Building the Web In 2012

Grant Root

Slides

- These slides can be found at:
  http://www.rootcentral.org/news.php

Scope of This Talk

- Building blocks
- Guiding principles
- Current directions
- No details on hosting or editing tools
- No tutorials
- No links
- Search terms (highlighted in yellow)

The Nature of the Web

- Stateless transactions
- Structured text documents
- Images
- Visual style imposed by the client software (browser)
- Client-side behaviors (scripts and applets)

Languages

- HTML (hypertext markup language)
  - Content (text and structure)
- CSS (cascading stylesheets)
  - Presentation (visual style)
- Javascript (no relation to Java)
  - Behavior

Challenges

- Bandwidth may be limited
    - Modems -> broadband -> mobile
- Screen sizes vary
    - Older monitors -> newer monitors -> mobile
- Browser capabilities vary
    - Brand, version, OS, user settings
- Accessibility is critical
    - Large fonts, screen readers

Progressive Enhancement

- Separate content, presentation and behavior
- Start simple, with plain text content
- Mark it up for meaningful structure
- Add enhancements in layers that can be ignored
  - Presentation
  - Behavior
- Screen reader users will love you, and so will Google!

Old and Busted

- HTML, style and scripting all mixed up in a bucket
- Hard to write, hard to maintain
- No progressive enhancement

```
<p onclick="someScript.js();return
  false;"><center><font color="red">
<bold>Hello!</bold></font></center></p>
```

New Hotness

```
<p class="greeting">Hello!</p>
```

```
.greeting {
color: red;
text-align: center;
font-weight: bold;
}
```

```
<script src="someScript.js"
  type="text/javascript"></script>
```

Content is King

- Start with a simple text document
- Analyze and meet visitors' needs
- Write simply and clearly
- Keep it short
- Check spelling and punctuation
- Include important search terms

Structure: HTML

- Markup language: tags to indicate structure
- Pick a standard
    - HTML 5 is the future
- Start simple and build your skills
- Use semantic markup
    - Use of appropriate tags will convey meaning to browsers, search engines and other programs
    - Tables are for tabular data only

No Fear!

- This is a complete and valid HTML 5 web page:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My Dog</title>
  </head>
  <body>
    <h1>My Dog</h1>
    <p>My dog is a German Shepherd. His name
    is Blackie. He is seven years old. He likes
    to sleep in my bed.</p>
  </body>
</html>
```

Style: CSS

- Link to a separate stylesheet file
  - Helps with maintainability and reduces errors
- Neutralize the browsers' built-in stylesheets
  - Use Yahoo's YUI 3 CSS Reset as an equalizer

```
<link rel="stylesheet" type="text/css"
    href="css/yui/reset-min.css" />
<link rel="stylesheet" type="text/css"
    href="css/main.css" />
```

Style: CSS

- Use media queries to bring in extra stylesheets
  - Make the page “printer-friendly”
  - Adapt to Smaller screens:
    - Responsive Web Design and Mobile First

```
<link rel="stylesheet" type="text/css"
    media="print"
    href="css/print.css" />
<link rel="stylesheet" type="text/css"
    media="screen and (max-width: 480px)"
    href="css/phone.css" />
```

Style: CSS

- Be aware of the version support
  - CSS 2 almost universally supported
  - CSS 3 support coming along in newer browsers
- Avoid complicated browser hacks if possible
- Allow for graceful fallback
- Use CSS inheritance to simplify styles
  - Style from general to specific
- Organize style sheets for better maintainability

Style: CSS

- Web fonts
  - Freedom of fonts
  - No more image replacement hacks
  - Text is always text – yay!
- CSS libraries and frameworks
  - Grids, typography help, form and print styles
  - YUI, Normalize.CSS, intuit.css, Blueprint, 960 Grid System, etc.

Behavior: Javascript

- What “behavior” do I need?
  - Maybe none at all!
  - E-mail (de-)obfuscation
  - Asynchonous page updates
  - Interactivity
  - Transitions, animations, other special effects
  - HTML / CSS detection and augmentation libraries

Behavior: Javascript

- Use Unobtrusive Javascript to add behavior in a way that is consistent with progressive enhancement and accessibility.
- Keep Javascript in separate files for maintainability.
- Use a Javascript library like JQuery
  - Abstracts away browser differences
  - Simplifies many common functions, such as Ajax and document object model (DOM) manipulation

Help with Browser Differences

- Testing services such as Browsershots
  - See your page in many browsers
- Capability detection tools such as Modernizr
  - Test for HTML 5 and CSS 3 features
- Shims, fallbacks, and polyfills
  - Implement some HTML 5 capabilities in browsers that don't natively support them

Adding Semantics

- Make your pages' structure more readable by programs
  - Search engines and other spiders
  - Browser integration with calendars, contacts, etc.
- Extra data built into the HTML
- Microformats, RDFa, microdata
- Schema.org

Search Engine Optimization

- No tricks
- Content, content, content
- Clean markup
- Semantics
- Site map
- Pinging search engines

Drinking from the Firehose

- Books
- Article sites (e.g. A List Apart, HTML 5 Doctor)
- Personal blogs and microblogs (Twitter)
- User groups (e.g. Dayton Web Developers Meetup)
- Google!

Questions

?