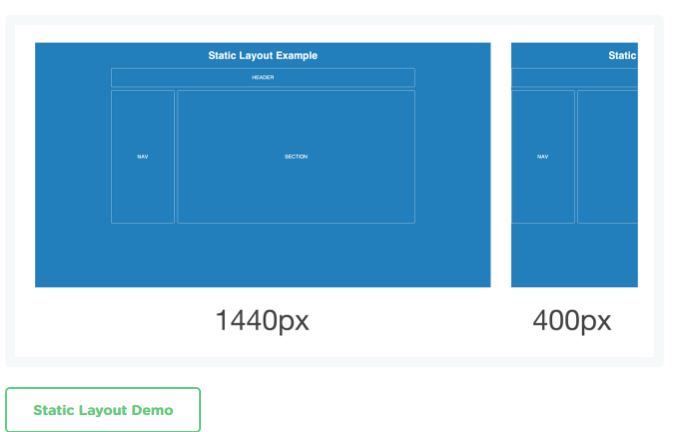
**Page Layouts**

**In short, each page layout name describes how the layout behaves when the page is viewed at different browser widths.** The horizontal width of the browser could change because the website is being viewed on different devices (mobile phones, tablets, desktops, and so on), the website visitor could simply resize the browser window on a desktop device, or the visitor might change their phone from portrait to landscape mode, and so on. These concepts can be difficult to grasp at first, but only because they’re so closely related to one another. Understanding the differences between each one is the key to understanding them individually.

Static Page Layout



A static page layout (sometimes called a “fixed” layout or “fixed width” layout) uses a preset page size and does not change based on the browser width. In other words, the page layout might have a permanent width of 960 pixels no matter what. This is how web pages were traditionally built for many years until modern influences like media queries and responsive design were introduced around the start of the 2010s.

Different devices will treat a static page layout in various ways, so the rendered page could be slightly unpredictable. For example, on a desktop browser, if the window is too small horizontally, then the page will be cut off and horizontal scroll bars will be displayed. On a mobile device like an iPhone, the page will be scaled automatically, allowing the user to zoom in on points of interest (provided that no metatags override this default behavior). When new websites are built, most of them don’t opt for a static layout, because it means that the mobile experience will require a separate website.

## Liquid Page Layout

## 

A liquid page layout (sometimes called “fluid” or “fluid width”) uses relative units instead of fixed units. Typically a liquid layout will use percentages instead of pixels, but any relative unit of measurement will work, such as ems.

A liquid layout often will fill the width of the page, no matter what the width of the browser might be. Liquid layouts don’t require as much thought as a responsive design, but there are some major drawbacks at very large or very small browser widths. If the browser is very wide, some content might be stretched too far. On large screens, a single paragraph might run across the page on a single line. Conversely, a multi-column layout on a small screen could be too crowded for the content.

## Adaptive Page Layout

## 

An adaptive page layout uses CSS media queries to detect the width of the browser and alter the layout accordingly. Adaptive layouts will use fixed unit like pixels, just like a static layout, but the difference is that there are essentially multiple fixed widths defined by specific media queries.

A media query is an expression of logic, and when used in combination with other media queries, they form a basic algorithm. So for example, an adaptive page layout might say something like, “If the browser 500 pixels wide, set the main content container to be 400 pixels wide. If the browser is 1000 pixels wide, then set the main content container to be 960 pixels wide.” Beyond the main content container, other parts of the page might change in width, swap places, or be removed. For example, a two column layout might become a single column layout if the browser is too narrow.

Adaptive layouts are a good stop-gap solution if a legacy static layout needs to be converted to support mobile devices. They also typically require less development time than a responsive layout. The major drawback is that device widths in-between the explicit breakpoints are often less than ideal, with either too much space or not enough space.

## Responsive Page Layout

## 

A responsive page layout uses both relative units and media queries, ostensibly combining the ideas of a liquid layout and an adaptive layout. As the browser increases or decreases in width, a responsive layout will flex just like a liquid layout. However, if the browser goes beyond certain widths defined by media query breakpoints, then the layout will change more dramatically to accommodate a wide or narrow width.

Typically responsive designs are built using a mobile-first approach. That means the mobile layout is designed first, and then as the browser becomes wider on tablets and desktops, the designer will find ways to expand the mobile layout. This tends to create better experiences overall, because it’s easier to expand a design than to try and simplify a large layout for mobile screens.