

LEARN WEB

ale66

RESPONSIVE

SO FAR

A page is *responsive* when it adapts to the device (viewport) it runs on to offer a specific experience which is adequate to the context.

We might forfeit some content for the sake of a clean appearance

The current trend in design is *mobile first*.

The simple **flexbox** option in the CSS make boxes *responsive*

THE BASIC SETUP: CSS

```
1 * {  
2   box-sizing: border-box;  
3 }
```

it applies to *all elements*

boxes are forced in their assigned dimension, which thus remain fixed

THE BASIC SETUP: HTML

```
1 <head>
2   <meta name = "viewport"
3       content = "width=device-width, initial-scale=1.0">
4   ...
5 </head>
```

Thanks to these we can set dimensions as percentages

Fix max (or min) dimension and let the browser adjust the rest

Default: circa 980px, a desktop screen. So on smaller screens everything would look tiny

Example: 200 or 250?

```
1 div {  
2   width: 200px;  
3   padding: 20px;  
4   border: 5px solid black;  
5 }
```

without **border-box**, it's 250...

```
1 div {  
2   width: 50%;  
3   padding: 20px;  
4   border: 5px solid black;  
5 }
```

will always take up half of the screen, whatever the padding

MEDIA QUERIES, 1

```
1  /* don't normally show these elements */  
2  .quotation {  
3      display: none;  
4      padding-right: 1.67%;  
5  }
```


MEDIA QUERIES, 2

On wide screens start showing them:

```
1 @media (min-width: 750px) {  
2     /* ok, it's a big screen:  
3         show quotations */  
4     .quotation {  
5         display: block;  
6     }  
7 }
```

Try it out on the [wdpg/13-5-1](#)

DEFINITIONS

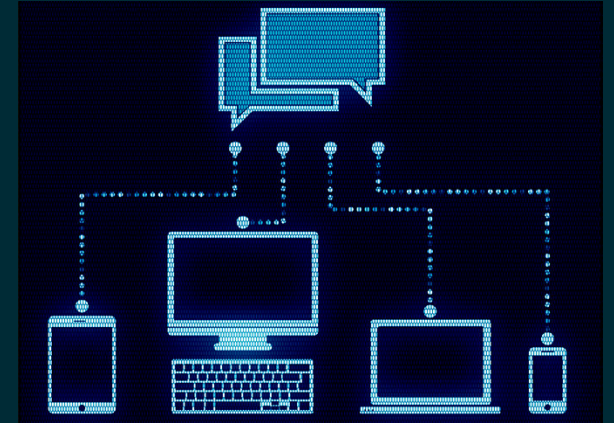
the size of an image is

<no. of pix. rows> x <no. of pix. cols>

Example: look at `./imgs/webcomm-cover.jpg`

The aspect ratio:

no. of pixel columns : no. of pixel rows



ACTIVITY: IMAGE PREPARATION

Resize images

Win: right-click on `wdp-cover.png` to open with **Photos**

- click on `...` then **Resize**
- what is the effect of increasing pixelage?
- and of altering aspect ratio?

ISSUES WITH IMAGES

scaling an image beyond its original size reveals pixelation and ugly edges

Scaling must be capped and not follow browser resizing.

MAKING IMAGES *FLUID*

```
1  img {  
2      max-width: 100%;  
3      height: auto;  
4  }
```

Expansion is capped to the original dimensions

shrinkage not so

► Example

➡ Online: wdpg.io/14-1-1

This code creates a fluid image that scales smaller or larger as the screen size changes but doesn't scale larger than its original dimensions.

CSS

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

The rule that makes
images fluid

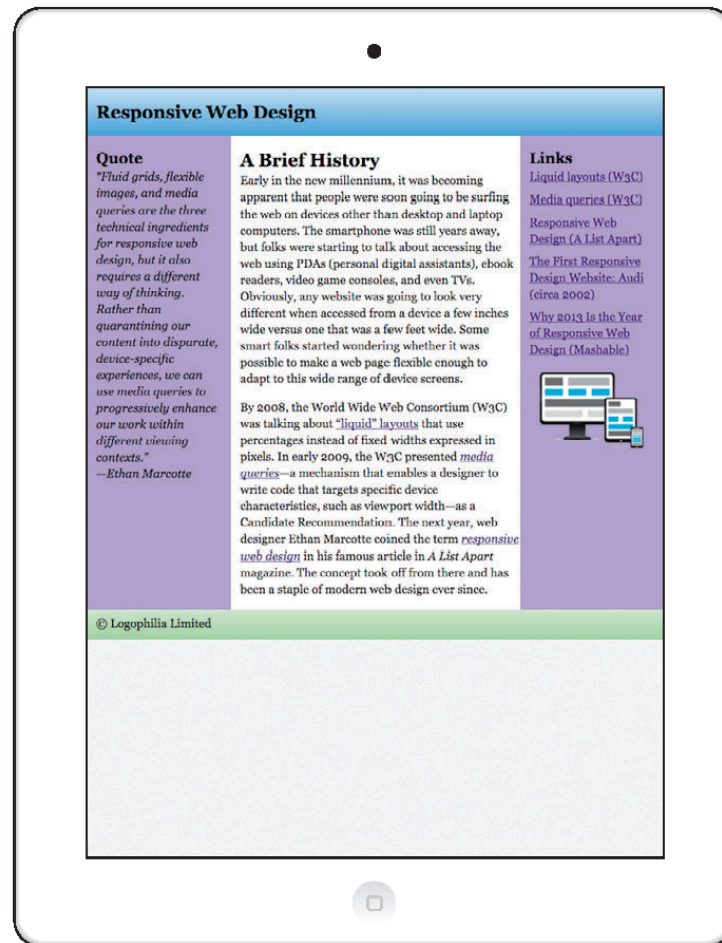
HTML

```
<header>  
  <h1>Responsive Web Design</h1>  
</header>  
<main>  
  <aside class="quotation">  
    <h3>Quote</h3>  
    etc.  
  </aside>  
  <article>  
    <h2>A Brief History</h2>  
    <p>Early in the new millennium, etc.</p>  
  </article>  
  <aside>  
    <h3>Links</h3>  
    etc.  
      
  </aside>  
</main>  
<footer>  
  <p>&copy; Logophilia Limited</p>  
</footer>
```

An image added to
the aside element



► **Figure 14.1** The image as it appears when its `aside` parent element is given the full width of a smartphone screen



► **Figure 14.2** When the `aside` element is displayed at a narrower width, the image scales down accordingly.

Try it on the WDPG playground!

github.com/ale66/learn-web

EXERCISE 14-1

Take 10 minutes to work on [exercise 14-1](#)

As soon as you're done think how you can transfer this to your in-class project: is it worth?

HAVING MULTIPLE SIZES, 1

May create professional-looking, responsive pages

► Example

⇒ Online: wdpg.io/14-2-1

This example uses the tag's sizes and srcset attributes to deliver an image responsively based on the browser viewport size.

HTML

```

```

The default image
for older browsers

The sizes to
display the image

The images that
the browser can
choose among

HAVING MULTIPLE SIZES, 2

► Example

⇒ Online: wdpg.io/14-2-1

This example uses the `` tag's `sizes` and `srcset` attributes to deliver an image responsively based on the browser viewport size.

HTML

```

```

The default image
for older browsers

The sizes to
display the image

The images that
the browser can
choose among

Best for corporate sites with images in a database etc.
Implementation is well-known to be error-prone.

EXERCISE 14-2

work on [exercise 14-2](#)

TYPOGRAPHY

TYPOGRAPHY: STATIC

HTML:

```
<body>
  <header>
    <h1>What's My Font Size?</h1>
  </header>
</body>
```

CSS:

```
body {
  font-size: 1em;
}
header {
  font-size: 1.5em;
}
h1 {
  font-size: 2em;
}
```

What if font sizes change to adapt to viewport size?

RELATIVE UNITS

rem: *relative* units of measure

```
1 html {  
2   font-size: 20px;  
3 }
```

Now 1rem=20px, 1.5em=30px and so on *for the whole page*

By default, browsers set **font-size: 16px**; so 1rem = 16px

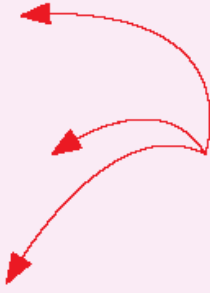
Vice versa, **em** is relative to the font size in the specific *container* we're in, not the whole document

TYPOGRAPHY: DYNAMIC

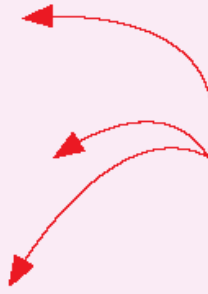
CSS

```
h1 {  
  font-size: 2rem;  
}  
h2 {  
  font-size: 1.5rem;  
}  
h3 {  
  font-size: 1.25rem;  
}  
@media (min-width: 750px) {  
  h1 {  
    font-size: 2.5rem;  
  }  
  h2 {  
    font-size: 2rem;  
  }  
  h3 {  
    font-size: 1.5rem;  
  }  
}
```

The header elements are given mobile-first rem font sizes.



The header elements are also given large-screen rem font sizes.



ISSUES, 1

```
HTML:
<header>
  <h1>Responsive Web Design</h1>
</header>
CSS:
header {
  height: 64px;
}
h1 {
  font-size: 2rem;
}
```

Responsive Web Design

► **Figure 14.6** The `h1` text looks good at `2rem`.

the **header** container is fixed at 64px!

rem alone won't do


```
HTML:
<header>
  <h1>Responsive Web Design</h1>
</header>
CSS:
header {
  height: 64px;
}
h1 {
  font-size: 2rem;
}
```

Responsive Web Design

► **Figure 14.6** The `h1` text looks good at `2rem`.

Responsive Web Design

► **Figure 14.7** The element doesn't render so well when a larger default font is used.

SOLUTION

Responsive Web Design

► **Figure 14.8** With the `header` element's `height` property now using relative `rem` units, the `header` scales along with the text as the default font size changes.

The only change I made was to declare `height: 4rem` on the `header` element. Using the relative unit makes the height responsive, so it increases (or decreases) along with the font size when the default font value is changed.

EXERCISE 14-3

Take 5/8 minutes to work on [exercise 14-3](#)

A CHEATSHEET

- For vertical measures such as `padding-top`, `padding-bottom`, `margin-top`, and `margin-bottom`, use `rem` units.
- For horizontal measures such as `width`, `padding-right`, `padding-left`, `margin-right`, and `margin-left`, use percentages.
- For horizontal measures in which you want more control of properties such as `width`, `max-width`, and `min-width`, use `rem` units.
- For vertical measures that you want to scale in relation to the viewport height, use `vh` units.
- For horizontal measures that you want to scale in relation to the viewport width, use `vw` units.

- For vertical measures such as `padding-top`, `padding-bottom`, `margin-top`, and `margin-bottom`, use `rem` units.
- For horizontal measures such as `width`, `padding-right`, `padding-left`, `margin-right`, and `margin-left`, use percentages.
- For horizontal measures in which you want more control of properties such as `width`, `max-width`, and `min-width`, use `rem` units.
- For vertical measures that you want to scale in relation to the viewport height, use `vh` units.
- For horizontal measures that you want to scale in relation to the viewport width, use `vw` units.

work on your `mystyle.css` from the COM page to include all setups

bring it with you across projects

EXERCISE 14-4

A bit labour-intensive Take 10 minutes to work on
exercise 14-4

GALLERY OF RESPONSIVE SITES

try stressing them, turning the phone, etc.

hicksdesign.co.uk

www.anderssonwise.com

www.bostonglobe.com

authenticjobs.com

CONCLUSIONS

Summary

- Make your images fluid by styling them with the declarations `max-width: 100%` and `height: auto`.
- In your `` tags, add the `sizes` and `srcset` attributes to scale and deliver images that are appropriate for any screen size.
- When styling font sizes, avoid absolute pixel values in favor of `rem` units.
- Also use `rem` units when styling vertical measures such as height, padding, and margins.