CS7026 – CSS3

Introduction to CSS3
What is CSS3?

- CSS3 is an extension of CSS 2.1 that adds powerful new functionality,

- It's no longer a single specification. Instead, it's been divided up into several modules.

- Each module is a standalone specification for a subsection of CSS, like selectors, text, or backgrounds.
What is CSS3?

- Some of these modules include:
  - The Box Model
  - Lists Module
  - Hyperlink Presentation
  - Speech Module
  - Backgrounds and Borders
  - Text Effects
  - Multi-Column Layout

- You can see a list of all the modules, plus their current status at [www.w3.org/Style/CSS/current-work](http://www.w3.org/Style/CSS/current-work).
What is CSS3?

- Every module has its own set of authors and its own timetable.

- The advantage of this is that the entire CSS3 specification doesn’t have to be held up waiting for one little bit to get worked out - the module that that little bit is in can wait, while the rest moves forward.
Overview of What’s New

- Much of CSS3 is a repeat of CSS 2.1, of course. But there are many additions and revisions.

- We will be having a look at some of the best-supported, popular, and useful changes to CSS from level 2.1 to level 3.
Overview of What’s New

- Image-free visual effects.

- There are a lot of new properties that allow you to create visual effects that previously could be accomplished only with images (or sometimes scripting), such as rounded corners, drop shadows, semitransparent backgrounds, gradients, and images for borders.

- Many of these new properties are in the Backgrounds and Borders module; others are in the Colors and Image Values modules.
Overview of What’s New

- Box transformations.

- CSS3 makes possible the manipulation of the box’s position and shape in two- or three-dimensional space, such as rotating, scaling, or skewing it.

- These effects are called ‘transforms’, and are covered in the **2D Transforms** and **3D Transforms** modules.
Overview of What’s New

- Unique fonts.
  
  The **Fonts** module introduces the `@font-face` rule that allows you to link to a font file on your server and use it to display the text on your page, instead of being limited to the fonts on your users’ machines.

  This makes beautiful typography so much more attainable.
Overview of What’s New

- **Powerful selectors.**

  - **CSS3** introduces over a dozen new selectors, mostly pseudo-classes and attribute selectors.
  
  - They allow you to target specific pieces of your HTML without needing to add IDs or classes, streamlining your code and making it more error-proof.
  
  - These selectors are included in the **Selectors** module.
Overview of What’s New

- Transitions and animations.

  - CSS3 transitions (covered in a module of the same name) are a simple type of animation that allow you to ease the change from one style on an element to another,

  - E.g. gradually and smoothly changing the colour of a button when you hover over it.

  - Fully-fledged CSS3 animations (again covered in a module of the same name) can make more complicated style changes and movements possible without needing Flash or JavaScript.
Overview of What’s New

- **Media Queries.**

  - The *Media Queries* module introduces syntax for feeding styles based on the capabilities of the user’s display or device.

  - E.g. the viewport width, screen resolution, and how many colours it can display.

  - Media queries are a great tool for creating mobile-optimized web sites.
Overview of What’s New

- Multiple-column layouts.
  - CSS3 introduces a few new modules that make multi-column layouts easier to create.

  - The *Multicolumn Layout* module deals with flowing the text of a single block into multiple columns, similar to newspaper layout.

  - The *Flexible Box* Layout module deals with making blocks align horizontally or vertically with each other and making them more flexible to the available space than floats or positioning can be.

  - There are also more experimental layout modules called *Template Layout* and *Grid Positioning*. 
Where CSS3 Stands

- So just how soon is all this new CSS3 stuff going to be finalised so we can use it??

- Each module is on its own timetable, and you can see the status of each at [www.w3.org/Style/CSS/current-work](http://www.w3.org/Style/CSS/current-work).

- The table lists the status, usually called a maturity level but sometimes called a stability status by the W3C, of the current version of the module as well as the next version, with links to each document.
Where CSS3 Stands

- The levels the W3C uses are (from least mature to most mature):

  1. **Working Draft:** The first publicly available version of the specification is referred to as the First Public Working Draft (FPWD). It along with further Working Draft (WD) are published for review by the community, in order to solicit further changes. A module or specification can go through several working drafts.

  2. **Last Call:** A working draft with a deadline for final comments. It indicates the working group thinks the module does what it should - though it usually receives significant changes after this point - and is probably planning to advance it to the next level.
Where CSS3 Stands

3. **Candidate Recommendation:** The working group believes the module meets requirements, is stable, and should be implemented by browsers and put into everyday use by web developers, in order to see how implementable it is. Changes are still possible after this point, but not many and not major.

4. **Proposed Recommendation:** A mature, well-reviewed document that has been sent to the W3C Advisory Committee for final endorsement. There are rarely changes after this point.

5. **Recommendation:** Complete and finalized. Normally referred to as a “standard.”
Where CSS3 Stands

- It should be clear from this list that web developers are not only allowed to use W3C specifications long before they are complete and finalised Recommendations, but that we are expected to.

- In fact, if you look at the list on the W3C site you will notice that not very many of the level 3 modules, are at Recommendation status.

- Even though it is not a finalised standard, we can use much of CSS3 now.
Use CSS3 Now

Quite a few CSS3 modules are at Candidate Recommendation status, indicating they should be used, but it’s also fine to use some pieces that are still in Working Draft status.

You should use only those pieces of CSS3 that are fairly stable and won’t harm non-supporting browsers by their lack.

And you should use them wisely! Don’t add CSS3 just because you can - decide if it makes sense for the site’s goals and its users, and add it where appropriate.
Luckily, the most stable pieces of CSS3 that we’d want to use do have good browser support already. This is the summary table from caniuse.com.
Progressive Enhancement

- These varying levels of browser support mean that your pages will look different in different browsers. This is OK, and frankly unavoidable, whether you use CSS3 or not.

- Web pages have never been able to look identical everywhere because of the user-controlled nature of the medium.

- Today, there’s an even wider variety of devices, monitors, browsers, and settings that people use to browse the web, so you’re even less likely to create a page that looks identical everywhere.

- As long as you focus on making pages that are usable and accessible for everyone, the cosmetic differences shouldn’t matter much.
Progressive Enhancement

- Progressive enhancement is a method of developing web pages where you
  - first make them work and look decent in base-level browsers and devices, and
  - then layer on non-essential, more advanced CSS and JavaScript enhancements for current and future browsers with better support.

- E.g., you may build a form in plain, semantic HTML that looks fine without CSS available and works without JavaScript enabled, and then enhance its appearance with CSS and its usability with JavaScript client-side validation, adding to serverside validation you already have in place.
Progressive Enhancement

- The goal is to create the richest experience possible for everyone by taking advantage of the capabilities of modern browsers while still making sites that are completely usable for everyone.

- “Progressive enhancement…aims to deliver the best possible experience to the widest possible audience—whether your users are viewing your sites on an iPhone, a high-end desktop system, a Kindle, or hearing them on a screen-reader, their experience should be as fully featured and functional as possible.”

Designing with Progressive Enhancement (www.filamentgroup.com/dwpe)
Progressive Enhancement vs. Graceful Degradation (an aside)

- When you develop with a graceful degradation methodology, you build your site completely, with all of the features and enhancements you want to include.

- Then you add in fallbacks for browsers that can’t support the fully featured version, making the site degrade in a way that won’t break in older browsers.

- In progressive enhancement, you’re not reverse-engineering a completed site to work with older browsers.

- You start out with clean, semantic HTML, good content and responsive design which will work in all devices and browsers, and then layer on extra styling and features in an unobtrusive way that won’t harm the base-level browsers, and which will automatically work as browsers improve.
Progressive Enhancement

- So, get older browsers as far as you can, and then keep on improving and pushing the boundaries of the site for newer browsers. This way, everyone gets the best possible site.

- As time goes by and users upgrade and browser support improves, more of your visitors will see more of your enhancements, effectively making your site better over time without your doing a thing.
Benefits of CSS3

- CSS3 allows you to create some really beautiful effects, adding a layer of polish and richness to your web designs.

- But most of the visual effects that CSS3 allows you to create can be accomplished without CSS3, using alternative tools such as images, JavaScript, or Flash.

- So there needs to be some other reason beyond that “it looks cool” to use CSS3...
Benefits of CSS3

- Using CSS3, you can:
  - Decrease the time you spend developing and maintaining pages.
  - Decrease the time spent in loading those pages.
  - Increase usability and accessibility.
  - Make your pages more adaptable across devices.
  - Enhance your search engine placement.
Reduced Development & Maintenance Time

- Many CSS3 techniques can be a replacement for “called” images. This frees you from having to spend the time creating, slicing, and optimizing those images.

- You can also tweak CSS more quickly than images if you need to make changes down the road, or simply test out different variations.

- Some CSS3 techniques also allow you to do away with scripts or Flash.

- Many CSS3 techniques can streamline your markup by requiring fewer nested divs and classes, and this also can translate into a little less work time for you.
Increased Page Performance

- Less markup and fewer images means fewer kilobytes for users to download, resulting in faster-loading pages.

- Fewer images, scripts, and Flash files also mean fewer HTTP requests, which is one of the best ways to speed up your pages.

- However, some graphically rich CSS3 techniques, such as gradients, can reduce HTTP requests but may also make the browser processor work very hard to render the effects, making the browser sluggish and decreasing usability.

- Don’t overuse complex effects, and test thoroughly those that you do implement.
Better Search Engine Placement

- Fast pages are not only good for your users, but they make Google happy.

- In March 2010, Google started rewarding fast pages by making speed a ranking factor, so pages that load faster can appear a little higher in the search results than their slower competitors.

- You may still get a bit of a boost in search engine placement if you replace images of text or Flash files of text with real text styled with CSS3.

- While search engines can read text in images’ alt attributes and some Flash files, regular text in heading tags is usually going to be given more weight by the search engines.
Increased Usability and Accessibility

- An even bigger benefit of real text instead of images of text is that real text is more usable for everyone, and particularly for people with disabilities.

- Real text can be
  - resized or recoloured by users to make it easier to read,
  - selected to copy and paste,
  - searched for with the browser’s Find function,
  - indexed by search engines and other tools, and
  - translated into other languages.
Increased Usability and Accessibility

- Another way to improve usability with CSS3 is to use media queries. These let you customize styles based on the characteristics of the user’s display, allowing you to tailor styles to the user’s device and settings.

- This technique can ensure your design is making the best use of space and is as readable as possible for the user’s browsing scenario.
Using CSS3 Wisely - Browser Prefixes

- When a browser implements a new property, value, or selector that is not yet at Candidate Recommendation status, it adds a prefix onto the front of the property with a code for its rendering engine.

- E.g., `-moz-border-radius` is the version of the `border-radius` property currently used by Mozilla-based browsers, like Firefox.
Using CSS3 Wisely - Browser Prefixes

Browser vendors sometimes add prefixes to experimental or nonstandard CSS properties, so developers can experiment but changes in browser behaviour don't break the code during the standards process.

Browser-specific prefixes for CSS Properties:

- `webkit` - (Chrome, newer versions of Opera, almost all iOS browsers (including Firefox for iOS); basically any WebKit based browser))
- `moz` - (Firefox)
- `o` - (Old versions of Opera)
- `ms` - (Internet Explorer)
Browser Prefixes – Why they Exist...

- If the browser were to jump straight to the unprefixed, standard property, they would be locked into whatever behaviour they originally use.

- A prefixed property indicates to developers that the property is somewhat experimental and subject to change.

- It gives the browsers flexibility to continue making changes if necessary, which allows the browsers to release and refine new properties more quickly.

- This, in turn, gives developers the opportunity to use new properties sooner and participate in the refinement process through testing in real world scenarios.
Browser Prefixes - Problems

- The chief complaint against them is that you often end up with a few lines of CSS that all accomplish the same thing, such as:

```
div {
  -moz-transform: rotate(45deg);
  -o-transform: rotate(45deg);
  -webkit-transform: rotate(45deg);
  transform: rotate(45deg);
}
```

- This repetition adds to the file size of your style sheets and can be just plain annoying.
Another problem with prefixes is that they don’t validate.

This isn’t a problem in and of itself - validation is just a troubleshooting tool, so if you know why you are getting an error for a prefixed property, you can just ignore that error and move on.

But having a whole bunch of “benign” errors for prefixed properties mixed in with the others can make it harder to spot the errors you’re really concerned about.
Browser Prefixes

“It’s a little like a vaccine—the shot hurts now, true, but it’s really not that bad in comparison to the disease it prevents. And in this case, you’re being vaccinated against a bad case of multi-year parser hacking and browser sniffing. We suffered through that long plague once already. Prefixes will, if used properly, ward off another outbreak for a long time to come.”

Eric Meyer (www.alistapart.com/articles/prefix-or-posthack)
How to Use Browser-Specific Properties

- [http://shouldiprefix.com/](http://shouldiprefix.com/)

When you use prefixed properties, you should always include the non-prefixed property as well, and always after all the prefixed versions.

So when the browser supports the non-prefixed property, it will use it, overriding the prefixed property listed earlier and using the newer, more correct behaviour.

Another optional guideline to follow when using browser prefixes is to always preemptively include all possible prefixed versions, even if some are not being used, on the chance that later they will be.
How to Use Browser-Specific Properties

- No matter which prefixed properties you choose to include, it’s a good idea to put comments in your CSS indicating which property is used by which browser. It’s not always as obvious as you might think.

- For instance, here’s how a group of border-radius properties might look with comments:

  - `moz-border-radius: 20px; /* Firefox */`
  - `webkit-border-radius: 20px; /* Safari 4 and earlier */`
  - `border-radius: 20px; /* Opera, Chrome, Safari 5, IE 9 */`

- By including these comments, it makes it easy to later remove the properties you no longer need when you decide to drop support for a particular browser.
OTHER BROWSER SUPPORT SOURCES

- Mozilla, Opera, and Safari quite helpfully maintain their own lists of what they support:
  - [https://developer.mozilla.org/en/Mozilla_CSS_support_chart](https://developer.mozilla.org/en/Mozilla_CSS_support_chart)
  - [www.opera.com/docs/specs](http://www.opera.com/docs/specs)
  - [https://developer.apple.com/library/content/documentation/AppleApplications/Reference/SafariCSSRef/Articles/StandardCSSProperties.html](https://developer.apple.com/library/content/documentation/AppleApplications/Reference/SafariCSSRef/Articles/StandardCSSProperties.html)

- Other browser support sites include:
  - [www.findmebyip.com/litmus](http://www.findmebyip.com/litmus) lists support for the major CSS3 properties and selectors, as well as HTML5 features, in many browsers.
OTHER BROWSER SUPPORT SOURCES

- “When can I use...” (http://caniuse.com/#cats=CSS3) lists support for several popular CSS3 properties and techniques in current, older, and upcoming versions of the big-five browsers.
- Standardista (www.standardista.com/css3) currently includes detailed support charts for borders, backgrounds, columns, @font-face, and selectors. More modules are added periodically.
- QuirksMode’s “CSS contents and browser compatibility” (www.quirksmode.org/css/contents.html) lists support for a variety of CSS3 and 2.1 selectors and properties. Each has its own page with details of how it should work and how browsers handle it.
- The site CSS Infos maintains lists of -moz- and -webkit- properties, showing which version of the browser each property appeared in. See http://css-infos.net/properties/firefox.php and http://css-infos.net/properties/webkit.php.
- Campaign Monitor provides a guide on email clients’ CSS support, with CSS3 properties indicated, at www.campaignmonitor.com/css, so you know what you can and can’t use in HTML email newsletters.
MORE ON PROGRESSIVE ENHANCEMENT

- “Progressive Enhancement: What It Is, And How To Use It?,” by Sam Dwyer (www.smashingmagazine.com/2009/04/22/progressive-enhancement-what-it-is-and-how-to-use-it)