

Resizing: Fixed, Fluid, or Responsive Layouts

The first workable prototype of the Internet came in the late 1960s with the creation of ARPANET, or the Advanced Research Projects Agency Network. Originally funded by the U.S. Department of Defense, ARPANET used packet switching to allow multiple computers to communicate on a single network. The technology continued to grow in the 1970s after scientists Robert Kahn and Vinton Cerf developed Transmission Control Protocol and Internet Protocol, or TCP/IP, a communications model that set standards for how data could be transmitted between multiple networks.

ARPANET adopted TCP/IP on January 1, 1983, and from there researchers began to assemble the “network of networks” that became the modern Internet. The online world then took on a more recognizable form in 1990, when computer scientist Tim Berners-Lee invented the World Wide Web.

While it’s often confused with the Internet itself, the web is actually just the most common means of accessing data online in the form of websites and hyperlinks. The web helped popularize the Internet among the public, and served as a crucial step in developing the vast trove of information that most of us now access on a daily basis.

Early websites were created to share scientific and technical reports. I was largely text-based in its early years. Today’s users would probably see early web usage as a primitive form of online text “chats” that were known as “message boards” There was no need to create online documents that had precise designs and layout. In the early web, content and readability was what mattered.

Because computers/servers networked with the web and user devices became more and more powerful, the web evolved into the dynamic and exciting multi-media experience that we expect today.

Why does this matter to today’s web designers? Because technology and devices continue to change. Web designs and the web programming/coding

needed to deliver designs is changing. Web development, which means delivering modern web designs, is complex. While there are services like Wix and HTML editors like Dreamweaver that can create great looking webs, these tools have limits.

When websites were being designed and created 15 years ago, designs were based on tables – grids like the Insert Table feature used in desktop publishing like MS Word. Called “layout tables”, this system had 2 flaws:

1. Alignment options with table cells and content were limited
2. Using tables to design websites meant that text readers and other accommodations for users with different abilities and needs was not possible.

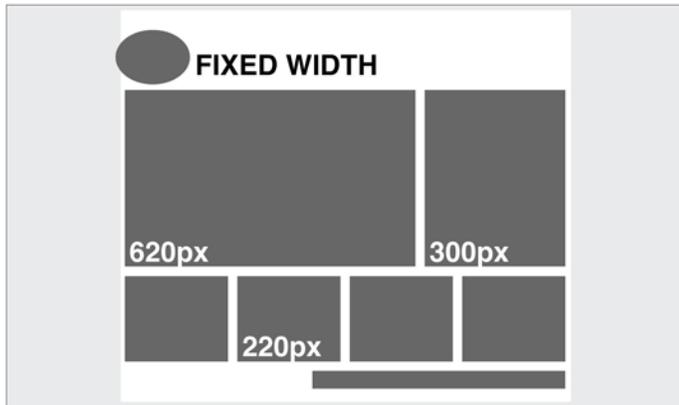
Because of this, modern web design is now based on a “box model”, called CSS, that defines how content is arranged when displayed and provides a context for each design element so that text readers and other technologies can interpret all content in a web.

The following design principles – all essential components of web design, are excerpted from the book, “The Principles of Beautiful Web Design” (Third Edition), by award-winning web designer Jason Beard.

Fixed Width

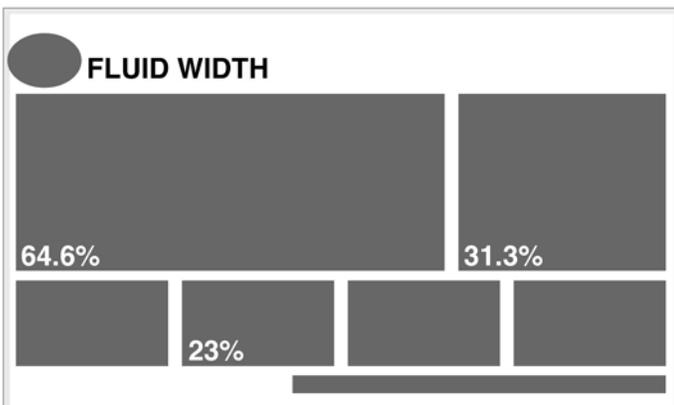
If you’ve started your layout with the [960 Grid System](#) that I talked about earlier, chances are you’re working toward a *fixed-width* layout. These designs are wrapped with a `div` that’s given a pixel-based width (say, 960 pixels), and centered within the `body` tag using `margin: 0 auto;`. If the browser window is wider than 960 pixels, you’ll see the `body` element’s background to the left and right of the container; and if it’s narrower, you’ll have a horizontal scrollbar. Fixed-width layouts are easier to design for—and maintain—because you’re

essentially building exactly what you design in Photoshop or your design program of choice. The graphic below shows a similar layout structure as fixed and as fluid in the same width of browser space:



Fluid Width

A *fluid* or *liquid* layout is designed with percentage-based widths, so that the container stretches when you resize the browser window. These take more thought to plan, as you have to foresee problems that might occur at every possible width. Sometimes pixel-width columns are mixed with percentage-based columns in a fluid layout, but the idea is to show the user as much horizontal content as will fit on their screen. Typically, fluid layouts take advantage of the `min-width` and `max-width` properties of CSS, ensuring the container doesn't become ridiculously narrow or wide.



One reason some designers reject fluid layouts is because they think it impedes them using a grid to create their designs. There are several resources available for designing fluid layouts on a grid,

including [a fluid adaptation of the 960 Grid System](#). Ultimately, though, the decision to use one type of layout over the other should really be determined by the target audience, and the accessibility goals of each individual website.

The pros and cons of each layout type are fairly well-defined, as the table below outlines:

	Pros	Cons
Fixed width	<ul style="list-style-type: none"> • Gives designer more control over how an image floated within the content will look • Allows for planned whitespace • Improves readability with narrower text blocks 	<ul style="list-style-type: none"> • Can appear dwarfed in large browser windows • Takes control away from the user
Liquid width	<ul style="list-style-type: none"> • Adapts to most screen resolutions and devices • Reduces user scrolling 	<ul style="list-style-type: none"> • Challenging to read when text is spanning a wide distance • Harder to execute successfully • Limits or imposes on whitespace

With these pros and cons in mind, I've designed more fixed-width layouts than liquid. I like having control over how the content will display, and working with the background space.

On the flip side, I sometimes enjoy the challenges that liquid layouts bring to the table. But, regardless of personal preferences, it's important to put the needs of your client first. If you're deciding on the width of a fixed-width layout, you have to think about the audience for which you're designing, and create a layout that meets the needs of those users.

An Alternative: Responsive Layouts

It's been said that for every rule, there's an exception. That's usually true in the web world, and designers have come up with many variants and hybrids of fixed-width and fluid-width layout structures. I've seen articles recently that list up to ten different layout types. For the purpose of this chapter, however, most of those could be categorized as either fixed or fluid. On *A List Apart* article, though, introduced a novel layout type that I think we'll be seeing a lot more of. The article was titled "[Responsive Web Design](#)" and it was written by Ethan Marcotte:

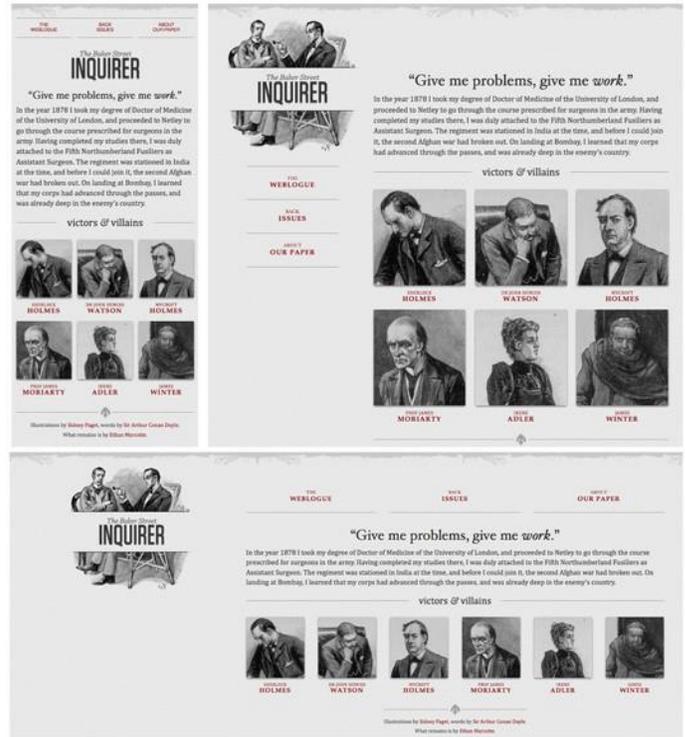
<http://www.alistapart.com/articles/responsive-web-design>

Rather than repeat the entire article here, I highly recommend that you check it out. If I had to name this layout technique myself, I would call it the Omnipotent Mercury Morphing Methodology, but that's probably why I didn't come up with it.

The idea is that as the mobile market makes browser window widths more and more inconsistent, the need for our layouts to be fluid and adapt to different widths is increasingly vital. The technique takes advantage of some fancy fluid scaling and CSS3 media queries to completely change the layout as it hits certain width thresholds.

The graphic at the top of the right column shows screenshots of Ethan's example at three different browser widths. Ethan's responsive web design (or Omnipotent Mercury Morphing Methodology) is a great solution to a problem we've yet to discuss: screen resolutions.

The above material was directly excerpted from "The Principles of Beautiful Web Design" (Third Edition), by award-winning web designer Jason Beaird and is presented here under "fair use" for educational purposes only.



Ethan Marcotte's example at three different browser widths.

Screen Resolutions

The number of horizontal and vertical pixels on a display screen is called the "screen resolution." The more pixels, the more information is visible without scrolling.

Screen resolutions have a pixel count such as 1600x1200, which means 1,600 horizontal pixels and 1,200 vertical pixels. A screen resolution of 1920x1200 means 1,920 pixels horizontally across each of 1,200 lines, which run vertically from top to bottom.

The ratio of width to height of an object. Stating the relationship of one side to the other, it is widely used to describe the shape of a TV or computer screen. For example, the aspect ratio of a standard-definition (SD) screen is 4:3, which is a relatively square rectangle.

The 4:3 means "4 to 3," or four units wide to three units high. High-definition TV (HDTV) has a 16:9 ratio, which is a wide rectangle that is still not as wide as most movie theater (cinema) screens.