Everything You Know About Web Design Just Changed

Jen Simmons
An Event Apart Seattle
April 2018
Everything You Know About Web Design Just Changed

Jen Simmons
An Event Apart Seattle
April 2018
“Everything about web design just changed.”
– Jen Simmons
We need more art.
Real graphic design.
That’s not how the web works.
The first time I saw the same web page through two different browsers at the same time, a feeling of panic swept through me. Why should the pages look so different? I knew various systems had different color spaces and resolutions, but these programs actually presented the pages in different ways. As a graphic designer, how could I design web pages if I didn't know how they would look? Could I let browsers reinterpret my work as they wished?

As it turned out, that is precisely what the Framers of the Web had in mind. They saw browser differences as beneficial. Because every document is marked with structural tags (headlines, list items, captions, and others) they believed users should control the presentation of documents they see. HTML (Hypertext Markup Language) lets the visitor choose how her browser displays the page.

That's like telling the artist how to hold the brushes! I wanted more control. It was either toss in the mouse and go back to print design, or jump into the Web and color outside the lines.

I threw my arm into the trash and started from scratch. I built my own tags and made pages the way I wanted them to look. I began using images to lay out pages in two dimensions, rather than one. When tables became available, I poured columns of text in them. I reduced the colors and made the file sizes small and—guess what?—people came! The numbers actually broke my access counter.

I started teaching others what I had discovered. Those of us who broke the rules challenged each other with every new page we made. Like the painters of Russia's avant garde in the Twenties (Rodченko, Malevich, Popova), we felt like pirates. I received flaming threats from people who said I didn't understand the Web. I was running the information, and I should go back where I came from because I didn't understand the power of the medium. I fired back with more pages. I made it clear we were not going to go away. If the Puritans wanted a fight, we would give them one. [1.1, 4, 6].

Structural Markup versus Layout

Print and on-line design are very different. As a print designer, I use programs like Adobe PageMaker and Adobe Illustrator—programs that write PostScript to describe pages. PostScript is a page-description language. As a web site designer, I am required to use HTML rather than PostScript to make my sites (although this is changing, as you will see in Chapters 12 and 13). Designers want to make web pages with the same degree of control they have over paper. Designers must have layout capabilities, if they are to design pages for artistic, entertainment, or marketing purposes. To quote hypermedia visionary Ted Nelson:

Multimedia must be controlled by dictatorial artists with full say on the final cut.

HTML is a structural markup language. It has no capability to display a fixed-size page or a fixed-size text container. Standard HTML gives the author almost no formatting control. It is meant mainly to describe the structure of information in a document, rather than its appearance. In theory, HTML browsers are responsible for handling display and layout. In practice, browsers make standard HTML look about as compelling as tax forms.

Structure has its uses. Structured documents can be fed into databases and searched easily. Their formatting can improve as HTML becomes more capable, enabling companies with large sets of documentation to enter information only once, then present it many different ways. This is of critical importance to people publishing large reference works, databases, searchable indexes, and documents for people with visual disabilities. As Tim Berners-Lee, the inventor of the Web, states:

Information to be published on the Web must also be formatted accordingly to the rules of the Web standards. These standardized formats assure that all Web users who want to read the material will be able to view it.

Different audiences have different needs. In a perfect world, markup and layout would be in balance. Instead, they are at polar extremes. Today, HTML is under heavy pressure from designers to be more visual, even as Postscript is being pushed to carry more structure.

What will happen in the future? Is the Web on a collision course with disaster? Will visually unrepairable designers battle against academic library scientists in a winner-take-all contest for the eyes and minds of web surfers everywhere?
Changing Approaches to Web Page Layout
Simple HTML
This is my first headline
Here's the Homestead!

To navigate, just click in the direction you want to go. I've put in some navigation aids in places where there isn't any better solution. Many things in the various rooms are linked, and more and more will be linked as I get more time. There are also places that don't go anywhere at this time, but will later. This is very definitely evolving and growing as I get the time.

The pictures throughout the house were taken with an Apple QuickTake 100.

Send me some mail, tell me what you think.

I don't actually live there anymore, so I can't add much to it, but I'm still interested in comments.

..........@outernet
The Unofficial Otter Pops Home Page

(Home of the Six Zippy Flavors)
MSLINK

MSLINK PROJECT HOME PAGE

Graphic Version

MSLINK project aims to provide internet access to people with Multiple Sclerosis by utilizing computers donated by corporations. MSLINK Home page provides a variety of services such as National Multiple Sclerosis Society Mid America Chapter Events, Discussion Area, MS Library and many others. Please take a look at What's New.

What's New and Coming Soon

The following is a list of recent additions to our web. Every month we'll remove the oldest items. The most recent changes are listed first, and each item is linked to the page with the updated content:

- We have added new documents to our library.
- We are opening a discussion session which will enable you to post questions and opinions.
- We are going to open an "Ask the Expert" section.

Certified Sound
Signal Systems

Class I Air Horns

Class II Air Horns

Class III Air Horns

Class IV Air Horns
We need art!

Tables for Layout
CREATING KILLER WEB SITES

The Art of Third-Generation Site Design

David Siegel
Creating Killer Websites
David Siegel, 1996

<!-- PAGE PARAMETERS -->
<HTML><HEAD><TITLE>David Siegel's Home Page</TITLE></HEAD>
<BODY BGCOLOR="#EDEEFF" TEXT="#000055" LINK="#C00000" ALINK="#FF3300" VLINK="#005522">

<!-- BEGIN OUTER MOST TABLE -->
<TABLE BORDER="1" CELLPACING="0" CELLPADDING="0" HEIGHT="1442">
<!-- Outermost table has only one row; this is the left side -->
<TR>
<TD WIDTH="570" HEIGHT="1442" VALIGN="top">
<img VSPACE="0" HSPACE="0" WIDTH="655" HEIGHT="174" SRC="newhome/back_legend.gif">
<img VSPACE="0" HSPACE="0" WIDTH="570" HEIGHT="37" SRC="newhome/arrow_horiz.gif">

<!-- THIS CELL CONTAINS THE RIGHT EXPANSION ARROW -->
<TD ALIGN="left" VALIGN="top"><img VSPACE="0" HSPACE="0" WIDTH="42" HEIGHT="148" SRC="newhome/arrow_tall_vert.gif"></TD>
</TR>
</TABLE>
</BODY></HTML>

7.1 Make your mistakes in Photoshop, not HTML. Creating a layer for each element keeps things flexible.

two images. Without it, there would be a crack, giving away my trick.

The Inner Tables
As the page takes shape, I plug in the cell contents one at a time. Once I splice the arrows back together, I can see the light at the end of the table. Right after that comes the "Common" image individually. These little tables make it easy to copy and paste as individual units when I want to add new items to my page (you can view the source HTML at the Book Site).[7.14]

That's all there is to it. I add some text links at top and bottom, and I'm ready for visitors.

This page is not as flexible as I'd like, of course. As I add new features, I need to call all the images over again, and then rearrange the page's structure. This page is hard-coded, as are most of mine, but that's not the way of the future. Creating Killer Websites.
4.4 A, B Absolute widths keep columns from collapsing, whereas relative widths move with the browser window.

```html
<TABLE BORDER="2">
  <TR>
    <TD WIDTH="100">
      Width=100
    </TD>
    <TD VALIGN=MIDDLE WIDTH="200">
      Width=200 (pixels)
    </TD>
  </TR>
</TABLE>
```

4.5 A, B HTML for absolute (left) and relative (right) column widths.

```html
<TABLE BORDER="2">
  <TR>
    <TD WIDTH="25%">
      Width=25%
    </TD>
    <TD VALIGN=MIDDLE WIDTH="75%">
      Width=75%
    </TD>
  </TR>
</TABLE>
```

4.6 A, B Using cellspacing for white space assures a non-collapsible gutter. (Visit the page yourself and view the Document Source to see the HTML.)
Creating Killer Websites
David Siegel, 1996
Big Healey
Roger Los
los.com/healey/big.html

Don't try this at home! Roger Los, an independent designer (and English sports car enthusiast) in Seattle, has created a very sophisticated frame-based site, using nested frames and JavaScript to keep the frames behaving well. Roger is a site designer with a rare blend of visual and technical acumen. The showpiece of the site is the speedometer dial that tells you what number visitor you are. Many sites have rotating counters; this is the only one I've seen that uses them well.

It's a simple enough trick: Take a photograph of a speedometer, cut it up, and replace the odometer with an active access counter. Doing it well is something else. Each digit is a separate GIF. A special program correlates the number to be displayed with the proper GIF, spits out the HTML, and the whole page serves up dynamically.

Go try it and see for yourself.

Locating Graphic Elements

On the Mac, to see how a page is built, click on individual graphic elements with your cursor, and move the mouse around slightly. You'll be able to "pick up" an image and move it around, pulling its outline out of place on the page so you can see how it fits with other elements. You can also use the View > Document Info command in Netscape Navigator to dissect a page.

Creating Killer Websites
David Siegel, 1996
Creating Killer Websites
David Siegel, 1996

9.18 Here I’ve used single-pixel GIFs to lay out a hotlist.
We need art!

Flash
Phillip
CSS

It’s the web!
designing with web standards

jeffrey zeldman

2003
chapter 7 > tighter, firmer pages guaranteed

Note that image height and width attributes, although useful, are not strictly necessary. Note also that, thanks to CSS, the border attribute is entirely unnecessary, even though we have included it here for the sake of old browsers. Our markup could be as clean and simple as this:

```html
<div id="primenav">
  <a href="/"><img src="/i/home.gif" alt="Home" /></a>
  <a href="/"><img src="/i/dailyreport.gif" alt="The Daily Report" /></a>
  <a href="/glamorous/"><img src="/i/glamorouslife.gif" alt="My Glamorous Life" /></a>
  <a href="/classics/"><img src="/i/classic.gif" alt="Classics, 1995-2002" /></a>
</div>
```

Compare and contrast the compactness and clarity of the preceding markup with the typical table layout version shown here:

```html
<table border="0" cellpadding="0" cellspacing="0" width="500">
  <tr>
    <td align="top" width="150" height="100" bgcolor="#339999">
      <a href="/"><img src="/i/home.gif" width="150" height="100" border="0" alt="Home" /></a>
    </td>
    <td align="top" width="140" height="100" bgcolor="#339999">
      <a href="/"><img src="/i/dailyreport.gif" width="140" height="100" border="0" alt="The Daily Report" /></a>
    </td>
    <td align="top" width="110" height="100" bgcolor="#339999">
      <a href="/glamorous/"><img src="/i/glamorouslife.gif" width="110" height="100" border="0" alt="My Glamorous Life" /></a>
    </td>
    <td align="top" width="100" height="100" bgcolor="#339999">
      <a href="/classics/"><img src="/i/classic.gif" width="100" height="100" border="0" alt="Classics, 1995-2002" /></a>
    </td>
  </tr>
</table>
```

There’s no contest. But compact markup and meta-structural thinking are not limited to CSS layouts. They can power and streamline table-based layouts, too.
Semantic Markup
Preparing Images

Although the site was designed in Photoshop, it's not your typical slice and dice job. Figure 10.2 shows the six images used to create the entire site. Three are for foreground use: the astronaut photo, the "best of breed" dog photo, and the transparent logo GIF used at the top-left corner of the menu bar.

The remaining three images are backgrounds. Arrow.gif is a screened image derived from the logo that will be placed in the background as a watermark. Bgpat.gif, consisting of single-colored pixels alternating with single transparent pixels, will be used to create background color effects in the menu bar. Nopat.gif, a plain-white background, will replace bgpat.gif in CSS rollover effects and might also be used to indicate the visitor's position within the site's hierarchy via an embedded style added to each page at the end of the project (see the following sidebar, "The Needless Image").
Fluid Layouts
This is my first headline

Lorem ipsum dolor sit amet consectetur adipisciing, elit inceptos nunc et litora, enim classe id morbi viverra. Leo in hac vel sed malesie fringilla faucibus. Cur aliquet. Dis nibh pharetra quisque inceptos interdum tortor, portus parturient varius magna cumque feugiat vestibulum egistas, suscipit facilisi accumsan tempus at velit lobortis. Non dictum feugiat hinnentos consectetur per odio ante porta eget sonorous, ac accumsan ligula lacinus vivamus sociis mollis habilisse magna, felis massa dis id malesuada donec ut pulvinar iaculis.

Let’s center the second even longer headline

Let's center the second even longer headline

Fluid Columns
It's the web!

We need art!
Fixed-Width Layouts
This is now the first set of content

This is my first headline


Nullam bibendum hendrerit augue, sod tempus odio. Apgicitia nec ipsum, tempor nullam bibendum hendrerit augue, sod tempus odio. Apgicitia nec ipsum, tempor

Nullam bibendum hendrerit augue, sod tempus odio. Apgicitia nec ipsum, tempor

Nullam bibendum hendrerit augue, sod tempus odio. Apgicitia nec ipsum, tempor
Fixed-sized Content
Responsive Web Design

It's the web!
Responsive Web Design is: A flexible grid (with flexible images) that incorporates media queries to create a responsive, adaptive layout.
This is the Article with a Longer Headline

Fluid Images & Media Queries
It’s not just about layout.
Responsive Web Design has been about so much more than layout.
MOBILE STRATEGY
M DOT WEBSITE?
SCREEN SIZE?
M CONTEXT?
WHAT'S M?
ONE WEB
UNKNOWN
DESKTOP & MOBILE ORG
REORG
BUT CONTENT WON'T FIT
TRUNC
THINK SINGLE COLUMN / M FIRST
MANY SYSTEMS
HEADLESS CMS
IMAGES BIG
RESPONSIVE IMAGES
PERFORMANCE
BUDGET SPEED
YES.
But let’s do talk about layout.
Intrinsic Web Design
Flexible Images

Flexible images or Fixed images, your choice
Flexible (Faux) Column Grid

Real Grid
Rows & Columns
Fixed, fluid, content-sized
Media Queries

Media Queries, as needed
Create set of layouts for different screens.

Design a flexibility model for your system of content.
Intrinsic Web Design
1. Mix Fluid with Fixed
Images
Fixed Images, like pre-RWD
Fluid Images, like RWD
Fluid, vertically
Set width & height
Tracks
Better way to squish fluid grids
Article with a Headline

main {
  display: grid;
  grid-template-columns: 3fr 1fr;
  grid-gap: 0 2rem;
}

img {
  width: 300px;
}
Mix fixed and fluid
Article with a Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Inceptos nunc et litora, cras class id morbi viverra. Leo in hac vel sed molestie fringilla fusce orci, dui nibh pharetra quisque inceptos interdum tortor, porta parturient varius magna curae feugiat vestibulum egestas, suscipit facilisi accumsan torquenter auctor lobortis. Non dictum feugiat himenene consequat per odio ante porto eget senectus, ac occumnn ligula laoreet vivamus sociis mollis habitasse magna, felis massa dis id malesuada donec ut pulvinar acus.

2. Four Stages of Squish
Fixed & Adaptive
Fluid & Responsive
fixed
FR units
minmax()
auto
min-content sizing
This is a sentence with words.

This is a sentence with words.

This is a sentence with words.


fixed
FR units
minmax()  
auto
2. Four Stages of Squish
3. Truly 2-D Layout
Rows and columns
White space
<table>
<thead>
<tr>
<th></th>
<th>200px</th>
<th>minmax()</th>
<th>2fr</th>
<th>1fr</th>
<th>auto</th>
</tr>
</thead>
<tbody>
<tr>
<td>100px</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1fr</td>
<td>1fr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2fr</td>
<td>2fr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Whitespace

youtube.com/layoutland
Can set heights and widths
Setting heights...
3. Truly 2-D Layout
4. Nested Contexts
Flow
Flexbox formatting context
Grid formatting context
Multicolumn formatting context
Flow  block formatting context
Flexbox  formatting context
Grid  formatting context
Multicolumn  formatting context
4. Nested Contexts
5. Ways to Contract and Expand
Shrink / Grow
(aka, Squish)
Wrap, reflow
Add / remove whitespace
Slide one thing behind another. Overlap.
Lorem ipsum dolor sit amet consectetur adipiscing, elit, inceptos nunc et litora, cras id morbi viverra. Leo in hac vel sed molestie fringilla fusce orci, dis nibh pharetra quisque inceptos interdum tortor, porta parturient varus magna curae feugiat vestibulum egestas, suscipii facillis accumsan torment aptent lobortis.
<table>
<thead>
<tr>
<th>20%</th>
<th>1fr</th>
<th>minmax(0, 40ch)</th>
<th>1fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>2fr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1fr</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. Ways to Contract and Expand
6. Media Queries, as needed
Intrinsic Web Design
1. Fluid & fixed
2. Stages of Squishinessness
3. Rows & Columns
4. Nested Contexts
5. Ways Expand & Contract
6. Media Queries, as needed
It's the web!

We need art!
<table>
<thead>
<tr>
<th>CSS Grid</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexbox</td>
<td>Floats</td>
</tr>
<tr>
<td>Alignment</td>
<td>Block</td>
</tr>
<tr>
<td>Writing Modes</td>
<td>Inline</td>
</tr>
<tr>
<td>Multicolumn</td>
<td>Inline-block</td>
</tr>
<tr>
<td>Viewport Units</td>
<td>Display: table</td>
</tr>
<tr>
<td>Transforms</td>
<td>Margin</td>
</tr>
<tr>
<td>Object Fit</td>
<td>Negative margins</td>
</tr>
<tr>
<td>Clip-path</td>
<td>Padding</td>
</tr>
<tr>
<td>Masking</td>
<td>everything else in CSS</td>
</tr>
</tbody>
</table>
Thanks!

jensimmons.com
@jensimmons
layout.land
labs.jensimmons.com