Personal History of Web Development

Circa: 1999 - 2014

1999: Frames

In the earliest (and darkest) days of web design we used frames to style our webpages. They were often problematic, but due to problems with the DOM, the lack of WYSIWYG tools at the time, and sophisticated backend frameworks, they were the first attempt to style the web page with more content than simple pages allowed.

An example of Frame based Layouts

2000-2005: Web as Print

In the net phase of web design, the web was viewed as a surrogate for print design. Layouts were designed in tables, which were then inline-styled using html tags. Web pages tended to load slowly, and were often just photoshop files cut up so little pieces of text or other elements could be reused.

Macromedia offered a suite of products that incorporated into this workflow with DreamWeaver, Flash, and Fireworks creating a full workflow for creating webpages.

An example of Table-based layout

Upsides:

Better Layout and Organization of Content

Better visual presentation

More SEO-friendly

No "Orphaned" users

Downsides:

"Static" content.

Content often took long load times.

Difficult to update, often requiring the designer to resubmit work.
Sidenote: Great time to have been a designer.

Using CGI (Common Gateway Interface) often relied upon very insecure PERL or C programs to deliver email. High volumes of spam.

Hacked websites often delivered spam or viruses, a common Windows experience.

2006–2008: Web 2.0

Web "2.0" had been around for as a theoretical concept for a while, but it was Time's (You) Person of the Year Award that recognized blog creators, wikipedia editors, and so forth as a major force.

The focus shifted from static content to user-generated and controlled content.

Sophisticated Back-end Frameworks

PHP 5 with better OOP and sanitation/validation checks.

Ruby on Rails, with routes, migrations, seeds, and tests.

A shift from Microsoft's ASP architecture to LAMP architecture.

Sophisticated Front-end Frameworks

JQuery, Dojo, YUI, and other frameworks to help developers manage the DOM.

> *The DOM is a mess.*
> *John Ressig, jQuery creator -- 2006*

2009 – 2013: Mobile First

Tremendous success of Apple's Iphone pushed web developers to change their context. Instead of thinking of the web as a print page, the focus shifted to mobile apps.

Raster graphics were replaced more by icon-fonts, SVGs, and other vector-based, infinitely scalable formats.

Elements that required plug-ins became natively supported, like video, audio, and other elements.

Design switched to a "Flat" UI look that emphasized large, flat colors and easy to touch icons and buttons. The goal was to remove visual "clutter" and distractions.

Problem

The biggest problem developers faced were speed implementations. "Native" code ran much faster than HTML.

Limited local storage another major limitation. This also pushed developers away from raster graphics, as well as difficulty in dealing with the load times over slower mobile network connections.

Partially solved by moving towards GPU-accelerated transitions with CSS, shifting towards "micro-libraries" of javaScript that were small and specialized, frameworks designed around mobile-experience, and techniques like concatenation of code, minification of assets, serving special assets to mobile clients, and only loading assets when they were needed.

Also partially solved because the speed of processors and power of the RAM in phones keeps going up, as well as browser manufacturers continually striving for faster speed in code execution against their competitors. As well, wireless companies upgraded their networks to serve content faster.

2014 - The Web is Anything

Not just mobile devices, but cars, watches, glasses, virtual headgear, console gaming applications, literally anything can be the web now.

Emphasis on feature-detection and Shims/Shams/Polyfills to support missing functinonalit.

Developer costs can be significant to make a mobile device behave on a variety of platforms.