



# Designing for Mobile Devices

by Sofie Bird

**Sofie Bird is a technical writer and web developer at KnowledgeDoc and formerly a lecturer in computer science and software engineering at RMIT. She has served on the Institute of Professional Editors Accreditation Assessment Board, designing and adjudicating the accreditation exam. She holds Masters' degrees in Communications, Commerce and Computer Science.**

## Rethinking online content

Mobile devices are fast outstripping desktops as our primary way to view information online. By 2014, it's expected the majority of internet activities will come from phones and tablets rather than traditional PCs.

*What does this mean for technical writers?*

It means rethinking the way we deliver online content such as web help, intranets and other information. Mobile devices are much more limited than their desktop counterparts, both in speed and screen size. A web page that takes seconds to appear on a desktop browser may take several minutes on a mobile, especially if the user isn't connected to a wi-fi network. A layout that looks beautiful on a full-size browser may be cluttered and impossible to use on a phone screen, and many elements that are used on desktops like mouseover or right-click don't translate well to a touchscreen interface.

One of the old staples of web help design – the table of contents frame – chews up valuable screen real-estate on a mobile device and requires far more precision to select a topic than a finger-tap allows. It's high-unusable on a smartphone, but few online help systems have a better solution for navigation. As mobile browsing grows, more and more users will be frustrated by the lack of mobile accessibility in current help system designs.

## New opportunities

Mobiles can open up incredible new opportunities. Almost all smartphones now support HTML5 to some degree, and JavaScript libraries such as Dojo and jQuery<sup>1</sup> can offer sophisticated interactive content and interfaces. Imagine:

- Help systems that predict what a user is trying to do, delivering the solution intuitively rather than forcing the user to work out what they're looking for before they can find it.
- Instructions that adapt themselves to the user's exact task instead of using generic examples.
- 'Offline-able' help systems that stay accessible and functional even if the user loses their network connection.

These are just a few of the ways in which we can take advantage of the emerging mobile world. More intelligent help systems, more intuitive content delivery and better user experiences are where this technology is headed. It's time to stop thinking about mobile devices as the poor second cousin.

Mobile delivery is the future of the web and it's a radical change. Forget the drama of converting your frames and table layouts: mobile delivery requires a whole new approach to content creation.

## Considerations for mobile devices

### Reduced bandwidth

It takes much longer to download files to a mobile device than a desktop and Australian mobile users are typically subjected to much stricter download limits on mobile data plans. Content that contains a lot of images or uses large image files is going to frustrate a user who has to wait several minutes for the page to load. Unfortunately, most help documents today use large screenshots and diagrams to aid written instructions.

When writing for mobile devices, it's always best to ask yourself if you need the image in the first place – a well-written instruction will reach your user faster. Sometimes images can be vital to an explanation; the trick is to ensure that the resulting help files are kept as small as possible. For example:

- Keep your images as simple and clear as possible, and keep colours and unnecessary detail to a minimum. Not only does that make it easier for the user to understand the image, but images without gradients or lots of detail are usually smaller in file size.
- Crop the image down as much as possible, so you aren't showing the user anything they don't have to see.
- Shrink the image to the actual size it needs to be in the document using an image editing program rather than forcing the browser to reduce the full-size image.
- Downsample your images where possible to reduce size without a noticeable change to the image. Most image editing programs show you a preview of the resulting image during downsampling so you can balance quality and file size.
- Choose a file format that is supported by all browsers and provides good compression, such as GIF.
- Avoid PNG files, which don't play well with older versions of Internet Explorer, and JPG files, which lose quality each time the file is saved.

### Higher latency

I'm going to get technical here, but hold onto your keyboards, I'll keep it simple.

The truth is, it's not just that mobile devices have a slower download speed. It also takes them a lot longer to ask for the content to begin with.

Latency is how long it takes to get a message (like "hey, give me that file") from the web browser to the server and back again. Latency comes into play every time the web browser asks for a file, such as a web page or an image within that page. The latency of wired connections like desktops is negligible – so much so that it's generally more efficient to split a large page into smaller ones. That's why news articles, comments or image collections are often split into pages; with the minimal latency, it's faster for a wired desktop browser to request several small files than one big one.

But the latency on a mobile device is much, much higher – sometimes even higher than the actual download speed of the file. So you don't want that mobile sending individual requests for a hundred tiny files, because it ends up spending more time asking for the files than actually downloading them. On a mobile device, splitting up your content into separate pages can actually make it less usable. One big page will download faster than two dozen smaller ones.

### Reduced screen size

Mobile phones and tablets have a much smaller screen than desktops. You can fit, at best, a moderate-sized paragraph of information on the screen at one time. This means:

- Images need to be less than 450 x 280 pixels to avoid forcing users to scroll to see the rest of the image – or, even worse, forcing them to scroll horizontally to read your content.
- Large headings and fancy markup that chew up space need to be redesigned to be less obtrusive.
- Navigation needs to be hidden unless required by the user; the standard side frame of topic links eats up almost the entire mobile screen.
- Sentences and paragraphs need to be very concise. A paragraph a few lines long quickly becomes a wall of text on a small screen.
- The most pertinent information should be at the top of the page to reduce scrolling.
- Pages should be kept as short and specific as possible so users don't get 'lost' in the page while looking for their answer.

### Reduced attention

When people are sitting at a desktop reading your content, they're there to *read your content*. There aren't many distractions, except other programs or websites open on their computer. But mobile device users are almost always multitasking when they get to your site. Whether they're holding a conversation, shopping, trying to complete a task or even just keeping one eye out for

traffic while walking, there are far more things trying to pull their attention away.

Let's be real, here – nobody *enjoys* reading on a tiny screen: if they wanted a leisurely afternoon with your website, they'd dig out their laptop. A mobile user is rarely browsing; they have an aim in mind. Something has caused them to pause, pull out their phone and start looking up your information. They need to know something and they want to know it quickly so they can get back to what they were really doing. Hamper this goal at your peril. Your content needs to be:

- Concise and to-the-point
- Easily findable and navigable
- Clearly sign-posted so they know when they've found what they're looking for
- Written in clear, easily understood language.

You need to work with the notion that your reader wants to spend as little time as possible on your site and probably has an eye on something else at the same time. Balancing this with your business needs from the content (for example, marketing and sales) takes careful design.

### Web apps – the new way of the web

How do we reconcile the need for short, specific topics with the need to download fewer files to reduce the lag caused by latency? How do we create a navigation system that's there when we need it and hidden when we don't? Enter the new way of the web.

HTML5 opens up some amazing new capabilities, not just to solve these issues, but to revolutionise how we deliver content online. With JavaScript libraries such as Dojo we can compress an entire help system down to a single file – everything is downloaded in one hit onto the device. There's no more latency or waiting for pages to load because everything's already on the device when the help system starts.

We can create fully-functioning applications that run entirely in the phone's browser, can save themselves to the phone like a native app and even run offline when there's no internet connection available. Navigation systems can be designed in any number of ways to provide access to content efficiently (in both time and screen space) and aesthetically.

Help systems can evolve from mere static content to an interactive program that actively helps the user find information quickly and easily. We can even create a help system that interacts with the system it's documenting and start *really* blurring the line between telling a user how to do something and doing it with them. It's a whole new way of thinking, and a radical change in how we design online documentation.

<sup>1</sup> A library is a collection of source code modules use to extend the functionality of a program. By using a library, you avoid having to code every feature from scratch and can be reasonably assured the library code does not contain any bugs. Dojo is one such library for JavaScript, most notable for its extensive bank of user-interface and mobile-friendly modules.