and clearer than its JavaScript-based predecessor.



http://alistapart.com/article/sprites

JS mouseOver vs CSS sprite

- When the page is loaded, it would not load single images one by one (nor hover-state images per request), but would rather load the whole master image at once.
- It may not sound like a significant improvement, but it actually was: the main disadvantage of the onMouse effects is that JavaScript-based hover effects require two HTTP requests for each image, which takes time and creates that unpleasant "flickering" of images.
- Because the master image is loaded with the whole page only once with CSS sprites, no additional HTTP requests are needed for hover, active or focus effects (because the image is already loaded), and no "flickering" effect occurs.

http://alistapart.com/article/sprites

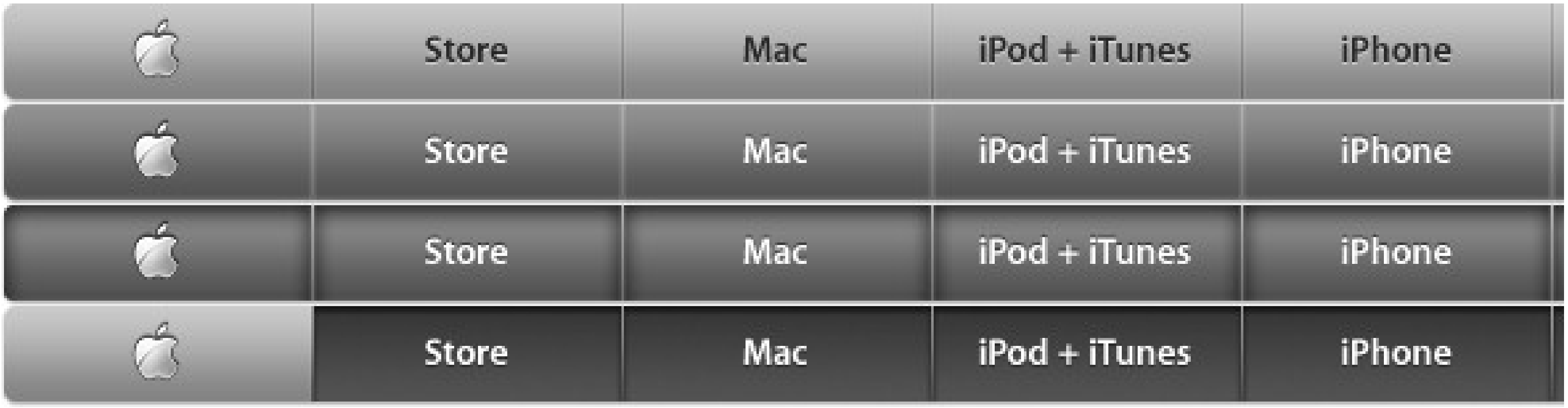
Consequence

- CSS sprites reduce HTTP requests and the loading time of pages.
- This is the main reason why CSS sprites are often used on websites with heavy traffic, where millions of page impressions would need “only” a tiny fraction of what could otherwise be 30,000,000.
- Hence, CSS sprites are commonly used, particularly for navigation (such as for hover effects), icons and buttons.

http://alistapart.com/article/sprites

Where Are CSS Sprites Used?

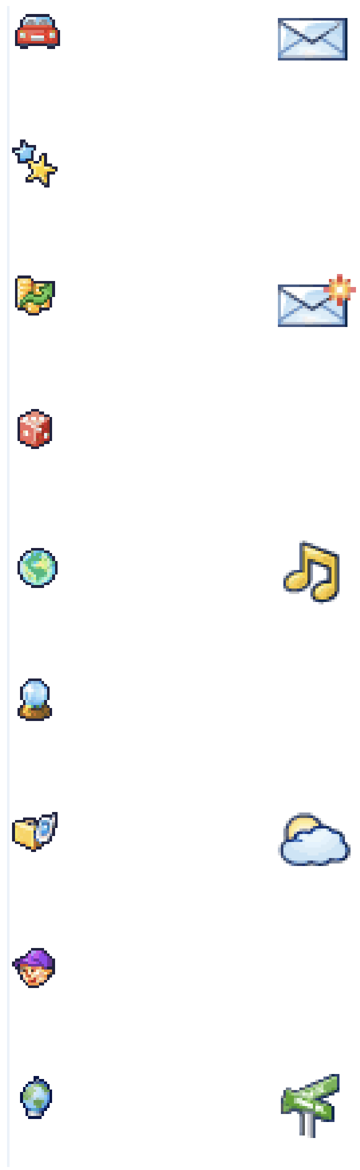
- CSS sprites can be used in various settings.
- Large websites can combine multiple single images in a meaningful manner, creating clearly separated “chunks” of the master images – the purpose being to keep the design maintainable and easy to update.



http://alistapart.com/article/sprites

Where Are CSS Sprites Used?

- The large empty space between the images is often used to make sure that the text resizing in browser doesn't cause side effects such the display of multiple images in the background. In fact, sprites usually work well in a pixel-based design, but they are hard to use in elastic (em-based) designs due to the restricted background-position-property.



http://alistapart.com/article/sprites

Where Are CSS Sprites Used?.

- Essentially, the structure that sprites take depends on the trade-off between maintainability and reduced server load; thus, it varies depending on the project you are working on.

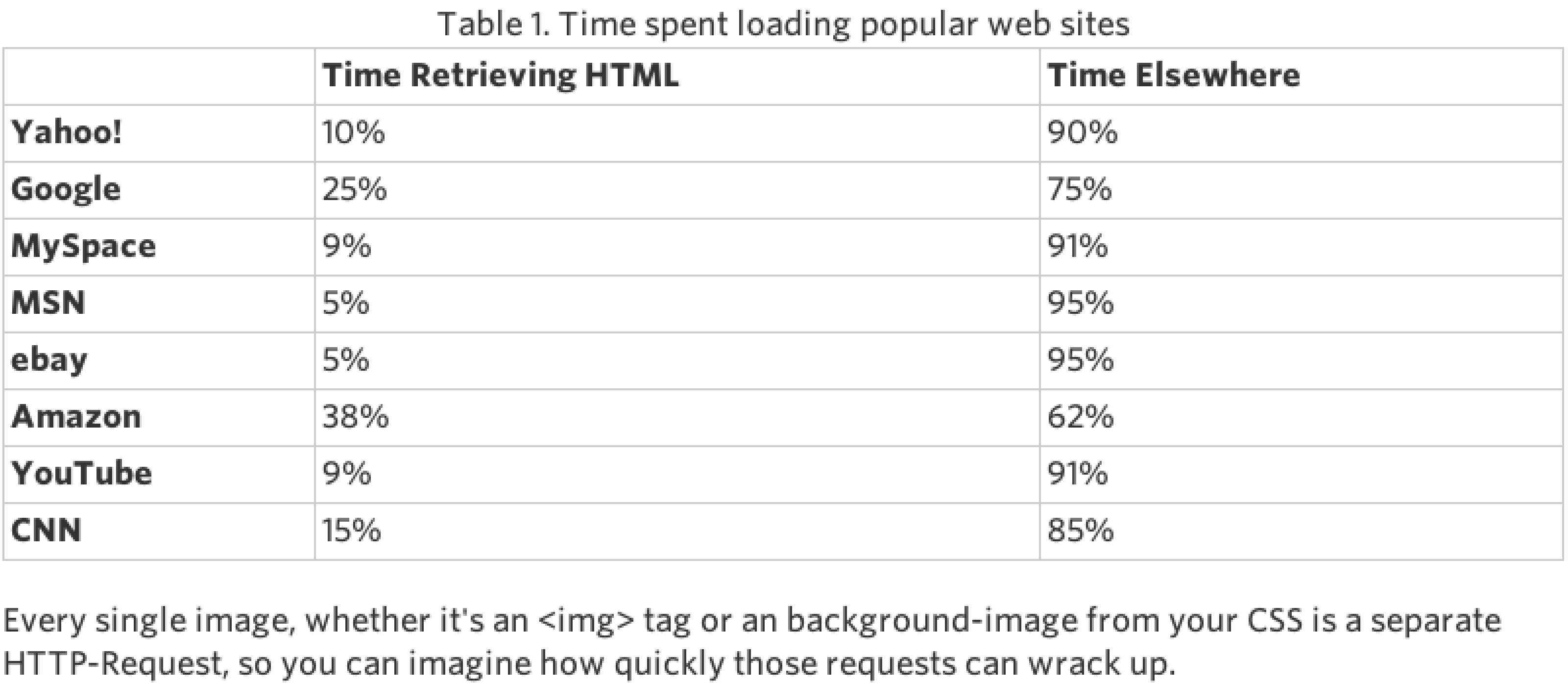


Table 1. Time spent loading popular web sites

	Time Retrieving HTML	Time Elsewhere
Yahoo!	10%	90%
Google	25%	75%
MySpace	9%	91%
MSN	5%	95%
ebay	5%	95%
Amazon	38%	62%
YouTube	9%	91%
CNN	15%	85%

Every single image, whether it's an <img> tag or an background-image from your CSS is a separate HTTP-Request, so you can imagine how quickly those requests can wrack up.

http://css-tricks.com/css-sprites/