Making Your Images and Typography Responsive

The most important thing about responsive design is flexibility. Everything must be flexible: layouts, image sizes, text blocks—absolutely everything. Flexibility gives your site the fluidity it needs to fit inside any container. —Nick Babich

This chapter covers

- Making fluid images that respond to screen size
- Delivering different-size images based on the user's screen size
- Making text adapt to the screen size by specifying responsive font sizes
- Making other page elements adapt to the screen size by specifying responsive measurements

In Chapter 13, you learned not only why you shouldn't use a fixed-width layout, but also why (and how) you should use responsive layouts that are both flexible and adaptive. Having your page layout change in response to different screen widths is a must in these days of wildly different screen sizes, but it's only part of the total responsive package. To make your pages truly adaptable to any device, you need to sprinkle both your page images and page typography with responsive pixie dust. You need to style images to scale up or down depending on the screen width, deliver different images based on the screen size, and use responsive type sizes. You learn these and other powerful responsive techniques in this chapter.

Making Images Responsive

Making an image responsive is one of the biggest challenges that web designers face. The scale of the challenge comes from two problems associated with making images responsive:

- Making a fixed-size image fit into a container with fluid dimensions. An image that's 600 pixels wide will fit nicely inside an element that's 800 pixels wide, but it overflows if that element is scaled down to 400 pixels wide. Solving this problem requires making images fluid so that the size adjusts to the changing screen size.
- Delivering a version of an image that's sized appropriately for the user's screen dimensions. It's one thing to offer up a 2,000 x 1,500-pixel image to desktop users, but sending the same image to smartphone users is a waste of upload time and bandwidth.

The next two lessons show you some basic methods for overcoming these problems.

Lesson 14.1: ***Creating Fluid Images***

Covers: Styling the `img` element for responsiveness

Online: wdpg.io/14-1-0

An image comes with a predetermined width and height, so at first blush, it seems impossible to overcome these fixed dimensions. Fortunately, an `<img>` tag is another page element. Yes, by default the image is displayed at its full width and height, like a `div` or any other block element. But in the same way that you can make a block element fluid by using percentages, you can make an image fluid.

You need to be a bit careful when working with images:

- In most cases, you don't want the image to scale larger than its original size since, for most images, this scaling will result in ugly pixilation and jagged edges.
- If you change one dimension of an image, it will almost certainly appear to be skewed because its original aspect ratio—the ratio of the width to the height of the image—will have been altered. Therefore, you have to change both the width and the height proportionally to retain the image's original aspect ratio. Fortunately, you can get the browser to do some of the work for you.

To handle both concerns, you can create a fluid image that responds to changes in screen size by applying the following rule:

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

Setting `max-width: 100%` allows the image to scale smaller or larger as its parent container changes size but also specifies that the image can never scale larger than its original width. Setting `height: auto` tells the browser to maintain the image's original aspect ratio by calculating the height based on the image's current width.

The following code shows an example.

PLAY

In some cases, you don't want the image height to scale larger than its original height, so you need to set `max-height: 100%` *and* `width: auto` *on the image.* ⇨ **Online:** wdpg.io/14-1-2

▶ *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;
}
```

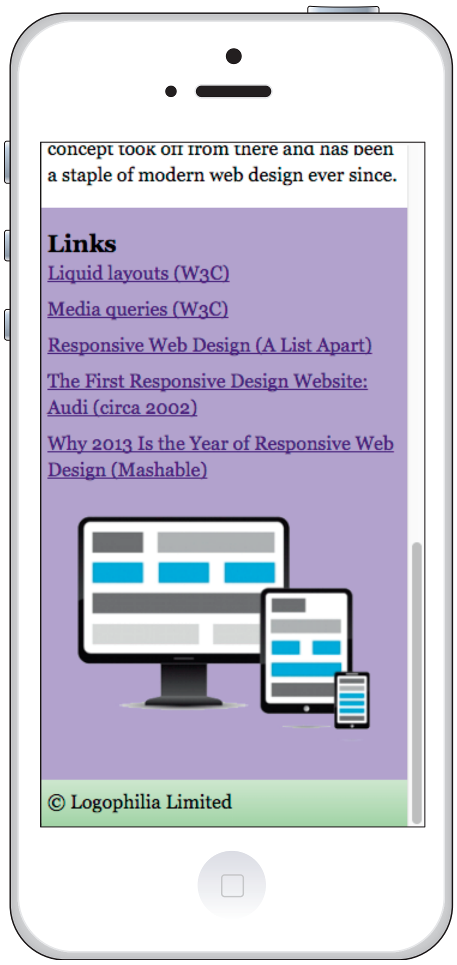
The rule that makes images fluid

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.
        <img src="/images/rwd.tif" alt="Responsive Web Design
image">
    </aside>
</main>
<footer>
    <p>&copy; Logophilia Limited</p>
</footer>
```

An image added to the aside element

Figures 14.1 and 14.2 show how the image size changes as the width of its parent `aside` element changes.



concept took off from there and has been a staple of modern web design ever since.

Links

Liquid layouts (W3C)

Media queries (W3C)

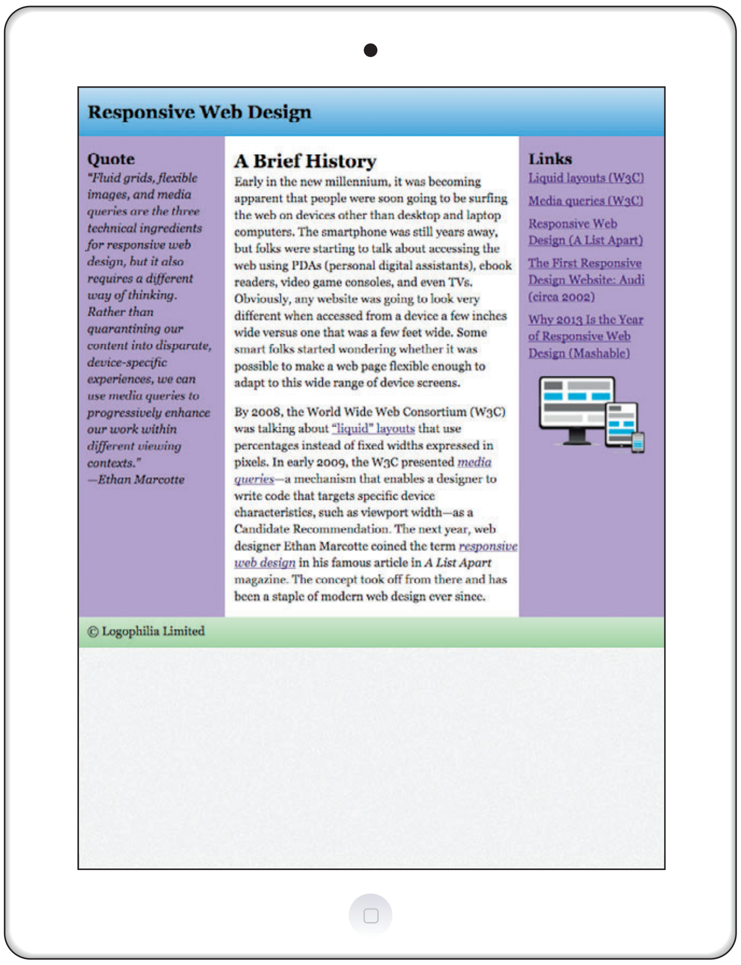
Responsive Web Design (A List Apart)

The First Responsive Design Website: Audi (circa 2002)

Why 2013 Is the Year of Responsive Web Design (Mashable)

© Logophilia Limited

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



Responsive Web Design

Quote

"Fluid grids, flexible images, and media queries are the three technical ingredients for responsive web design, but it also requires a different way of thinking. Rather than quarantining our content into disparate, device-specific experiences, we can use media queries to progressively enhance our work within different viewing contexts."
—Ethan Marcotte

A Brief History

Early in the new millennium, it was becoming apparent that people were soon going to be surfing the web on devices other than desktop and laptop computers. The smartphone was still years away, but folks were starting to talk about accessing the web using PDAs (personal digital assistants), ebook readers, video game consoles, and even TVs. Obviously, any website was going to look very different when accessed from a device a few inches wide versus one that was a few feet wide. Some smart folks started wondering whether it was possible to make a web page flexible enough to adapt to this wide range of device screens.

By 2008, the World Wide Web Consortium (W3C) was talking about "liquid" layouts that use percentages instead of fixed widths expressed in pixels. In early 2009, the W3C presented *media queries*—a mechanism that enables a designer to write code that targets specific device characteristics, such as viewport width—as a Candidate Recommendation. The next year, web designer Ethan Marcotte coined the term *responsive web design* in his famous article in *A List Apart* magazine. The concept took off from there and has been a staple of modern web design ever since.

Links

Liquid layouts (W3C)

Media queries (W3C)

Responsive Web Design (A List Apart)

The First Responsive Design Website: Audi (circa 2002)

Why 2013 Is the Year of Responsive Web Design (Mashable)

© Logophilia Limited

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

Lesson 14.2: ***Delivering Images Responsively***

Covers: The `sizes` and `srcset` attributes

Online: wdpg.io/14-2-0

The other side of the responsive-image coin involves delivering to the user a version of the image that has a size that's appropriate for the device screen. You might deliver a small version of the image for smartphone screens, a medium version for tablets, and a large version for desktops. In the past, you needed a script to handle this task, but in HTML5, you can do everything right in your `<img>` tag thanks to two new attributes: `sizes` and `srcset`.

The `sizes` attribute is similar to a media query in that you use an expression to specify a screen feature, such as a minimum or maximum height, and then specify how wide you want the image to be displayed on screens that match that configuration. You can specify multiple expression-width pairs, separated by commas. Here's the general syntax:

```
sizes="(expression1) width1,
       (expression2) width2,
       etc.,
       widthN"
```

Notice that if the last item doesn't specify an expression, the specified width applies to any screen that doesn't match any of the expressions. Suppose that you want images to be displayed with width 90vw on screens that are less than or equal to 500px and 50vw on all other screens. Here's how you'd set that up:

```
sizes="(max-width: 500px) 90vw, 50vw"
```

Next, add to your `<img>` tag the `srcset` attribute, which you set to a comma-separated list of image file locations, each followed by the image width and letter `w`. Here's the general syntax:

```
srcset="location1 width1w,
        location2 width2w,
        etc.">
```

This code gives the browser a choice of image sizes, and it picks the best one based on the current device’s screen dimensions and the preferred widths you specified in the `sizes` attribute. Here's an example:

```
srcset="/images/small.tif 400w,
        /images/medium.tif 800w,
        /images/large.tif 1200w">
```

The following example puts everything together to show you how to deliver images responsively.

BEWARE

When you're testing the `srcset` attribute by changing the browser window size, you may find that the browser doesn't always download a different-size image. Although the browser may detect that a smaller image should be used based on the `srcset` values, it may opt to resize the existing image, because it has already downloaded that image.

REMEMBER

The default image—that is, the image specified with the `src` attribute—is the fallback image that will be displayed in older browsers that don't support the `srcset` attribute. Good mobile-first practice is to make the default image the one you prefer to deliver to mobile users.

▶ Example

Online: wdpg.io/14-2-1

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

HTML

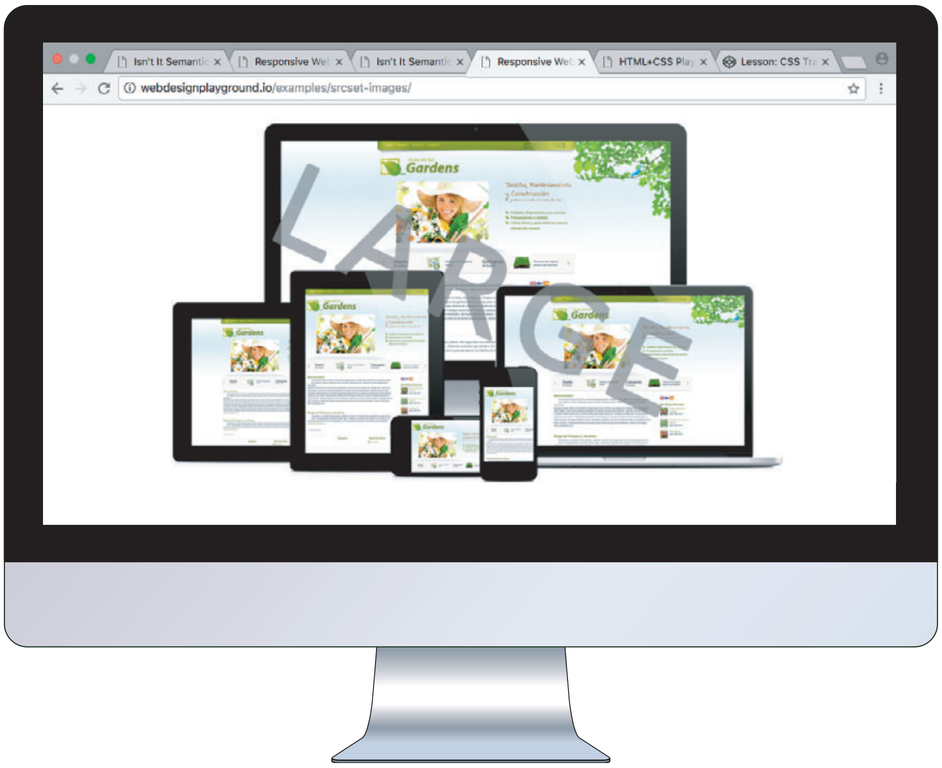
```
<img
   src="/images/img-small.tif"
   sizes="(max-width: 700px) 100vw, 75vw"
   srcset="/images/img-small.tif 450w,
         /images/img-medium.tif 900w,
         /images/img-large.tif 1350w">
```

The default image for older browsers

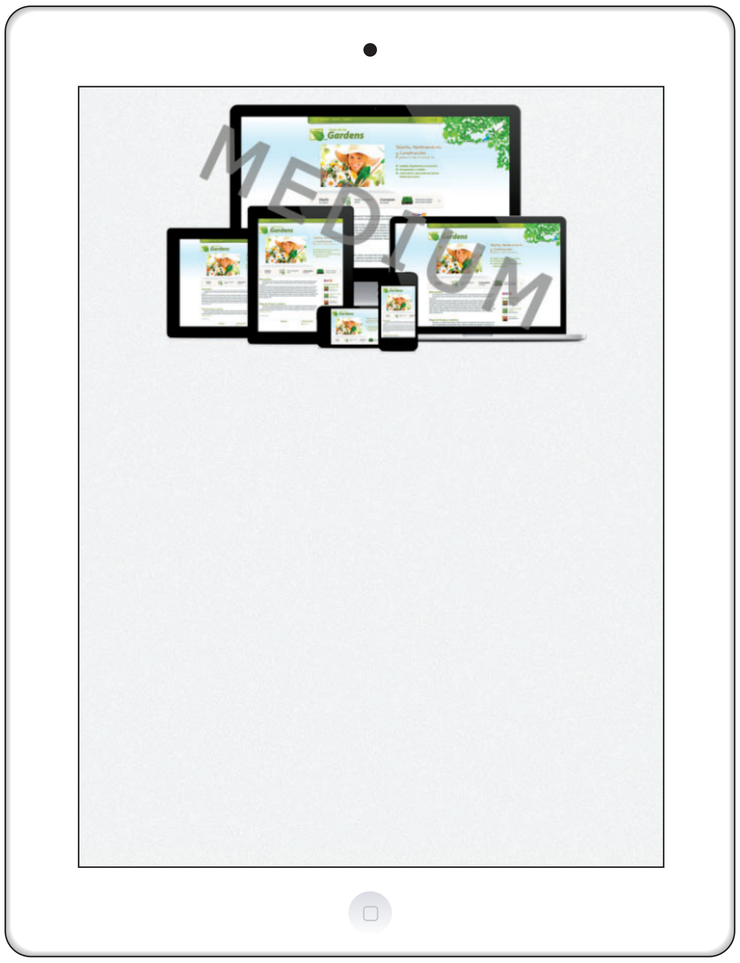
The sizes to display the image

The images that the browser can choose among

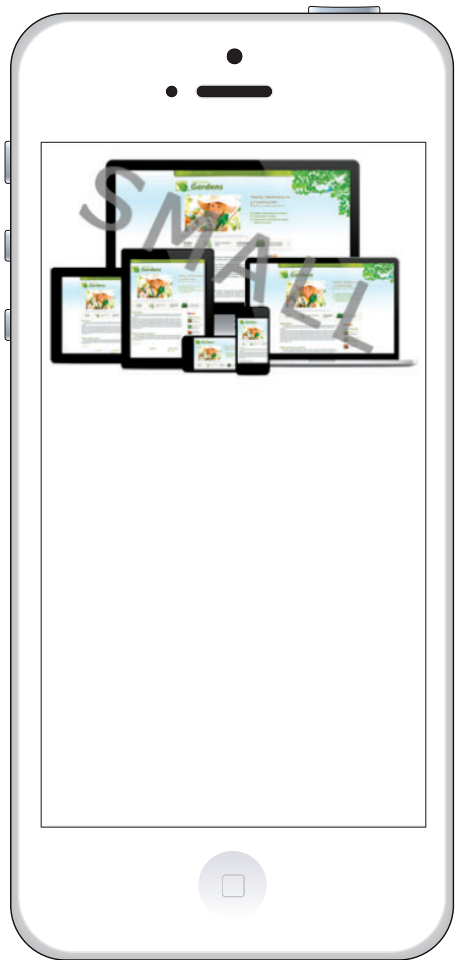
Figures 14.3 through 14.5 show how the image that's delivered to the browser changes as the size of the screen changes.



▶ Figure 14.3
A wide browser viewport gets the large image.



▶ Figure 14.4 A tablet-size viewport gets the medium image.



▶ Figure 14.5 A smartphone-size viewport gets the small image.

Making Typography Responsive

Is your goal to enrage some of the people who visit your website? I thought not, but you may be doing that if you use pixels for your site typography. Web browsers such as Google Chrome and Mozilla Firefox enable users to specify a default font size, which is set to `16px` in all modern browsers, but people with aging eyesight or visual impairments often bump this default to `24px`, `32px`, or even higher. If you use the declaration `font-size: 16px` for, say, your page's body text, *all* your visitors—and in particular those who increased their default font size—will see your text at that size. Cue the rage.

Fortunately, it's easy to avoid that scenario by switching to relative units for your `font-size` values. One possibility is the `em` unit, where `1em` corresponds to the browser's default font size—or, crucially, the *user's* specified default font size. If that default is `16px`, `1.5em` corresponds to `24px`, and `3em` corresponds to `48px`. If the default is `24px`, `1.5em` would render at `36px`, and `3em` would render at `72px`.

REMEMBER

To run your own tests in Chrome, change the default font size by opening Settings, clicking Customize Fonts, and then using the Font Size slider to set the size you want.

REMEMBER

To run your own tests in Firefox, change the default font size by clicking Menu, clicking Preferences (Mac) or Options (Windows), clicking the General tab, and then using the Size list in the Fonts & Colors section to set your preferred size.

That solution may seem to be perfect, but there's an inheritance fly in this responsive soup. First, let me point out that *inheritance* means that for certain CSS properties, if a parent element is styled with the `font-size` property, its child and descendant elements are automatically styled the same way. (See Chapter 19 to learn more about this crucial CSS concept.) To see the problem, first consider the following HTML and CSS and then answer one question: If the default font size is `16px`, what is the font size, in pixels, of the `h1` element?

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;
}
```

Your intuitive guess may be that because the `h1` element is declared with `font-size: 2em`, it must get rendered at `32px`. Alas, that's not the case, and to understand why, you need to know that the `font-size` property is inherited, which leads to the following sequence:

1 The `body` element's font size (`1em`) is set to `16px`.

2 The `header` element inherits the font size from the `body` element, so the `header` element's font size (`1.5em`) is set to `24px`.

3 The `h1` element inherits the font size from the `header` element, so the `h1` element's font size (`2em`) is set to `48px`.

That's not a deal-breaker when it comes to using `em` units; you need to be aware of this fact and take the inherited font sizes into account.

If you don't feel like doing the math required to work successfully with `em` units, there's an alternative: the `rem` unit. `rem` is short for *root em* and refers to the font size of the page root, which is the `html` element. Two things to note:

- Because the root's font size is the same as the default font size, and because the `rem` unit scales in the same way as the `em` unit, the `rem` unit is responsive.
- Because the `rem` unit always inherits its font size only from the `html` element, there are no inheritance gotchas to worry about. An `h1` element declared with `font-size: 2rem` will always render at twice the default font size.

This isn't to say that you should always use `rem` over `em`. There may be situations in which you *want* a child element's font size to be relative to its parent's font size, in which case `em` units are the best choice.

Lesson 14.3: Using Responsive Font Sizes

Covers: Using `rem` units for `font-size`

Online: wdpg.io/14-3-0

The following code updates the example page to replace the `font-size` property's absolute `px` units with relative `rem` units.

▶ Example **Online:** wdpg.io/14-3-1

This code updates the example page to replace the `font-size` property's absolute `px` units with relative `rem` units.

CSS

```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.

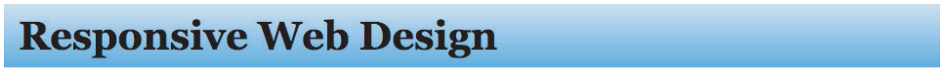
Lesson 14.4: *Using Responsive Measurements*

Covers: Using `rem` units for measurements

Online: wdpg.io/14-4-0

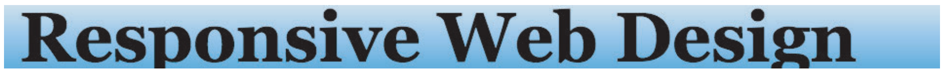
Unfortunately, the bad design results that come from using absolute units such a `px` aren't restricted to font sizes. To see what I mean, consider the following code, the results of which are shown in Figure 14.6:

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



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

Looks good! But what happens when I change the default font in my web browser (Firefox) to 30px? Figure 14.7 shows the sad story.



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

At the larger default size, the heading is larger than the `header` element in which it's contained, resulting in an overall crowded feel to the text and (much worse) to cutting off the descenders of the *p* and *g*.

Why did this happen? The header element's `height` property uses an absolute value of `64px`. That height won't budge a pixel no matter what font size you use as the default. But consider the following revised code and the result shown in Figure 14.8:

REMEMBER
This example is artificial because in practice, you'd rarely set an explicit height on an element. Instead, it's always better to let the content dictate an element's height naturally.

```
HTML:
<header>
    <h1>Responsive Web Design</h1>
</header>
CSS:
header {
    height: 4rem;
}
h1 {
    font-size: 2rem;
}
```

▶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.

How you use relative units for measurements depends on many factors, not least of which is the design effect you're trying to achieve. It's possible, however, to suggest a few guidelines:

- 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.

BEWARE

Because a percentage is relative to the parent element, you may find that using percentages for padding or margins leads to unexpected or bizarre results. In such cases, you should switch to `rem` units for more control.

Example

Online: wdpg.io/14-4-1

This code updates the example page to replace all the absolute `px` measurements with relative `rem` or percentage units.

CSS

```
.container {
    max-width: 60rem;
}
header {
    padding: 1rem
1.67%;
}
h1 {
    padding-left: 1.67%;
}
.quotation {
    padding-right: 1.67%;
}
article {
    flex-basis: 20rem;
    padding-top: 1rem;
    padding-left: 1.67%;
}
p {
    margin-bottom: 1rem;
}
aside {
    flex-basis: 10rem;
    padding: 1rem
 1.67%;
}
div {
    padding-bottom: .5rem;
}
footer {
    padding: 1rem
 1.67%;
}
```

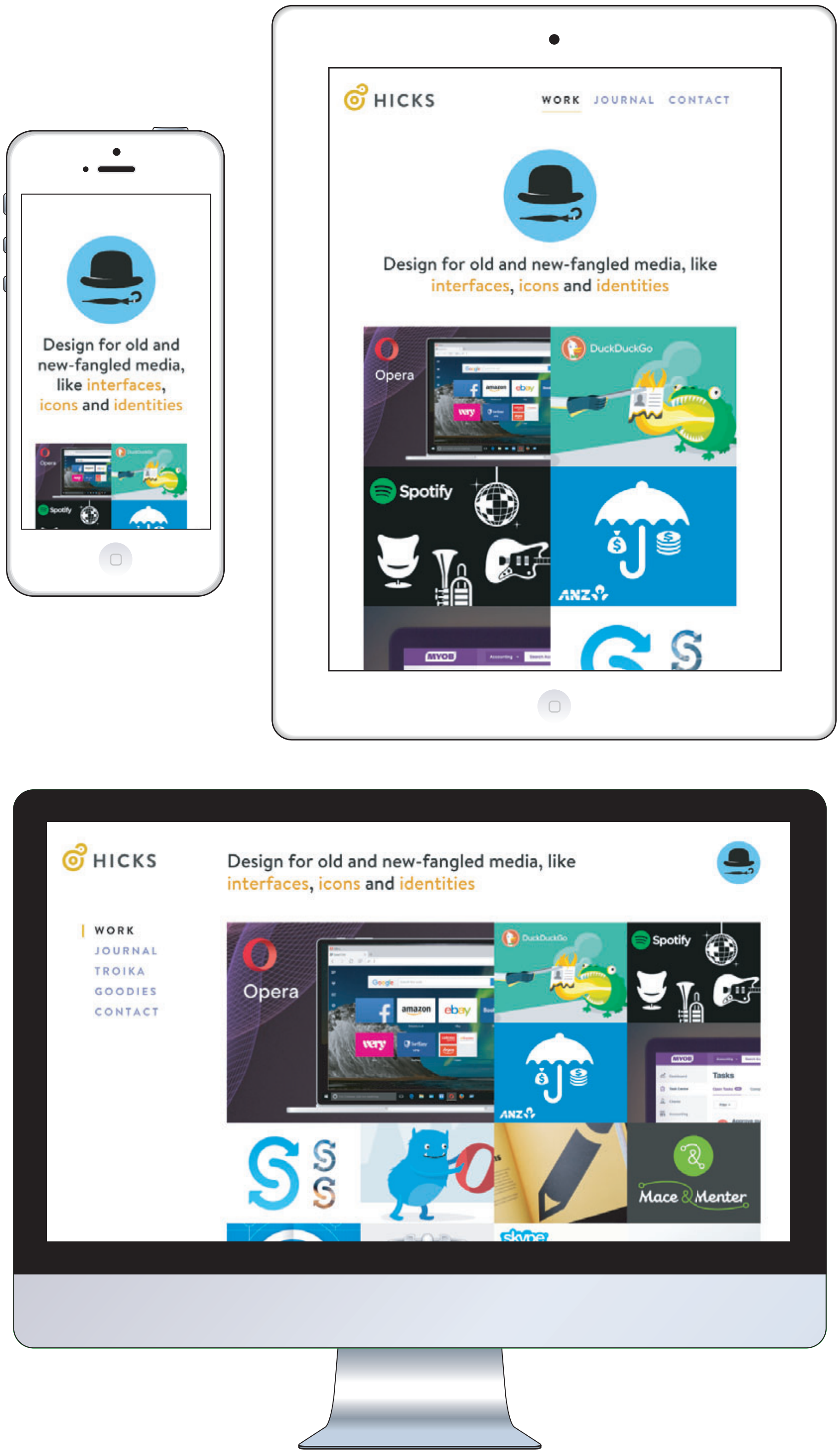
rem units used for greater control

rem units used on all vertical measures

Percentages used on all the other horizontal measures

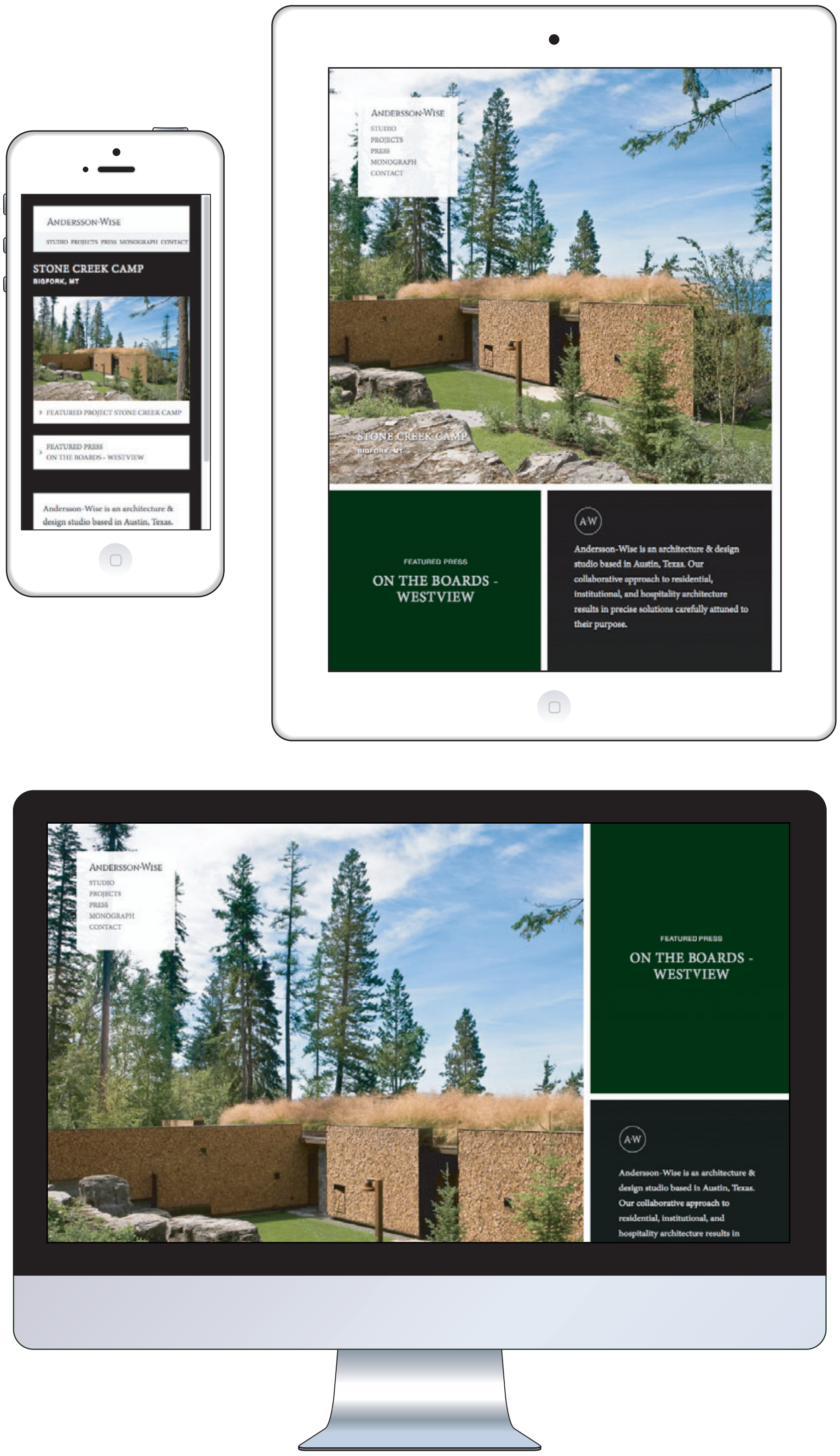
Gallery of Responsive Sites

▶ Hicks Design (https://hicksdesign.co.uk) offers a gallerylike layout that presents a clean, uncluttered look that scales perfectly to any size screen.


HICKS
WORK
JOURNAL
TROIKA
GOODIES
CONTACT
Design for old and new-fangled media, like interfaces, icons and identities
Opera
DuckDuckGo
Spotify
ANZ
Mace & Menter

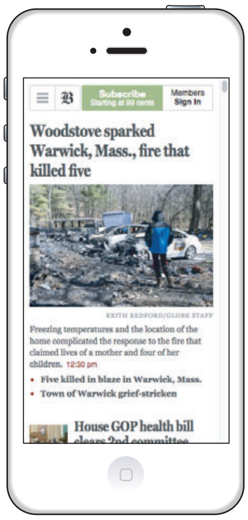
Making Your Images and Typography Responsive

▶ The Andersson-Wise site (www.anderssonwise.com) gracefully restructures its layout as it scales from the desktop version to the tablet and smartphone versions.



Andersson-Wise is an architecture & design studio based in Austin, Texas. Our collaborative approach to residential, institutional, and hospitality architecture results in precise solutions carefully attuned to their purpose.

▶The Boston Globe front page (www.bostonglobe.com) responsively changes from a three-column layout on the desktop to a two-column layout on a tablet and to a one-column layout on a smartphone.



Subscribe Starting at 99 cents | Members Sign In

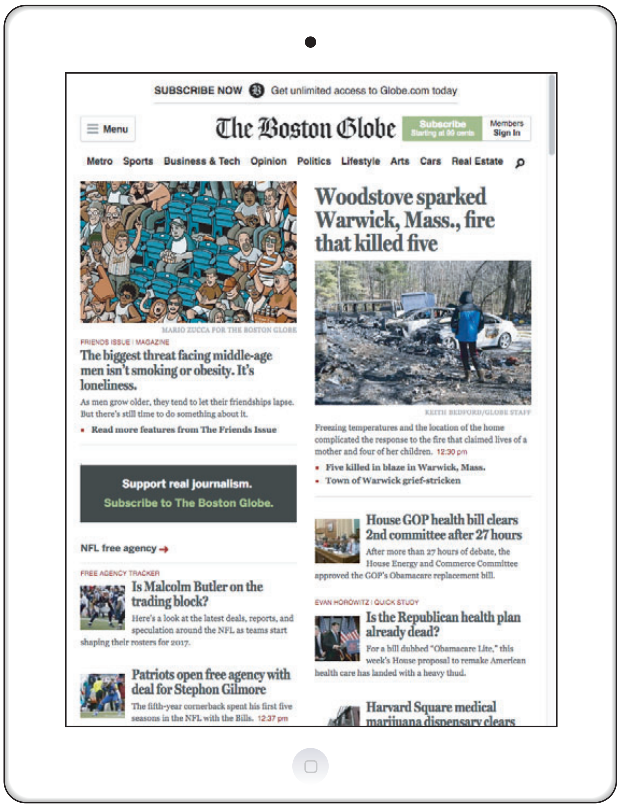
Woodstove sparked Warwick, Mass., fire that killed five

KEITH BEDFORD/GLOBE STAFF

Freezing temperatures and the location of the home complicated the response to the fire that claimed lives of a mother and four of her children. 12:30 pm

- Five killed in blaze in Warwick, Mass.
- Town of Warwick grief-stricken

House GOP health bill clears 2nd committee



SUBSCRIBE NOW Get unlimited access to Globe.com today

Menu | The Boston Globe | Subscribe Starting at 99 cents | Members Sign In

Metro Sports Business & Tech Opinion Politics Lifestyle Arts Cars Real Estate

MARIO ZUCCA FOR THE BOSTON GLOBE

FRIENDS ISSUE | MAGAZINE

The biggest threat facing middle-age men isn't smoking or obesity. It's loneliness.

As men grow older, they tend to let their friendships lapse. But there's still time to do something about it.

- Read more features from The Friends Issue

Support real journalism. Subscribe to The Boston Globe.

NFL free agency →

FREE AGENCY TRACKER

Is Malcolm Butler on the trading block?

Here's a look at the latest deals, reports, and speculation around the NFL as teams start shaping their rosters for 2017.

Patriots open free agency with deal for Stephon Gilmore

The fifth-year cornerback spent his first five seasons in the NFL with the Bills. 12:37 pm

Woodstove sparked Warwick, Mass., fire that killed five

KEITH BEDFORD/GLOBE STAFF

Freezing temperatures and the location of the home complicated the response to the fire that claimed lives of a mother and four of her children. 12:30 pm

- Five killed in blaze in Warwick, Mass.
- Town of Warwick grief-stricken

House GOP health bill clears 2nd committee after 27 hours

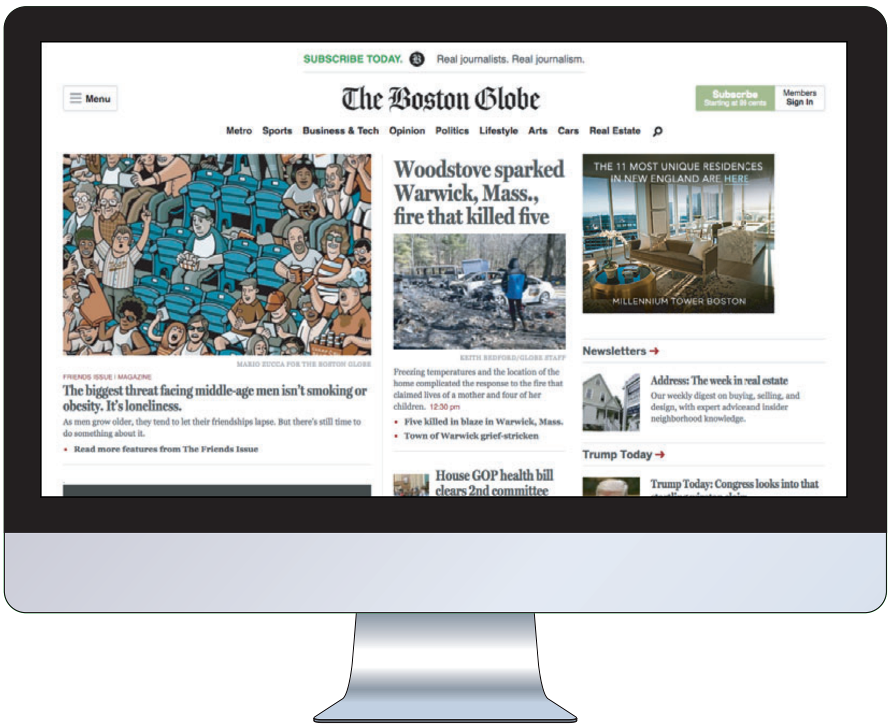
After more than 27 hours of debate, the House Energy and Commerce Committee approved the GOP's Obamacare replacement bill.

EVAN HOROWITZ | QUICK STUDY

Is the Republican health plan already dead?

For a bill dubbed "Obamacare Lite," this week's House proposal to remake American health care has landed with a heavy thud.

Harvard Square medical marijuana dispensary clears



SUBSCRIBE TODAY. Real journalists. Real journalism.

Menu | The Boston Globe | Subscribe Starting at 99 cents | Members Sign In

Metro Sports Business & Tech Opinion Politics Lifestyle Arts Cars Real Estate

MARIO ZUCCA FOR THE BOSTON GLOBE

FRIENDS ISSUE | MAGAZINE

The biggest threat facing middle-age men isn't smoking or obesity. It's loneliness.

As men grow older, they tend to let their friendships lapse. But there's still time to do something about it.

- Read more features from The Friends Issue

Woodstove sparked Warwick, Mass., fire that killed five

KEITH BEDFORD/GLOBE STAFF

Freezing temperatures and the location of the home complicated the response to the fire that claimed lives of a mother and four of her children. 12:30 pm

- Five killed in blaze in Warwick, Mass.
- Town of Warwick grief-stricken

House GOP health bill clears 2nd committee

THE 11 MOST UNIQUE RESIDENCES IN NEW ENGLAND ARE HERE

MILLENNIUM TOWER BOSTON

Newsletters →

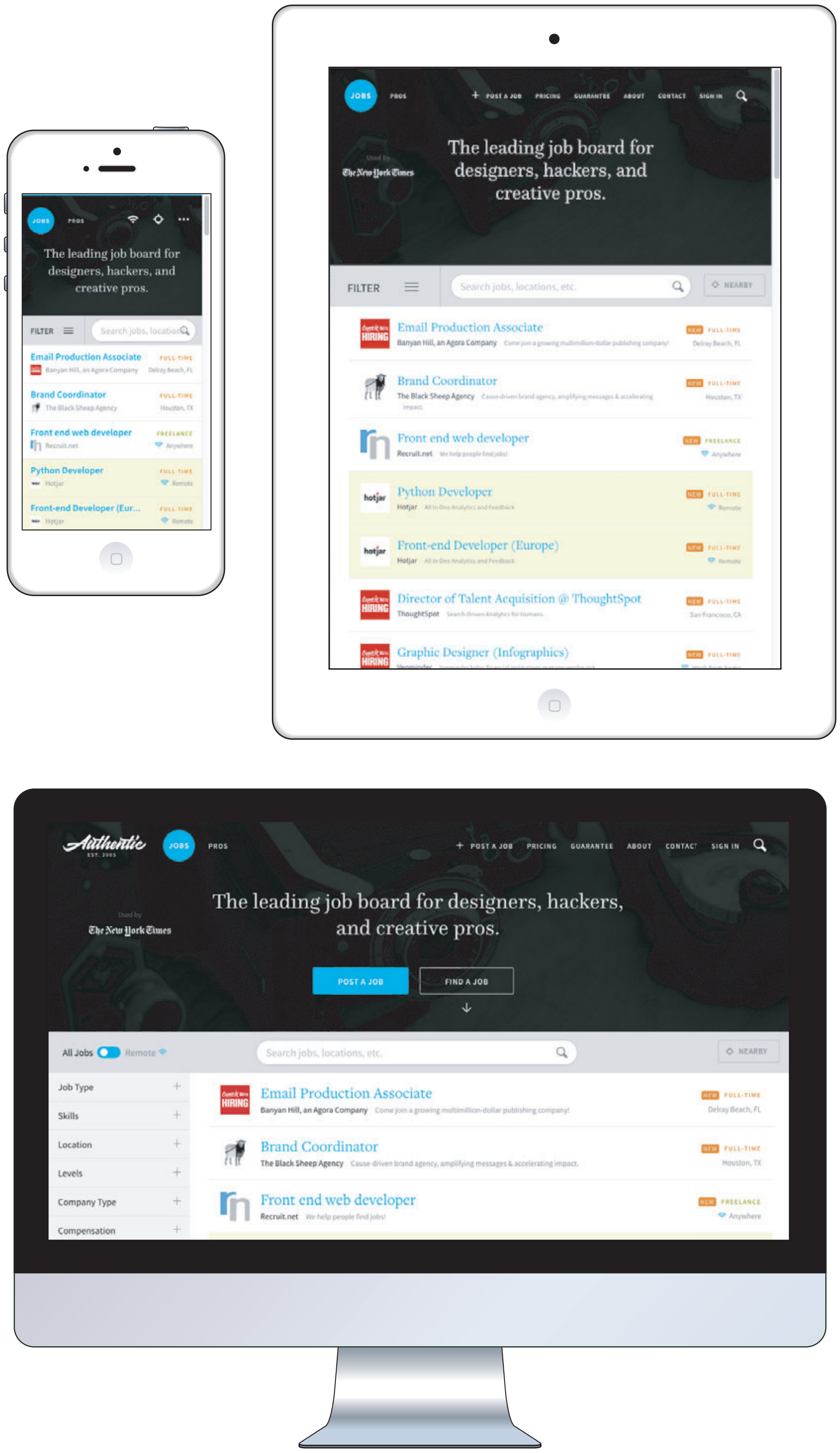
Address: The week in real estate

Our weekly digest on buying, selling, and design, with expert adviceand insider neighborhood knowledge.

Trump Today →

Trump Today: Congress looks into that

▶ The Authentic Jobs site (https://authenticjobs.com) displays a simple job list on a smartphone-size screen and progressively adds more detailed information as the screen size increases.



Authentic EST. 2005 JOBS PROS + POST A JOB PRICING GUARANTEE ABOUT CONTACT SIGN IN

Used by The New York Times

The leading job board for designers, hackers, and creative pros.

POST A JOB FIND A JOB

All Jobs Remote | Search jobs, locations, etc. | NEARBY

Job Type + | Skills + | Location + | Levels + | Company Type + | Compensation +

Email Production Associate	Banyan Hill, an Agora Company — Come join a growing multimillion-dollar publishing company!	NEW FULL-TIME Delray Beach, FL
Brand Coordinator	The Black Sheep Agency — Cause-driven brand agency, amplifying messages & accelerating impact.	NEW FULL-TIME Houston, TX
Front end web developer	Recruit.net — We help people find jobs!	NEW FREELANCE Anywhere
Python Developer	Hotjar — All in One Analytics and Feedback	NEW FULL-TIME Remote
Front-end Developer (Europe)	Hotjar — All in One Analytics and Feedback	NEW FULL-TIME Remote
Director of Talent Acquisition @ ThoughtSpot	ThoughtSpot — Search-Driven Analytics for Humans	NEW FULL-TIME San Francisco, CA
Graphic Designer (Infographics)		NEW FULL-TIME

Summary

- Make your images fluid by styling them with the declarations `max-width: 100%` and `height: auto`.
- In your `<img>` 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.