CSS3

INTRODUCTION

Levels of CSS

CSS Level 1 (CSS1) became a W3C Recommendation on 17 December 1996. It was revised on 11 January 1999.

CSS Level 2 (CSS2) became a W3C Recommendation on 12 May 1998. It was revised on 11 April 2008.

CSS Level 2 Revision 1 (CSS2.1) became a W3C Recommendation on 7 June 2011.

CSS3 is the **latest version** of the Cascading Style Sheet Languages.

Instead of writing CSS3 as a single specification, the W3C decided to **divide CSS3 into a series of individual specifications - called "modules"**.

This process allows the W3C to bring modules through the five step advancement process **at different rates** - rather than wait till all the individual modules are defined.

There are a wide range of **CSS3 modules** such as:

Media Queries
Selectors Level 3
CSS Color Module Level 3

CSS Current work

http://www.w3.org/Style/CSS/current-work

CSS3 Module Status

http://www.css3.info/modules/

Despite some of the confusing titles given to various CSS3 modules, **there is no such things as CSS4**!

“Some of our modules start out at level 3, if they extend something from CSS2.1. Others start out at level 1, if they're something new (for example, Flexbox). However, the level that a module is at has no correlation with what version of CSS it's in. They're all CSS3 (or just CSS), regardless of what level they're at.”

http://www.xanthir.com/b4Ko0

Why use CSS3?

CSS3 offers a range of **cool effects** such as gradient, border-radius, text-shadow, box-shadow, transforms and animations.

CSS3 helps **reduce the number of decorative images** we have to use, meaning faster builds, faster page loads, easier maintenance.

CSS3 allows us to use **web fonts**, making our web pages more flexible and exciting.

CSS3 allows us to use **media queries** - enabling us to build sites that work across all devices.

Using CSS3 now

Browser support

Most modern browsers **support a wide range of CSS3 features**. However, before deciding whether to use a CSS3 feature, it is worth checking browser support.

There are many websites that provide information about **CSS3 and browser support**.

When can I use…

http://caniuse.com/

Browser Support for CSS3

http://www.impressivewebs.com/css3-browser-support/

CSS browser compatibility

http://www.quirksmode.org/css/contents.html

HTML5 & CSS3 Support

http://www.findmebyip.com/litmus/

But what if the CSS3 feature you want to use is **not supported by one of your target browsers**? There are two main options:

1: Progressive enhancement
2: Polyfill

Using progressive enhancement

Progressive enhancement is about creating web pages that meet two criteria:

1. Provide the **basic content and functionality** of web pages to everyone, regardless of their device or bandwidth.

2. Provide an **enhanced version** of web pages for those with more advanced devices or better bandwidth.

A simple example might be a **box with rounded corners** - which we can create using border-radius.

```
.test { border-radius: 10px; }
```

The box will have have rounded corners in all modern browsers, **but not in IE6, IE7 or IE8** as these browsers do not support the border-radius property.

Progressive enhancement means that **the box will look acceptable in older browsers**, and slightly better in modern browsers.

Modernizr is a JavaScript Library. It does not force older browsers to support CSS. Instead, it adds classes to the HTML element based on each browser's support for various features.

http://www.modernizr.com/

```
<html class="js canvas canvastext no-
geolocation rgba hsla multiplebgs borderimage
borderradius boxshadow opacity cssanimations
csscolumns cssgradients cssreflections
csstransforms csstransforms3d csstransitions
video audio localstorage sessionstorage
webworkers applicationcache fontface">
```

This allows us to create styles for specific CSS3 features, and then **fallback styles for browsers that do not support those feature**.

```css
.csscolumns
{
    -moz-column-count: 2;
    -webkit-columns: 2;
    -o-columns: 2;
    columns: 2;
}

.no-csscolumns
{
    float: left;
    margin: 0 0 20px;
}
```

Modernizr is an **excellent tool to help achieve** progressive enhancement.

Using Polyfills

Polyfills are JavaScript libraries that force specific browsers to support various CSS3 features.

CSS3 polyfills allow us to use features like **CSS3 Selectors and CSS3 Vendor Prefixes** safely across a wider range of browsers.

Selectivizr

http://selectivizr.com/

Prefix-free

http://leaverou.github.com/prefixfree/

Respond.js

https://github.com/scottjehl/Respond

Flexie

http://flexiejs.com/

CSS Sandpaper

https://github.com/zoltan-dulac/cssSandpaper

CSS3 PIE

http://css3pie.com/

Vendor extensions

Vendors (browser makers) often release **CSS features** that are still in development. This allowing vendors to test CSS features before they become a formal standard.

Each vendor uses it’s own **unique identifier** associated with CSS properties or values.

Vendor prefixes

Vendor	Prefix
-webkit-	WebKit-based browsers
-moz-	Mozilla-based browsers
-ms-	Microsoft
-o-	Opera

In theory, browsers **should ignore** vendor-extensions that do not relate to their prefixes.

However, browsers such as Opera and Firefox **now support Webkit prefixes** as well as their own prefixes.

Where are they placed?

Some vendor extensions are placed **before properties**.

```
div
{
    -webkit-appearance: ?;
    -moz-appearance: ?;
}
```

Other vendor extensions are placed **before values**.

```
div
{
    background-image:
-webkit-linear-gradient(left, yellow, black);
}
```

In some rare cases, the vendor extensions appear **before both properties and values**.

```
div
{
    -webkit-transition: -webkit-transform 1s;
}
```

Before or after?

Vendor specific properties should always be **placed before** the correct CSS3 property or value.

```css
div
{
    -webkit-border-radius: 10px;
    -moz-border-radius: 10px;
    -ms-border-radius: 10px;
    -o-border-radius: 10px;
    border-radius: 10px;
}
```

Where possible, include a non-CSS **fallback option**.

```
div
{
    background: rgb(100, 66, 255);
    background: rgba(100, 66, 255, 0.5);
}
```

Russ Weakley

Max Design

Site: maxdesign.com.au

Twitter: twitter.com/russmaxdesign

Slideshare: slideshare.net/maxdesign

Linkedin: linkedin.com/in/russweakley