References

HTML5 & CSS3 Develop with Tomorrow's Standards Today, Hogan, Pragmatic Programmers

Various Web pages as indicated on individual slides
## New/Improved Features in CSS3

<table>
<thead>
<tr>
<th>Borders</th>
<th>Backgrounds</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>border-color</td>
<td>background-origin and background-clip</td>
<td>HSL colors</td>
</tr>
<tr>
<td>border-image</td>
<td>background-size</td>
<td>HSLA colors</td>
</tr>
<tr>
<td>border-radius</td>
<td>multiple backgrounds</td>
<td>opacity</td>
</tr>
<tr>
<td>box-shadow</td>
<td></td>
<td>RGBA colors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text effects</th>
<th>User-interface</th>
<th>Selectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>text-shadow</td>
<td>box-sizing</td>
<td>attribute selectors</td>
</tr>
<tr>
<td>text-overflow</td>
<td>resize</td>
<td></td>
</tr>
<tr>
<td>word-wrap</td>
<td>outline</td>
<td></td>
</tr>
<tr>
<td></td>
<td>nav-top, nav-right, nav-bottom, nav-left</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basic box model</th>
<th>Generated Content</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>overflow-x,</td>
<td>content</td>
<td>media queries</td>
</tr>
<tr>
<td>overflow-y</td>
<td></td>
<td>multi-column layout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Web fonts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>speech</td>
</tr>
</tbody>
</table>
## Some Pseudo-classes

<table>
<thead>
<tr>
<th>Pseudo-class</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:nth-of-type</td>
<td>p:nth-of-type(2n+1){color: red;}</td>
<td>Finds all n elements of a certain type.</td>
</tr>
<tr>
<td>:nth-child</td>
<td>p:nth-child(2n+1){color: red;}</td>
<td>Finds a specific child element counting forward.</td>
</tr>
<tr>
<td>:last-child</td>
<td>p:last-child{color:blue;}</td>
<td>Finds the last child element.</td>
</tr>
<tr>
<td>:nth-last-child</td>
<td>p:nth-last-child(2){color: red;}</td>
<td>Finds a specific child element counting backward.</td>
</tr>
<tr>
<td>:first-of-type</td>
<td>p:first-of-type{color:blue;}</td>
<td>Finds the first element of the given type.</td>
</tr>
<tr>
<td>:last-of-type</td>
<td>p:last-of-type{color:blue;}</td>
<td>Finds the last element of the given type.</td>
</tr>
<tr>
<td>:after</td>
<td>span.weight:after { content: &quot;lbs&quot;; color: #bbb; }</td>
<td>Used with content to insert content after the specified element.</td>
</tr>
</tbody>
</table>
```css
tr:nth-last-child(2){
  color: green;
}

tr:last-child{
  font-weight: bolder;
}

tr:last-child td{
  font-weight: bolder;
}

tr:last-child td:last-child{
  font-size: 24px;
}

tr:nth-last-child(-n+3) td{
  text-align: right;
}
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
<th>Quantity</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coffee mug</td>
<td>$10.00</td>
<td>5</td>
<td>$50.00</td>
</tr>
<tr>
<td>Polo shirt</td>
<td>$20.00</td>
<td>5</td>
<td>$100.00</td>
</tr>
<tr>
<td>Red stapler</td>
<td>$9.00</td>
<td>4</td>
<td>$36.00</td>
</tr>
<tr>
<td><strong>Subtotal</strong></td>
<td></td>
<td></td>
<td><strong>$186.00</strong></td>
</tr>
<tr>
<td><strong>Shipping</strong></td>
<td></td>
<td></td>
<td><strong>$12.00</strong></td>
</tr>
<tr>
<td><strong>Total Due</strong></td>
<td></td>
<td></td>
<td><strong>$198.00</strong></td>
</tr>
</tbody>
</table>
IE again

IE 8 and earlier does not support css selectors

So another workaround selectivizr.js

http://selectivizr.com/ Use with jQuery or other library

<script type="text/javascript" src="[JS library]"></script>
<!-[if (gte IE 6)&(lte IE 8)]>
  <script type="text/javascript" src="selectivizr.js"></script>
  <noscript><link rel="stylesheet" href="[fallback css]" /></noscript>
<![endif]-->

Thursday, February 3, 2011
Media Queries

Select CSS based on media features

- width
- height
- device-width
- device-height
- orientation
- aspect-ratio
- device-aspect-ratio
- color
- color-index
- monochrome
- resolution
- scan
- grid
Media in link

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>SizeDetect</title>
  <link rel="Stylesheet" href="ipad.css" type="text/css" media="screen and (max-device-width: 768px)"/>
  <link rel="Stylesheet" href="phone.css" type="text/css" media="screen and (max-device-width: 320px) ">
</head>
<body>
  <p>Red = phone, Blue = iPad, Black = desktop</p>
</body>
</html>
```

Red = phone,

max-device-width gives an upper bound on the actual device-width. Any device with a smaller width will match so the iphone matches max-device-width: 768px. The last style sheet that matches overrides properties set in previous one
Device-width

<link rel="Stylesheet" href="ipad.css" type="text/css" media="screen and (device-width: 768px)"/>
device has exact width 768px

<link rel="Stylesheet" href="ipad.css" type="text/css" media="screen and (min-device-width: 700px) and (max-device-width: 768px)"/>
device has width between 700 and 768px

px = CSS pixels not physical pixels
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>SizeDetect</title>
    <link rel="Stylesheet" href="example.css" type="text/css"/>
</head>
<body>
<p>Red = phone, Blue = iPad, Black = desktop</p>
</body>
</html>

@media (device-width: 768px) { p {color: blue; font-size: 3em; } }

@media (device-width: 320px) { p {color: red; } }
How to detect iPhone 4

In CSS

@media (-webkit-min-device-pixel-ratio: 2) { p {color:yellow}}
Orientation

<!DOCTYPE html><html lang="en">
<head>
    <meta charset="utf-8" />
    <title>Orientation</title>
    <link rel="Stylesheet" href="orientation.css" type="text/css"/>
</head>
<body>
    <p>Red = landscape, Blue = portrait</p>
</body>
</html>

orientation.css

@media (orientation: landscape) { p {color: red; font-size: 3em; } }

@media (orientation: portrait) { p {color: blue; } }

Thursday, February 3, 2011
Android

Red = landscape, Blue = portrait

Red = landscape, Blue = portrait
Loading CSS based on user agent

```html
<!DOCTYPE html><html lang="en">
<head><meta charset="utf-8" /><title>SizePlatform</title>
    <script lang="javascript">
        var loadCSS = function(file) {
            var link = document.createElement('link');
            link.href = file;
            link.rel = 'stylesheet';
            link.type = 'text/css';
            document.getElementsByTagName('head')[0].appendChild(link); }
    </script>
</head>
<body onload="
    if (navigator.userAgent.match(/iPhone/i)) { loadCSS('iphone.css');
    } else if  (navigator.userAgent.match(/iPad/i)) { loadCSS('ipad.css')
    } else if  (navigator.userAgent.match(/Android/i)) { loadCSS('android.css')
    }
">
<p>Red = phone, Blue = iPad, Android = Yellow</p>
</body>
</html>
```
User agent

<!DOCTYPE html><html lang="en">
<head>
    <meta charset="utf-8" />
    <title>User Agent</title>
</head>
<body onload="alert(navigator.userAgent)">
<p></p>
</body>
</html>
One Css file, Different CSS classes

.ipad p { color: blue; font-size: 3em; }
.iphone p { color: red; }
.android p { color: yellow; }
One Css file, Different CSS classes

```html
<!DOCTYPE html><html lang="en">
<head>
  <meta charset="utf-8" />
  <title>CSSClass</title>
  <link rel="Stylesheet" href="joint.css" type="text/css"/>
  <script type="text/javascript">
    var cssclass = "desktop";
    if (navigator.userAgent.match(/iPhone/i)) {
      cssclass="iphone";
    } else if  (navigator.userAgent.match(/iPad/i)) {
      cssclass="ipad";
    } else if  (navigator.userAgent.match(/Android/i)) {
      cssclass="android";
    }
    document.documentElement.className += ' ' + cssclass;  
  </script>
</head>
<body>
<p>Red = phone, Blue = iPad, Android = Yellow</p>
</body></html>
```
Android User Agent

Android supports many screen sizes

100+ Android tablets announced in January

Screen size may be more important than Android user agent
Loading different pages in JavaScript

```javascript
if((navigator.userAgent.match(/iPhone/i)) ||
   (navigator.userAgent.match(/Android/i)) ||
   (navigator.userAgent.match(/iPod/i)))
{
    location.replace("http://sitename.com/m/");
}
```

A poor way to do it as requires two round trips to server
Apache

in .htaccess

#redirect mobile browsers
RewriteCond %{HTTP_USER_AGENT} ^.*iPhone.*$
RewriteRule ^(.*)$ http://mobile.yourdomain.com [R=301]
RewriteCond %{HTTP_USER_AGENT} ^.*BlackBerry.*$
RewriteRule ^(.*)$ http://mobile.yourdomain.com [R=301]
RewriteCond %{HTTP_USER_AGENT} ^.*Palm.*$
RewriteRule ^(.*)$ http://mobile.yourdomain.com [R=301]

This way there is only one round-trip to server
Web Accessibility
Web Accessibility

Designing Web sites that are flexible to meet different user needs, preferences, and situations

May not be able to see, hear, move

May not have or be able to use a keyboard or mouse
Provide equivalent alternatives to auditory and visual content.

<img src="foo" alt="this is an image of a Foo Bear" border="0" align="middle">

Use alt tag in img, input, applet

Screen readers read the alt tag
Don't rely on color alone

Ensure that text and graphics are understandable when viewed without color.
Use markup & style sheets and do so properly

Control presentation with style sheets rather than with presentation elements and attributes

Don't use h1 tag just to change font
Create tables that transform gracefully

For data tables
   Use TD to identify data cells and TH to identify headers

For data tables that have two or more logical levels of row or column headers
   Use THEAD, TFOOT, and TBODY to group rows
   Use COL and COLGROUP to group columns
Ensure that pages featuring new technologies transform gracefully

Organize documents so they may be read without style sheets
Ensure user control of time-sensitive content changes

Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or stopped

People with photosensitive epilepsy can have seizures triggered by flickering or flashing
Sample Accessibility Checker

http://achecker.ca/checker/index.php

For a list of tools see
http://www.w3.org/WAI/ER/tools/complete
HTML5 & Accessibility

Roles
Used by screen readers

```html
<header id="page_header" role="banner">
  <h1>AwesomeCo Blog!</h1>
</header>
```
## Widget Roles

<table>
<thead>
<tr>
<th>Alert</th>
<th>Scrollbar</th>
<th>ComboBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alertdialog</td>
<td>Slider</td>
<td>Grid</td>
</tr>
<tr>
<td>Button</td>
<td>Spinbutton</td>
<td>Listbox</td>
</tr>
<tr>
<td>Checkbox</td>
<td>Status</td>
<td>Menu</td>
</tr>
<tr>
<td>Dialog</td>
<td>Tab</td>
<td>Menubar</td>
</tr>
<tr>
<td>Gridcell</td>
<td>Tabpanel</td>
<td>Radiogroup</td>
</tr>
<tr>
<td>Link</td>
<td>TextBox</td>
<td>Tablist</td>
</tr>
<tr>
<td>Log</td>
<td>Timer</td>
<td>Tree</td>
</tr>
<tr>
<td>Marquee</td>
<td>Tooltip</td>
<td>Treegrid</td>
</tr>
<tr>
<td>MenuItem</td>
<td>Treeitem</td>
<td></td>
</tr>
<tr>
<td>MenuItemCheckbox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MenuItemRadio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Progressbar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radio</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Document & Landmark Roles

**Document Structure Roles**

- article
- columnheader
- definition
- directory
- document
- group
- heading
- img
- list
- listitem
- math
- note
- presentation

**Landmark Roles**

- region
- row
- rowheader
- separator
- toolbar
- application
- banner
- complementary
- contentinfo
- form
- main
- navigation
- search