

Responsive design

Writing media queries (MQ)

Meta viewport tag

A responsive web page needs to have a <meta> tag in the head of the document that forces the device to report its actual width. This is because there is a possibility that a mobile device will zoom out and therefore display the desktop version. This is also true where a web page is fully responsive without using media query breakpoints.

The meta tag can be written like this:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This will make the device display the web page at the correct size and allow the user to zoom in if needed.

Another way can be like this:

```
<meta name=viewport content="width=device-width, initial-scale=1.0, maximum-scale=1.0">
```

This will prevent a user from zooming in because maximum scale is set to 1.0. Adding a greater value here can control the amount of zoom e.g. 1.5.

How MQ are expressed

Media Queries define a set of CSS to be used for a screen of a particular size. They are written within a CSS document and look something like this:

```
@media screen and (max-width: 750px) {  
  
}
```

Or like this:

```
@media screen and (min-width: 700px) and (max-width: 959px) {  
  
}
```

The media query contains a statement regarding a screen width (breakpoints). Breakpoints can be stated as minimum width, maximum width or both.

The MQ also comes with its own set of braces - `{ }`. These braces contain the CSS for the media query like so:

```
@media screen and (max-width: 750px) {  
  
  body {  
    font-size: 20px;  
    line-height: 26px;  
    background-color: #feffca;  
  }  
  
}
```

Methods for setting up the CSS in responsive design

Example 1

Media queries within a single CSS document

A common method of using MQ is to place them within a single CSS document. Using one CSS in this way also helps to reduce the number of HTTP (link) requests from the HTML document.

'Base' styles are the first to appear in the CSS. These are styles entered in the normal way and will determine the foundation for the appearance of the HTML and the cascade for the MQ that are to follow. The base styles do not need a media query.

The CSS for the MQ are placed after the base styles in order of size, smallest first. The CSS will look something like this:

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 16px;
    line-height: 22px;
    background-color: #FFF;
}
#wrapper {
    max-width: 940px;
    margin-left: auto;
    margin-right: auto;
    padding: 20px; /* prevents content going up to edge on mobile device */
}

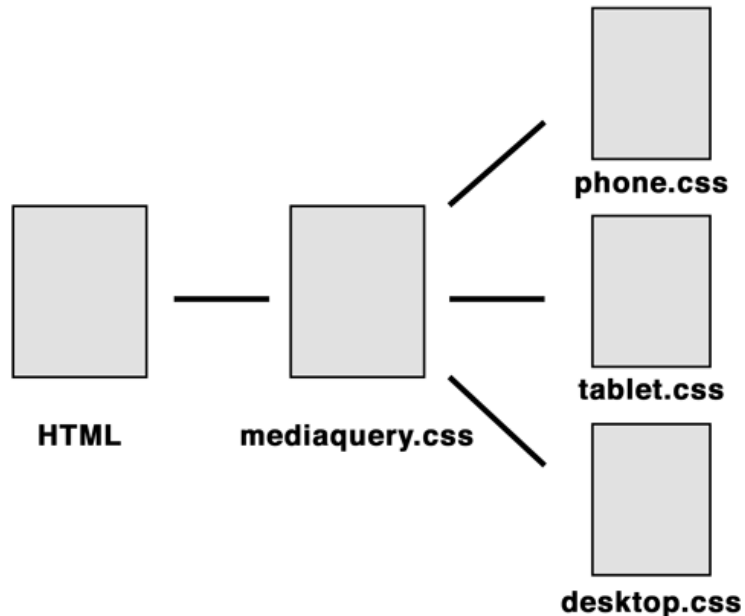
@media screen and (min-width: 700px) and (max-width: 959px) {
    body {
        font-size: 20px;
        line-height: 26px;
        background-color: #feffca;
    }
}
```

```
@media screen and (min-width: 960px) {  
  body {  
    font-size: 24px;  
    line-height: 32px;  
    color: #009;  
  }  
}
```

Example 2

Within a media query CSS document

A CSS document can be created that contains media queries with links to separate CSS documents. The document with the links is referred to as a 'mediaquery.css'. The separate CSS documents contain the CSS relevant for the MQ breakpoints.



The links in the mediaquery.css look like this:

```
@import url("phone.css") only screen and (max-width:480px);
```

```
@import url("tablet.css") only screen and (min-width:481px) and (max-width:959px);
```

```
@import url("desktop.css") only screen and (min-width:960px);
```

In the head of the HTML document the link to the mediaquery.css document looks like this:

```
<link href="css/mediaquery.css" rel="stylesheet" type="text/css">
```

Example 3

Using separate CSS documents

These examples show how the references to the CSS are written in the head of the HTML document.

This example links to a main/global CSS and a media query CSS. The breakpoints are written inside the mediaquery.css:

```
<link href="css/main.css" rel="stylesheet" type="text/css" media="screen">
<link href="css/mediaquery.css" rel="stylesheet" type="text/css">
```

This example uses a main/global CSS and a media query CSS where the breakpoints are written as attributes in the link to the file.

```
<link rel="stylesheet" href="css/base.css">
<link rel="stylesheet" media="screen and (min-width: 960px)" href="css/desktop.css" />
```

or

```
<link rel="stylesheet" href="css/base.css">
<link rel="stylesheet" media="screen and (min-width: 700px) and (max-width: 959px)"
href="css/tablet.css" />
<link rel="stylesheet" media="screen and (min-width: 960px)" href="css/desktop.css" />
```

With the above examples, the main/base CSS needs to be referenced first in the code, followed by the breakpoint CSS.

The Cascade

Responsive web design makes much use of the cascade in CSS. For instance, in *example 1* the *base* styles will influence the MQ in the styles that follow. This is because the base styles do not have breakpoints defined. Therefore they will be effective at all screen widths. This is useful for brevity in the CSS document because the base styles do not need to be redefined within the MQ breakpoint CSS. If the designer wants something different in the MQ breakpoint CSS than that defined in the base styles, then only the specific property for the style needs to be stated within the breakpoint. The following illustrates this:

```
/* base styles */
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 16px;
    line-height: 22px;
    background-color: #FFF;
}

@media screen and (min-width: 700px) {
    body {
        font-size: 20px;
        line-height: 26px;
    }
}
```

Here, a different font size is stated in the *body* style for widths of 700px and over. The font family and background colour, as stated in the base styles, do not need to be written again in the MQ as they will still be effective.

The Cascade and min/max-width breakpoints

The cascade can be controlled by the use of *min-width* and *max-width* breakpoints in media queries. Think of the min/max-width breakpoints as switches that limit the effective range of the CSS:

A *min-width* breakpoint prevents those styles from being effective in sizes smaller than stated.

A *max-width* breakpoint prevents those styles from being effective in sizes larger than stated.

When *min-width* breakpoints are used in media queries, those styles will cascade into subsequent MQ in the CSS.

When *max-width* declarations are used in media queries, those styles will not cascade into subsequent MQ in the CSS.

In practice, media queries are written with min-width breakpoints, combined with MQ that state min and max-width breakpoints. There is no hard and fast rule; it depends on the requirements of the design.