W3Conversions

Using New Media with Adobe Dreamweaver

Stephanie Sullivan

Stephanie Sullivan

- W3Conversions - Web Standards and Accessibility Company
- Corporate Trainer, Consultant & sub-contractor for XHTML/CSS development
- Adobe Community Expert
- Co-Lead Adobe Task Force for Web Standards Project (WaSP)
- Partner - CommunityMX.com - web tutorial site
- Author - Mastering CSS with Dreamweaver CS3 [New Riders - Voices that Matter] authored with Adobe's Greg Rewis
- List Mom for WebWeavers & moderator for SEM 2.0
- Contact - stef@w3conversions.com

What ARE Standards

- Web standards recommend separating the *content* of the document from the *presentation* and *behavior* layers
    - (X)HTML - content
    - CSS - presentation
    - Javascript - behavior

What About Flash?

- Flash can contain content, presentation and behavior
- Actionscript 3 complies with ECMA standards
- Flash can and should be made accessible
- Flash is sometimes best combined with (X)HTML and CSS for interactive areas

HTML Markup
Content Layer

What is a Web Site?

- Marketing content
- Product information and sales
- Informational content
- The web site **IS** its content
- The decision then is -- How do you best present the content?

Planning

- What parts of your page should be “Plain Old Semantic HTML” [POSH]?
- What portions could be enhanced using Ajax?
- Where is Flash or Flash video best used?

Content and SEO

- Spiders crawling the web are looking for words
- Key words and search terms, relating to your client's content, should be planned from the outset and used in:
  - Title element
  - Headings
  - Main text (Don't use pronouns for keywords)
- Site map should be planned with key search terms in mind

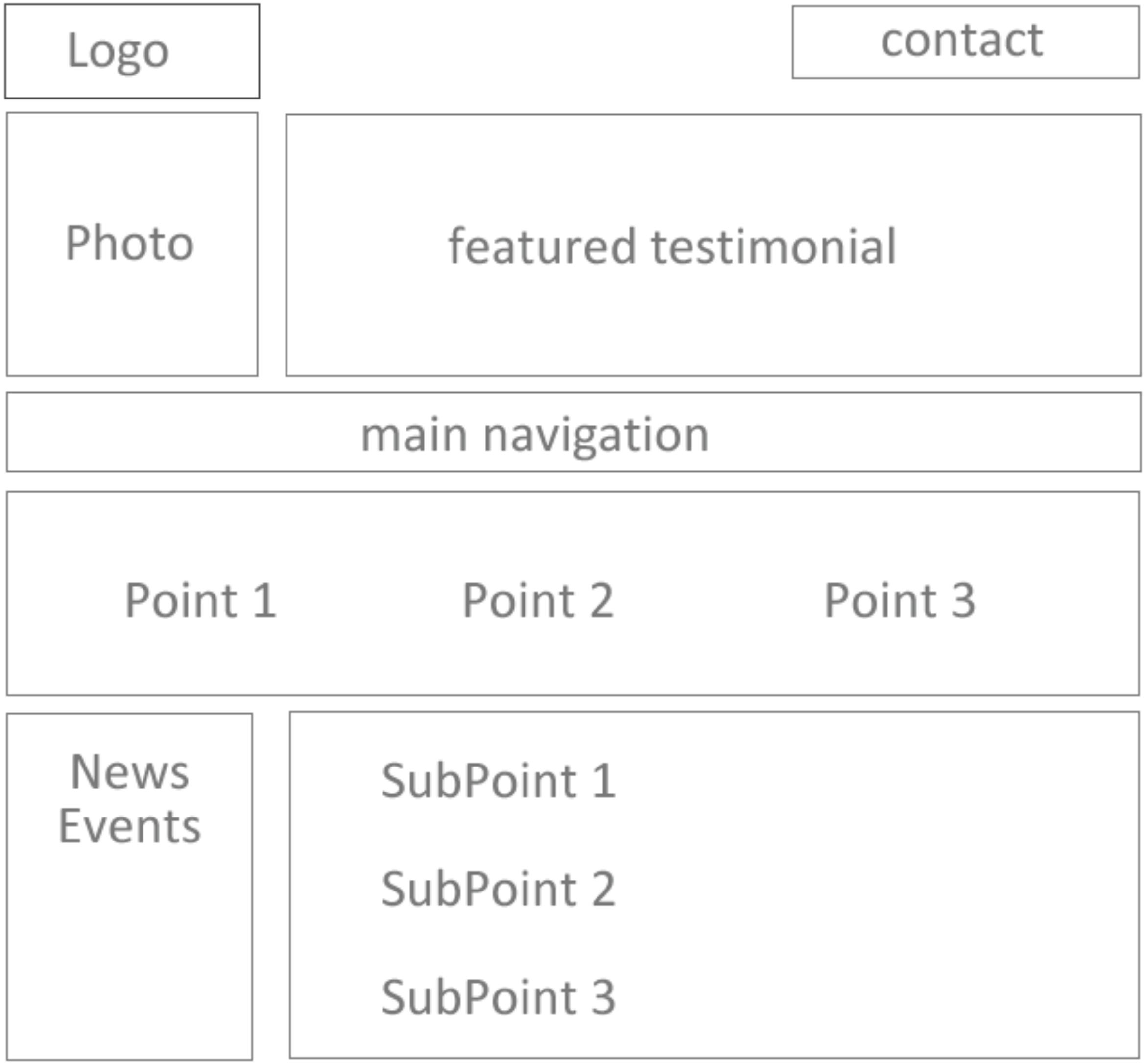
Content and Accessibility

- Your page may be accessed by a variety of user agents and devices
- Your page may be accessed by people with a variety of physical difficulties - low vision, blindness, carpal tunnel, motor difficulties, etc
- Be sure your content is accessible for them all

Create a Wireframe

- Decide how to emphasize key content
- Be logical and consider usability and accessibility

WireFrame the Content



Logo

contact

Photo

featured testimonial

main navigation

Point 1

Point 2

Point 3

News
Events

SubPoint 1

SubPoint 2

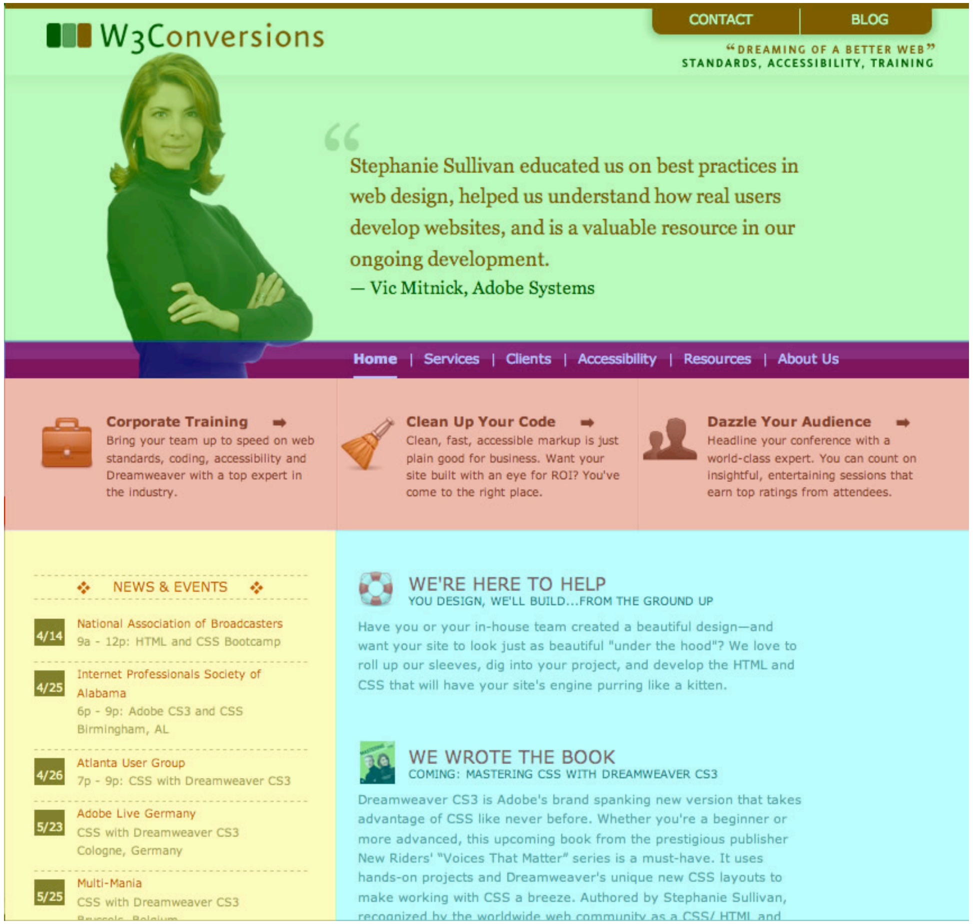
SubPoint 3

Logical Markup

- Content should be marked up relating to its inherent meaning
- A heading should be an h1, h2, h3, h4...
- Text should be in P elements
- Lists should be used (ordered, unordered and definition)
- This is called semantic markup. It's simply the logical meaning of the element itself.

Divide the Design

- Break the design into divs (divisions)



W3Conversions

CONTACT | BLOG

"DREAMING OF A BETTER WEB"
STANDARDS, ACCESSIBILITY, TRAINING

Stephanie Sullivan educated us on best practices in web design, helped us understand how real users develop websites, and is a valuable resource in our ongoing development.
— Vic Mitnick, Adobe Systems

Home | Services | Clients | Accessibility | Resources | About Us

Corporate Training ➡
Bring your team up to speed on web standards, coding, accessibility and Dreamweaver with a top expert in the industry.

Clean Up Your Code ➡
Clean, fast, accessible markup is just plain good for business. Want your site built with an eye for ROI? You've come to the right place.

Dazzle Your Audience ➡
Headline your conference with a world-class expert. You can count on insightful, entertaining sessions that earn top ratings from attendees.

NEWS & EVENTS

4/14 National Association of Broadcasters
9a - 12p: HTML and CSS Bootcamp

4/25 Internet Professionals Society of Alabama
6p - 9p: Adobe CS3 and CSS
Birmingham, AL

4/26 Atlanta User Group
7p - 9p: CSS with Dreamweaver CS3

5/23 Adobe Live Germany
CSS with Dreamweaver CS3
Cologne, Germany

5/25 Multi-Mania
CSS with Dreamweaver CS3
Brussels, Belgium

WE'RE HERE TO HELP
YOU DESIGN, WE'LL BUILD...FROM THE GROUND UP

Have you or your in-house team created a beautiful design—and want your site to look just as beautiful "under the hood"? We love to roll up our sleeves, dig into your project, and develop the HTML and CSS that will have your site's engine purring like a kitten.

WE WROTE THE BOOK
COMING: MASTERING CSS WITH DREAMWEAVER CS3

Dreamweaver CS3 is Adobe's brand spanking new version that takes advantage of CSS like never before. Whether you're a beginner or more advanced, this upcoming book from the prestigious publisher New Riders' "Voices That Matter" series is a must-have. It uses hands-on projects and Dreamweaver's unique new CSS layouts to make working with CSS a breeze. Authored by Stephanie Sullivan, recognized by the worldwide web community as a CSS/ HTML and

Document Flow

- The “flow” is the natural order of occurrence of the elements within the HTML
- When working with CSS for page layout, the document flow impacts the visual position of page elements – depending upon the method of positioning
- Don’t fight the flow, use it!

The Display Property

HTML elements, by nature, have one of two renderings:

- Inline
  - Inline-level elements render horizontally until they run out of space, then wrap to the next line.
  - They only take as much space as they need
    Examples: img, span, a, em, strong
- Block
  - Block-level elements render vertically as if there's a line break above and below them
  - They take up 100% of their parent container
    Examples: p, div, h1, ul, blockquote

CSS
Presentation Layer

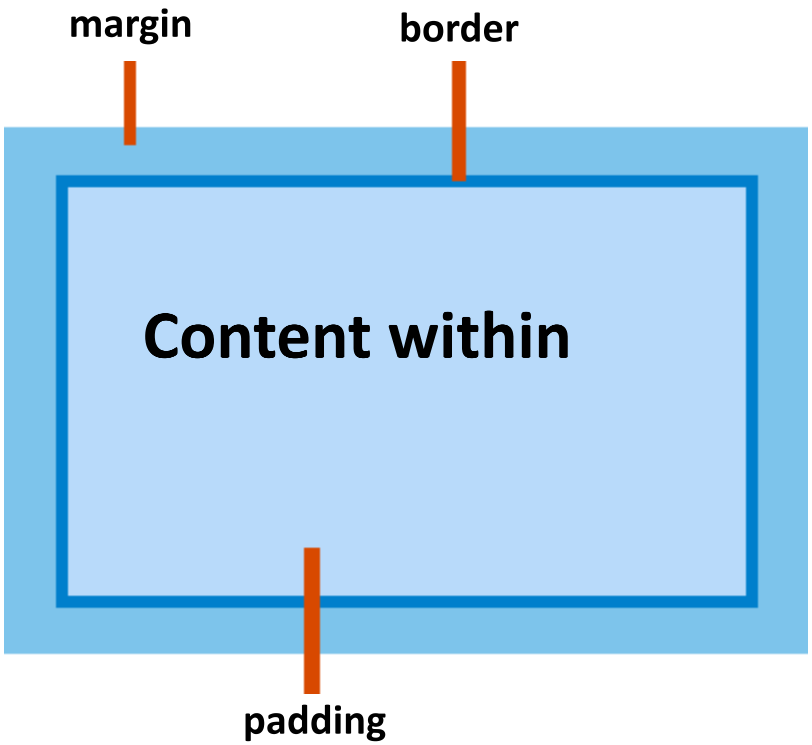
The Display Property

CSS can be used to change the display property of an element

- display: block can be given to a span or an image to make them stack vertically
- display: inline is sometimes used as a fix for Internet Explorer's
  3 px bug (added to your math)
- display: none causes a block to render no box at all

- Changing the display property of an element changes its presentation, but not the nature of the element itself.

Understanding the Box Model



Types of Positioning

- The four types of positioning available using CSS:
- Static
  - The default location of the element in the document flow
- Relative
  - The element’s position is relative to its position in the document flow
- Absolute
  - A “XY” coordinate based upon its parent container
- Fixed
  - A “XY” coordinate based on the viewport

Principles of Floating

- A float must be given a width
- A float must be given a directional value of left or right (there is no top or bottom)
- If you want a float to appear alongside a non-floated element, it must precede that element in the source order of the document
- A float never covers text or inline images
- A float will appear next to another element until there is not enough space, then it will drop down to the next line

Principles of Clearing

- Since a float is taken "out of the flow" of the document, floats inside another container must be cleared in order for their parent container to enclose them completely
- A clearing element in a *non-floated* div will clear **all** floated elements
- A clearing element within a *floated* div will clear **only** within that div
- There are a variety of ways to clear:
  - clearfix on div itself
  - break or empty div with clearing class

Float Drop

- Evident when one div starts below the level of the div next to it

- **Causes:**
- An element, like an image, that is wider than can fit in the space provided. The div will move down until it can fit next to the floats. (Make sure clients who are editing their own sites are aware of their size specs and limitations.)
- Bad math or the 3px text jog in Internet Explorer (unaccounted for in your math)

Five Types of Layouts

- Absolute Positioning
- Fixed
- Liquid
- Elastic
- Hybrid

Absolutely Positioned

- Fixed, pixel-based width
- Pros / Cons
  - ✓ Float drop not a problem since there is no floating
  - ✓ Headers and footers are simple due to set width
  - ◉ Columns are absolutely positioned and taken out of the flow of the document - footer will not “see” them

Fixed

- Specific pixel width - centered

- Pros / Cons
  - ✓ Full background color on columns is easy to achieve (faux columns)
  - ✓ Easy to know exact dimensions for elements within the main content area and avoid float drop*
  - ◉ Columns do not expand with increased text size

Liquid

- Overall width and columns based on percentage of user’s viewport

- Pros / Cons
  - ✓ Allows for creative use of headers - repeat on X axis or show more when browser is wider
  - ◉ Background column color more challenging (liquid faux columns)
  - ◉ More difficult to know exact dimensions for elements to avoid float drop - use min-width

Elastic

- Width based on user's default text size

- Pros / Cons
  - ✓ Layout and columns expand with text size changes not browser width
  - ✓ Allows for creative use of headers - repeat on X axis or show more when browser is wider
  - ◉ More difficult to know exact dimensions for elements to avoid float drop - use min-width

Hybrid

- Overall width based on percentage, while the side columns are based upon em's

- Pros / Cons
  - ✓ Column widths expand with increased text size
  - ✓ Allows for creative use of headers - repeat on X axis or show more when browser is wider
  - ◉ Still challenging to know exact dimensions for elements to avoid float drop - use min-width

Javascript
Behavior Layer

What is AJAX?

Asynchronous JavaScript and XML

NOT!

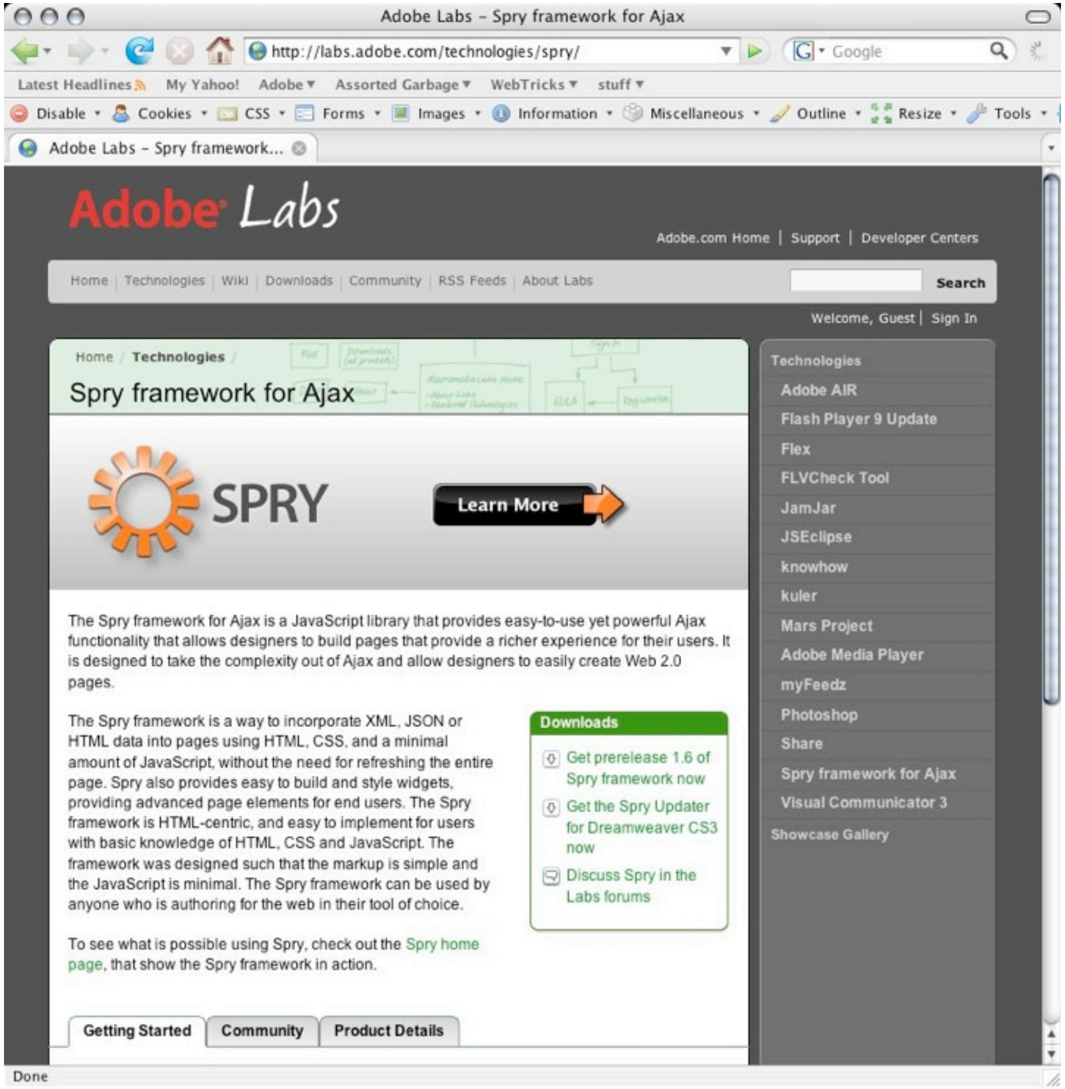
AJAX is now just Ajax. A term used to describe manipulation, via JavaScript, of web page content without a page refresh.

The data does not have to come through XML.

Dreamweaver CS3

- Dreamweaver CS3 ships with a “built-in” Ajax framework called **Spry**

http://labs.adobe.com/technologies/spry/



Spry framework for Ajax

The Spry framework for Ajax is a JavaScript library that provides easy-to-use yet powerful Ajax functionality that allows designers to build pages that provide a richer experience for their users. It is designed to take the complexity out of Ajax and allow designers to easily create Web 2.0 pages.

The Spry framework is a way to incorporate XML, JSON or HTML data into pages using HTML, CSS, and a minimal amount of JavaScript, without the need for refreshing the entire page. Spry also provides easy to build and style widgets, providing advanced page elements for end users. The Spry framework is HTML-centric, and easy to implement for users with basic knowledge of HTML, CSS and JavaScript. The framework was designed such that the markup is simple and the JavaScript is minimal. The Spry framework can be used by anyone who is authoring for the web in their tool of choice.

To see what is possible using Spry, check out the Spry home page, that show the Spry framework in action.

What is the Spry Framework?

- The Spry framework for Ajax is a JavaScript library for web designers. It provides functionality that allows designers to build pages that provide a richer experience for their users.
- It is designed to bring Ajax to the web design community who can benefit from Ajax, but are not well served by other frameworks.
- No browser plug-ins or server-side modules are required for Spry to work.
- Dreamweaver CS3 has features that ease the development of Spry pages but Spry itself is completely tool-agnostic. Any code editor can be used to develop Spry pages.

Spry versus other AJAX Frameworks

- Existing frameworks for AJAX are developer-centric, requiring extensive knowledge of JavaScript, the DOM and XML

```
else
{
var curDataSetRow = dsContext.getCurrentRow();
if (curDataSetRow)
outputStr += curDataSetRow[token.data];
}
```

- The Spry framework is designed with simplicity in mind, by encapsulating the majority of the necessary JavaScript into libraries which allow non-developers access to sophisticated functionality

```
Spry:setrow = “ds1”
```

The Pillars of Spry

- The Spry Framework consists primarily of three core libraries
- Spry Widgets
  - Accordion Pane
  - Spry Menu
  - Sliding Panels
- Spry Data
  - Data Sets
  - Data References
  - Regions
- Spry Effects
  - Appear/Fade, Slide, Blinds, etc.

Spry Widgets

Spry Widgets

- Spry widgets are advanced web components expressed in basic HTML markup, CSS and a little JavaScript.
- Customization and styling is easily done using your existing HTML & CSS skills.
- Spry widgets are accessible. They respond to keyboard navigation and degrade gracefully when JavaScript its turned off.
- It's all about progressively enhancing the page without sacrificing adherence to standards and best practices.

Types of Spry Widgets

- Widgets (currently) include:
  - Accordion
  - AutoSuggest
  - Collapsible Panel
  - Form Controls (Checkbox, Password, Radio Buttons, Select, etc)
  - HTML Panel
  - Menu Bar
  - Sliding Panels
  - Tabbed Panel
  - Tooltip

Using the Accordion Widget

- Anatomy of an accordion
  - Javascript file
  - HTML Markup
    - Container
      - Panel
        - Label
        - Content
- But… the only important thing is the structure.
  Label and Content can be any block level element.
  - Container DIV or UL
    - Panel DIV or LI
      - Label H2

Demonstration

Spry Data

- The Spry Data set transforms complex data sources into a familiar row/column format that can be placed anywhere within your page.
- Supported data sources include XML, JSON* and HTML*
- Easily add Dynamic Regions to your page that control retrieval and placement of data without writing any JavaScript.

✳New in Spry 1.6

Building a Spry page with Dynamic Data

- Attach JavaScript Libraries
  - <script src="SpryAssets/xpath.js" type="text/javascript"></script>
  - <script src="SpryAssets/SpryData.js" type="text/javascript"></script>

- Create Dataset

  -
    ```
    <script type=“text/javascript”>
    var ds1 = new Spry.Data.XMLDataSet(“data/mercury.xml”, “missions/mission”);
    </script>
    ```

Building a Spry page with Dynamic Data, continued

- Add Spry Regions to the page
  - <div spry:region="ds1">Content</div>
- Designate the elements which should be "dynamic"
  - <div spry:region="ds1">{date}</div>
- Specify which elements repeat or have a "master/ detail" relationship
  - <li spry:repeat="ds1">{dynamic data}</li>
  - <div spry:detailregion="ds1">{price}</ div>

Demonstration

Building a Spry page with HTML data

- Over the last few releases, the Spry team has been steadily introducing new features within the framework that will allow the developer to **progressively enhance** their pages to load content dynamically.
- Features include
  - Spry.Utils.updateContent() utility method
    - uses XMLHttpRequest and InnerHTML
  - spry:content attribute within regions
    - shows/hides regions of the page based upon the availability of Javascript
  - HTML Data Set

Spry HTML Data Set

- Allows developers to use the content within an HTML document as its own data source
- No duplication of HTML content because the HTML Data Set extracts its data directly from HTML documents

Demonstration

Q&A

Resources

- www.w3conversions.com
  (my personal site - go to resources page)
- www.communitymx.com
  (over 2,500 Adobe-related tutorials)
- labs.adobe.com/technologies/spry/
  (latest version of Spry)
- Mastering CSS with Dreamweaver CS3
  (www.w3conversions.com/book/)