

Responsive Design

Tuesday April 21st 2015 marked the start of what some doomsayers called “Mobigeddon”. On this day, Google released a major update to its mobile search algorithm which downgraded sites that were not mobile compatible, and boosted the rankings of websites that were mobile friendly.

Predictions of massive crashes in search traffic were generally over exaggerated (and generally spread by the spam emails of web design companies trying to sell mobile websites, but there was a general agreement that the impact on search traffic was noticeable and significant.

Between 40% and 60% of search traffic are said to come from mobile devices - figures vary depending on which source you consult, but the trend is definitely going in the direction of more mobile sessions and fewer desktop sessions.

So how do you make a website mobile friendly? There are 3 options:

1. Create a simple website with large text that flows across the screen
2. Create a separate website for mobile versions and use browser detection to redirect people
3. Create a website that is fluid in design and responds to the width of the browser window gracefully.

Option 2 was the most popular method up until 2013. Until this time, browser support for the more advanced features of CSS was not strong enough to allow websites to get away with it.

These days, nearly all browsers support CSS media queries which are the cornerstone of making a website responsive.

CSS Media Queries

Whats a CSS Media Query I hear you say? Well, in a nutshell, a media query is a special form of style sheet rule that allows you to specify the maximum and minimum browser size which the rules should operate in. Consider the following example:

```
<div id="leftdiv">this is on the left</div>
<div id="rightdiv">This is on the right</div>

<style>
#leftdiv{float:left;width:350px}
#rightdiv{float:left;width:350px}
</style>
```

This would produce a page that has the two lines of text next to each other like this:

this is on the left	This is on the right
---------------------	----------------------

If you viewed the page on a narrow screen, you could have text overlapping the edge of the screen. In some cases you can fix this by using `width:50%` instead of fixed width size, but if you have a table or image in one of the columns, this can cause problems.

What you can do in this case is to add extra rules in to make the page layout change depending on the size of the browser window. For example:

```
<div id="leftdiv">this is on the left</div>
<div id="rightdiv">This is on the right</div>

<style>

#leftdiv{float:none;width:100%}
#rightdiv{float:left;width:100%}

@media only screen and (min-width: 700px) {
    #leftdiv{float:left;width:350px}
    #rightdiv{float:left;width:350px}
}
</style>
```

So if you view this page in a browser wider than 700px you will see two columns of content, but resize the browser down below 700px wide and you will see one column of content.

Note that we put the mobile styles first then use “min-width: 700px” afterwards to cater for wider screen sizes. This method is called ‘mobile first’ responsive strategy. You could do it the other way round and make the stylesheets “desktop first”. The argument for using mobile first strategy is that it creates more streamlined code for mobile versions, and then the more complex css is handled by desktops. The advantage of doing desktop first is that you don’t need to add any hacks to make it work in IE8 (although few people use IE8 now anyway). It is generally considered good practice now to use ‘mobile first’ responsive layouts.

What widths should you design for?

It’s not really worth looking at the screen sizes of different devices when considering where to make your media size boundaries. The reason for this is that there are so many different device browser sizes, and even if you target the latest iPhone size, there’s no telling when a ultra HD display device might come along and change things. Therefore you should set your design widths according to how your individual design looks. Make sure you consider all pages on your site too. Some pages might have large tables of data which are difficult to resize down, or they may have detailed images which need to be viewed in the largest possible size.

Summary

These days, all websites should be mobile friendly (with a few exceptions for complex admin systems that are not appropriate for mobile delivery). The best way to achieve mobile friendly websites it to use mobile first responsive web design.