

Questions 4 – Responsive Web Design

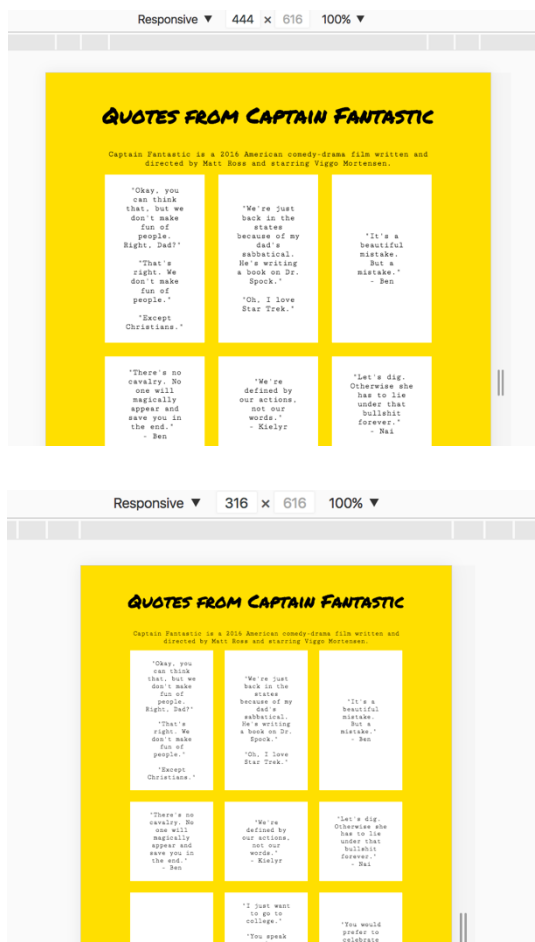
1. Study how to use the media queries. Make very simple web page with only few elements and change their background color or size according to viewport size. Select three different sizes. With Firefox or Chrome browsers responsivity is easy to test by selecting developer tools.

Answer in responsive-task1.html

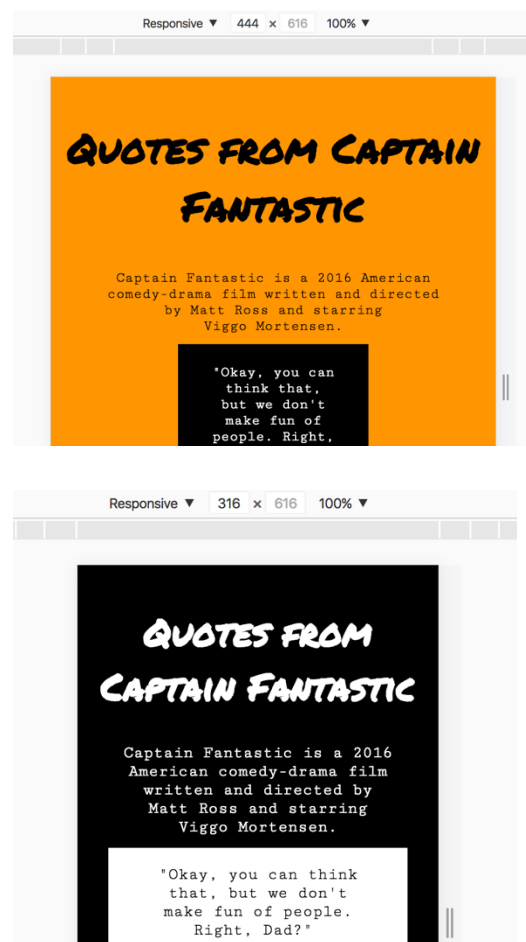
2. Test how the viewport meta tag affects your page

Adding the viewport meta tag to my responsive.html page didn't change anything on Firefox as the media queries were already working just fine and the page was adjusting to the screen size. However, on Chrome, adding the tag made a huge difference as the media queries didn't work and the page was rendered at desktop screen width, just scaled to fit the screen.

Before adding the viewport tag



After adding the viewport tag



3. What is CSS3's display property? What kind of use it has on responsive web pages?

Every element on a web page is a rectangular box and CSS's display property specifies how this box behaves. The property has over dozen possible values but here are a few commonly used ones are inline, inline-block, block and none.

- **Inline:** Default value. Displays an element as an inline element (like ``)
- **Block:** Displays an element as a block element (like `<p>`), with a line break in the end of element.
- **Inline-block:** Displays an element as an inline-level block container. The inside of this block is formatted as block-level box, and the element itself is formatted as an inline-level box.
- **None:** The element will not be displayed at all (and hence has no effect on layout)

In responsive layouts the display property can be utilized for example to control whether certain elements are shown side by side or if they take up each on their own line. To achieve this a group of elements can be displayed as inline-blocks on a larger viewports and as block elements on a smaller viewport. Display property is also useful for creating for example different type of menus for different viewport sizes. While menu bar might look nice on a larger screen, on a smaller device, it might be better to hide the menu behind a button. The display property makes hiding and revealing different elements easy.

4. Study how vh, vw, vmin and vmax work on element sizing. Especially use these on some element's font size definition. How does it behave when viewport size increases and decreases? Is it good idea to have fully responsive font size?

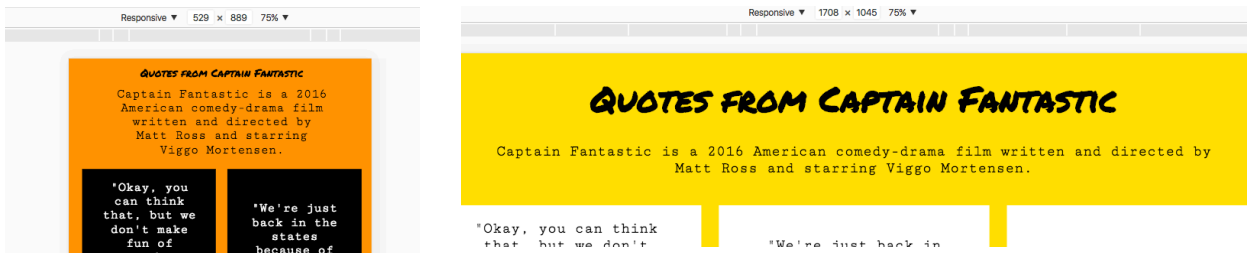
Length units vh, vw, vmin and vmax are CSS length units relative to the viewport size of the browser or device.

- 1vw = 1% of viewport width
- 1vh = 1% of viewport height
- 1vmin = 1vw or 1vh, whichever is smaller
- 1vmax = 1vw or 1vh, whichever is larger

Using these units in element sizing means that the element sizes change in relation to viewport's height or width, depending on the unit used. As the size of an element is a percentage of viewport's dimension, the results are obviously very different on different viewport sizes. On 1600 pixels wide viewport 1vw is 16px, while on 300 pixels wide viewport 1vw is only 3px.

My experience when testing these units was that finding especially an appropriate font-size takes a lot more time and experimentation than when using for example pixels as the unit of

measurement. While size of title tied to viewport's width might look really good on wider screen, on for example mobile device, the title will be ridiculously small. (Thankfully, the units vmin and vmax provided a bit more consistent results.)



Sizing other elements (such as the div-elements holding the quotes on my example page), is far easier as it doesn't really differ from using percentages which I'm personally fairly comfortable with. However, with font-sizes, using only viewport related units requires quite a bit more work. To ensure good user experience for every single user requires so much more testing, that I would not recommend using fully responsive font-sizes on any larger pages.

(Testing I did for these exercise can be found in file responsive-task4.html)

5. Use responsive page design principles on the web page you made earlier. Some elements maybe tricky to handle, like for example HTML tables. There is nice trick how to make table elements responsive, see <https://css-tricks.com/responsive-data-tables/>

Answer consists of the following files:

- responsive-task5.html
- responsive-task5.css
 - The "normal" styling for the page
- responsive-task5-adjustments.css
 - Adjustments made to the page to make it better work on different screen sizes
- resources-folder

Sources

- CSS-Tricks. Viewport Sized Typography [article]. Retrieved 2.4.2017 from <https://css-tricks.com/viewport-sized-typography/>
- W3School. CSS Display Property [online tutorial]. Retrieved 22.3.2017 from https://www.w3schools.com/cssref/pr_class_display.asp
- CSS-Tricks. Display [article]. Retrieved 22.3.2017 from <https://css-tricks.com/almanac/properties/d/display/>