WEB STANDARDS

WHAT ARE WEB STANDARDS?

- ***Web Standards are specifications or best practices that define different aspects of the World Wide Web technologies, such as coding standards.***

- In 1994 Tim Berners-Lee founded the World Wide Web Consortium.

- *"The W3C mission is to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web."*

- *"W3C publishes documents that define Web technologies. These documents follow a process designed to promote consensus, fairness, public accountability, and quality. At the end of this process, W3C publishes recommendations, which are considered Web standards."*

  www.w3.org

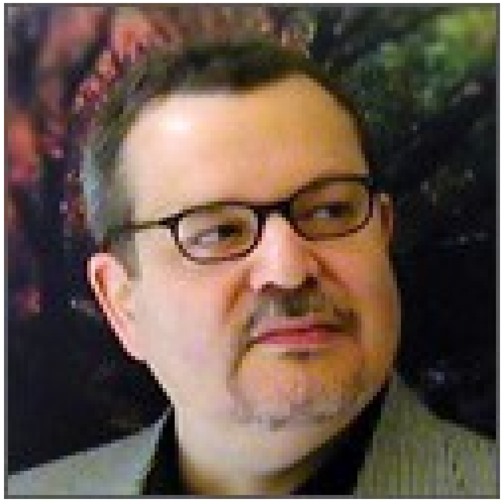
"Web for all, Web on everything"

WHAT ARE WEB STANDARDS?

- ***Web Standards are specifications or best practices that define different aspects of the World Wide Web technologies, such as coding standards.***

- *"Founded in 1998 (co founded by Jeffrey Zeldman), The Web Standards Project (WaSP) fights for standards that reduce the cost and complexity of development while increasing the accessibility and long-term viability of any site published on the Web"*

  www.webstandards.org



- What makes WaSP different?

  A grassroots coalition to persuade Netscape, Microsoft and other browsers to support the standards

- The W3C is the global body for web standards, but other standards bodies exist, such as the European Computer Manufacturers Association (ECMA).

WHY WERE WEB STANDARDS ESTABLISHED?

- WWW started with HTML – Started out as a basic language with a small number of tags, etc. But as it grew in use it became more complex (e.g. Imaging and font control)
- Developers didn’t worry about standards, instead focused on getting the page to look/work correctly.
- Early generation of browsers, such as Netscape Navigator and Microsoft Internet Explorer tolerated nonstandard-markup and would only partially support web standards or did incorrectly.
- Forced developers to ignore standards as well. For example if browsers didn’t support PNG format then developers wouldn’t use them.

WHY WERE WEB STANDARDS ESTABLISHED?

- Inconsistency across browsers would isolate new technologies
- For example CSS,
  - if Netscape 4 ignore CSS rules applied to <body> element,
  - but IE got it right but added loads of padding, then developers wouldn’t want to risk using it.
- This attitude bred a culture of **“Best Viewed In”** websites - coding websites that offer browser-specific pages.
- **Code-forking** became common place – two or three versions of the same page for specific browsers.
- Went against what HTML was designed to do – allow the public to view information on any platform or device!

WHY DOES NOT HAVING WEB STANDARDS MATTER?

- It takes longer to code, having to code separate incompatible scripts for specific browsers.
- More importantly, having code-hacks and workarounds creates more code and adds to the cost of band width.
- Hard to distinguish the presentation of a document from its content

BENEFITS OF USING WEB STANDARDS (1)

- Reduced Markup – less code means faster pages
  - Less code:- More server capacity means less money needed for server space and bandwidth.
  - One script to serve all browsers and platforms!
  - Fewer maintenance problems as junk mark-up and proprietary code is removed.

BENEFITS OF USING WEB STANDARDS (2)

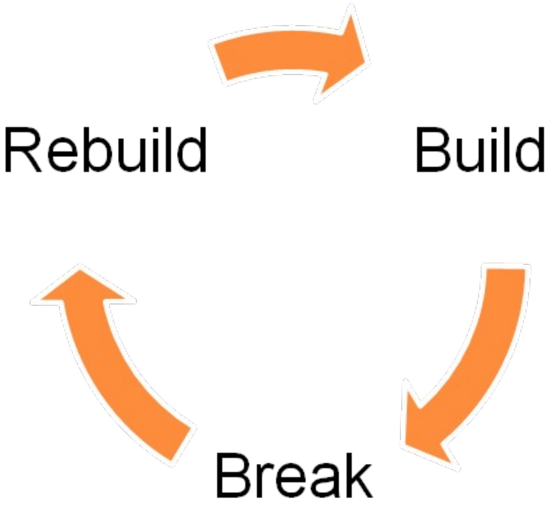
- Increased separation of content and presentation
  - By using CSS to control a sites design, updates and redesigns become easier.
  - Site wide changes can be made instantly through the update of single style sheet.

BENEFITS OF USING WEB STANDARDS (3)

- Improved accessibility: web standards enable us to reach the highest possible number of browsers and devices.
  - Web Standards allow content to be easily read in any browser, SmartPhone, PDA etc. (W3C mission is for 'Web for everything')
  - Importantly it caters for visually impaired users; allows screen readers to easily interpret the content.

BENEFITS OF USING WEB STANDARDS (4)

- Forward compatibility
  - Using web standards, ensures that scripts you write will be always be readable in the future.
  - End the costly cycle of:



Rebuild

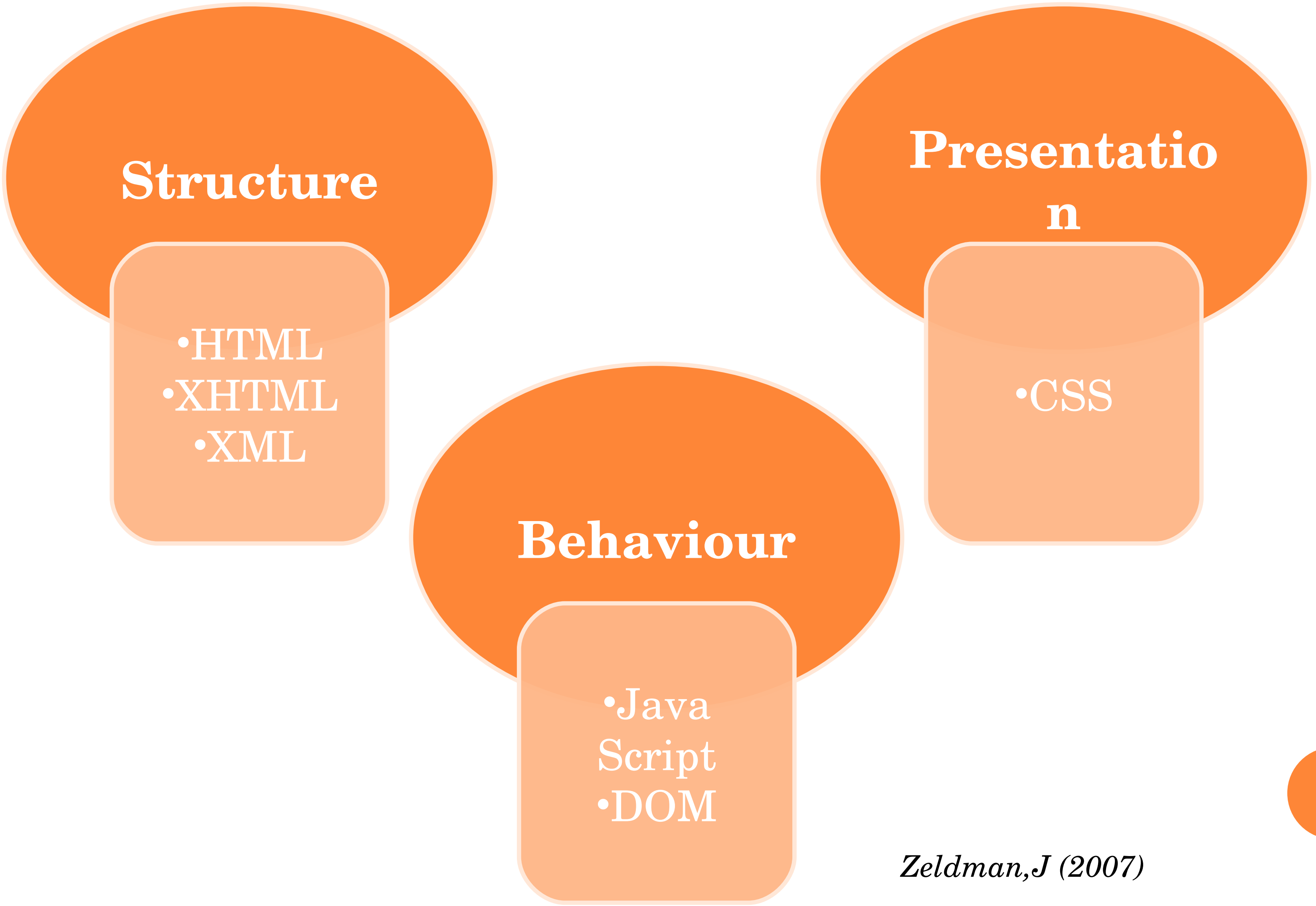
Build

Break

SO HOW DO I USE WEB STANDARDS? (*I WANT EXAMPLES, AND I WANT THEM NOW!*)

- Web Standard Technolgies:
  - XHTML
  - CSS
  - JavaScript (ECMAScript)
  - W3C DOM
- Been supported in browsers for last 10 years.

SO HOW DO I USE WEB STANDARDS?



Structure

- HTML
- XHTML
- XML

Presentatio
n

- CSS

Behaviour

- Java Script
- DOM

Zeldman,J (2007)

XHTML
(EXTENSIBLE HYPERTEXT MARKUP LANGUAGE)

- Reformulation of HTML 4 in XML.
- The same but has stricter rules to what is valid mark-up.
- For example:
  - All tags and attributes must be in lowercase.
  - Attributes must be enclosed with double quotes.
  - All tags must eventually close.

XHTML – QUICK EXAMPLE

- Creating a list of web technologies on a web site:
- **Attempt One**

```
HTML <br />
CSS <br />
JavaScript <br />
PHP <br />
```

- **Problem:**
  - how would we be able to apply different formatting to each element?
  - Text wrapping, ok for this example but if we had a long text element, on a PDA the formatting may become more difficult.

XHTML – QUICK EXAMPLE

- Creating a list of web technologies on a web site:
- <u>**Correct Attempt**</u>

```
<ul>
  <li> HTML </li>
  <li> CSS </li>
<li> JavaScript </li>
<li> PHP </li>
</ul>
```

- Uses completely valid mark up, that will wrap text correctly and allows for independent CSS formatting.

CSS
(CASCADING STYLE SHEETS)

- Allowed presentation to be separated from structure
- Can change layout without touching markup
- Quick and easy to redesign and quicker processing time.
- With CSS, it still allows for backward compatibility, even if an older version opens it, the content will still be readable and useable, and the CSS will be ignored.

JAVASCRIPT

- Enables you to create sophisticated behaviours and effects that work across multiple platforms.

DOWNSIDES TO WEB STANDARDS

- Backward Compatibility Issues
  - Professional development practice to make scripts backward compatible.
  - (As explained) Old browser versions didn't apply standards.
  - Applying web standards could eliminate users still using older versions (diminishing number).
- More difficult to change from the 'old way' of coding to the stricter rules of web development.

WAYS FORWARD

- **Transitional forward compatibility**

  (Mix of traditional and standard-based techniques)

  - **Benefits:** rational backward compatibility plus forward compatibility.
  - **Downside:** Instances of structure and presentation may still be together.

- **Strict Forward compatibility**

  (complete separation of structure, presentation and behaviour.)

  - **Benefits:**
    - No versioning, simplicity and logic to markup.
    - Document structure.
    - Faster and easier, less expensive production and maintenance.
    - Forward compatibility forever.
  - **Downside:**
    - Sites likely to look plain in old browsers.
    - Some techniques easy to achieve with HTML but in CSS are more complicated.
    - Not all browsers can yet handle DOM-based behaviours.

REFERENCES

- Cederholm, D 2004, ‘Web Standards Solutions: The Markup and Style Handbook’, New York: Springer-Verlag
- Macmichael, R.A. 2004, ‘Cleaning up the clutter: why web standards matter’, ‘Perspectives’, Vol ‘6’, Issue ‘2’, Page ’64’
- Web Standards Project, http://www.webstandards.org/
- World Wide Web Consortium, http://www.w3.org/
- Zeldman, J. 2007, ‘Designing with Web Standards’, Second Edition, Berkeley, CA: New Riders