

WDD 131

[Home](#)[W1](#)[W2](#)[W3](#)[W4](#)[W5](#)[W6](#)[W7](#)

CSS Media Queries

Overview

CSS media queries let you apply different styles to your website based on the characteristics of the device being used to view it. Media queries allow the website to "respond" and adapt to different screen sizes, orientations, and even user preferences.

► Associated Course Learning Outcome(s)

Prepare

CSS media queries are essential to responsive web design. The `@media` at-rule specifies condition(s) to create a block of CSS rules that are applied if the condition(s) evaluates to true. This is extremely useful as selected elements can be repositioned, resized, hidden, exposed, etc. based upon the user's viewport size.

Here is the general syntax is setting up a CSS media query:

```
@media not|only mediatype and (expressions) {  
    /* CSS rules go here inside the @media query's opening and closing curly brace */  
}
```

Here is an example CSS media query:

```
1 @media screen and (min-width: 640px) {  
2     h1 {  
3         font-size: 2.5rem;  
4         margin: 1rem;  
5         color: navy;  
6     }  
7 }
```

Code explanation

- 1 The `@media screen and (min-width: 640px) {` is the signature line of the `@media` query block of CSS rules to be applied if the conditions are met. In this case the conditions are if the media type is a screen and the viewport minimum width is at least 640 pixels wide.
- 2 The heading `h1 {` **<h1> element selector** which starts the defined **CSS rule**.
- 3 - 5 The `@font-size: 2.5rem; margin: 1rem; color: navy;` are **declarations** to be applied to any `<h1>` elements if the `@media` query conditions are met.
- 6 The first `}` closes the CSS rule for `<h1>` code elements.
- 7 The last `}` closes the media query.

Note that curly brackets `{ }` are used to **contain** a specific media query and are also used to define CSS Rules. A common mistake is to close the `@media` query container too soon or not at all. Using the automatic indentation feature of VS Code document formatting features will help you recognize issues with your CSS syntax.

Demonstration Video:  [Media Queries](#) [2:54 minutes]

Activity Instructions

For this activity, you will create a simple HTML page with two CSS files. You will use CSS media queries to apply the appropriate CSS file based upon the viewport width.

A responsible use of an AI generative tool could be to ask questions about how to formulate a specific code piece. For example, "how to combine selectors in CSS", and an example will be given with which you can apply to your own code.

Step 1: File and Folder Setup

1. Add a new HTML file named "**media-query.html**" in a "**week02**" named folder in your **wdd131** repository folder.
2. Add two CSS files named "**media-query.css**" and "**media-query-large.css**" to a subfolder named "**styles**" within the **week02** folder.

Step 2: HTML

1. In your **media-query.html** file, create a valid HTML page with standard **head** content including
 - ✓ Meta Charset Attribute
 - ✓ Meta Viewport Element
 - ✓ Title Element
 - ✓ Meta Description Element
 - ✓ Meta Author Element
 - ✓ Link to a Google Font named "Roboto" using the contemporary code embed provided by fonts.google.com
 - ✓ Link references to your CSS two (2) files.

► Example

2. In the **body** of the HTML document, add a **header** with an **h1**, a **main** element with two **section** elements, and a **footer** element.

1. The h1 element should contain the words "Media Query Example".
2. Each section h2 heading contains a scripture with book, chapter, and verse.
3. The section paragraph contains a copy of the actual scripture referenced in the heading.
4. The footer should contain your name.

► Example

Step 3: CSS

1. Style the document as shown in the example screenshots given below.

1. **media-query.css**

- ✓ Do **not** include a media query.
- ✓ Use the Google Font - Roboto in the **body** rule.
- ✓ The **header**, **main**, and **footer** each
 1. have a maximum width of 640 pixels,
 2. are centered on the page using **margin: 1rem auto**,
 3. include a faint border,

4. use appropriate padding, and
5. have a blueish background color of your choice.

Remember that you can combine these three selectors into one rule or create a class that contains these common CSS declarations.

- ✓ Declare the main element to be a CSS grid with a grid template of only one column, and a equal gap of 1rem, and

If you had a question to pose to the class in Microsoft Teams, use specific and descriptive language, e.g., *"I have a question about the media-query assignment part 5.1.4. I do not know how to set the main element grid to only one column with an even gap of 1rem."*

- ✓ Declare the section elements to also have padding and a lighter or white background.

► Check Your Understanding

2. In the **media-query-large.css**,

1. Write a containing media query to be applied at a viewport width of 500px or greater.

► Check Your Understanding

2. Change through declaration within the media query the **header** to a black background with white text, and
3. Change the **main** element to display two columns of equal size.

► Check Your Understanding

Example Screenshots

Media Query Example

2 Nephi 32:9

But behold, I say unto you that ye must pray always, and not faint; that ye must not perform any thing unto the Lord save in the first place ye shall pray unto the Father in the name of Christ, that he will consecrate thy performance unto thee, that thy performance may be for the welfare of thy soul.

Helaman 12:24

24 And may God grant, in his great fulness, that men might be brought unto repentance and good works, that they might be restored unto grace for grace, according to their works.

Rubia M. Francesco

Phone View Screenshot

Media Query Example

2 Nephi 32:9

But behold, I say unto you that ye must pray always, and not faint; that ye must not perform any thing unto the Lord save in the first place ye shall pray unto the Father in the name of Christ, that he will consecrate thy performance unto thee, that thy performance may be for the welfare of thy soul.

Helaman 12:24

24 And may God grant, in his great fulness, that men might be brought unto repentance and good works, that they might be restored unto grace for grace, according to their works.

Rubia M. Francesco

Wider View Screenshot

► Check Your Understanding

Setting Breakpoints

Breakpoints should be set to support the design and content of the page. There are no set breakpoints to use.

"Do NOT define breakpoints based on device classes. Defining breakpoints based on specific devices, products, brand names, or operating systems that are in use today can result in a maintenance nightmare. Instead, the content itself should determine how the layout adjusts to its container."

[How to choose breakpoints - Responsive Web](#) - Google Web Fundamentals

Testing

You should be continuously checking your browser rendered work using [Live Server](#) extension.

1. Test your work by resizing your browser window or by using [DevTools device settings](#) to see the changes in the layout.
2. Use the browser's DevTool [CSS Overview](#) to check color contrast and other CSS your other usage.
3. You will need to commit and push your work to your GitHub Pages enabled wdd131 site in order to use the [Page Audit Tool](#).

Copyright © Brigham Young University-Idaho | All rights reserved