

Responsive Design

BY NICKY BLEIEL | Senior Member

RESPONSIVE DESIGN MAKES it possible for our content to adapt to any device. This means that technical communicators no longer need to spend time designing and creating deliverables for different devices, we can instead focus on developing and delivering high-quality content. Responsive design gives us the opportunity to create once and deliver to thousands of devices: new ones, older ones, and ones that don't even exist yet.

Here are several reasons why the timing is right for responsive design:

- ▶ Mashable declared 2013 the “Year of Responsive Web Design” (<http://mashable.com/2012/12/11/responsive-web-design/>) and SlashGear reported the forecast for sales of smartphones and tablets will outpace PCs through 2017 (www.slashgear.com/tablet-shipments-to-outpace-total-pc-shipments-in-q4-2013-says-idc-12297574/).
- ▶ Google changed its algorithm to favor responsive content (www.smartinsights.com/search-engine-optimisation-seo/mobile-seo/googles-changing-recommendations-on-seo-for-mobile/).
- ▶ Mobile users want everything that desktop users have—they expect “content parity” (<http://bradfrostweb.com/blog/mobile/content-parity/>).
- ▶ A single responsive output makes supporting a continuous publishing model easier, because we no longer need to build and maintain “Web” and “mobile Web” versions.
- ▶ The usability of our content will be improved because we won't be confusing users with different experiences on different devices. Google has found that sequential and simultaneous usage of different devices is the norm (http://services.google.com/fh/files/misc/multiscreen_world_final.pdf).

An Extremely Short Primer on Responsive Design

Responsive Web design is a technique for designing Web pages that automatically adjust to the device accessing them. A variety of technologies are used, the core ones being media queries, fluid layouts, and flexible images. In addition to HTML and CSS, jQuery, JavaScript, and sometimes PHP are used to create responsive designs. The HTML code of the page never changes, but its presentation adjusts based on the CSS rules specified for the device/browser. No content is lost in responsive design; the content and the page layout simply adjust by stacking or collapsing. (There is no horizontal scrollbar.) Responsive design has another component; it is touch-enabled. This is technically an adaptive design element, but it is necessary to be truly responsive.

A great example of a “stacking” responsive website is www.microsoft.com and a blog post that explains how it works: <http://moz.com/blog/seo-of-responsive-web-design>. And here is an example of responsive content that adapts by collapsing: <http://showme.doctohelp.com/WidgetExplorer/Documents/Welcome.htm>.

Understanding the power of responsive design and how it works is important, but it may not be necessary for technical communicators to code responsive outputs ourselves. We should consider using help authoring tools that give us the option to customize and build responsive outputs. If we do that instead of coding, we can concentrate our efforts on optimizing our content.

“Mobile First” Best Practices for Technical Communicators

Creating responsive outputs means that technical communicators need to adopt a “mobile first” mindset—our content needs to be designed to work well on

mobile (or smaller) screens, because when you write or make other changes for mobile, then the content will work well on tablets, the desktop, etc.

A few “mobile first” best practices:

Keep image files small. Large files will increase load time, which is something mobile users have no patience for; in fact, 74% of mobile users will leave a website that takes more than five seconds to load (www.sitepoint.com/4-common-responsive-web-design-pitfalls/).

Use concise writing (already a good best practice). This isn't just about screen size; those working on smaller devices have less patience to wade through content. (A tertiary benefit is the reduction of translation costs.)

Employ progressive information disclosure. This will make information easier to find and puts the choice to learn more in the hands of the reader.

Improve navigation. Add related links at the end of topics so no one needs to use the device's “back” button, because that makes it easier to navigate away from your content.

Make links easier to use. Don't bury too many links in the same paragraph because they will be harder to use on mobile devices.

Streamline your hierarchies. A table of contents that is four to five levels deep is more challenging to navigate on mobile devices. Avoid sublevel bullets and numbering. It is harder to see the relationships between levels on smaller devices.

Clean up your terminology. Your content needs to work on devices where you tap *and* those where you click. Avoid using device-specific terms.

Excellent resources to learn more about responsive Web design: *Responsive Web Design* by Ethan Marcotte and *Implementing Responsive Design* by Tim Kadlec. 



This column is a quick reference guide exploring “essential” technical communication topics. Please send questions or comments to Nicky Bleiel at nickyb@componentone.com. Follow her on Twitter: [@nickybleiel](https://twitter.com/nickybleiel).