

GYMNASIUM

RESPONSIVE WEB DESIGN

Lesson 3 Handout

Mobile First Overview

ABOUT THIS HANDOUT

This handout includes the following:

- A list of the core concepts covered in this lesson.
- The assignment(s) for this lesson.
- A list of readings and resources for this lesson including books, articles and websites mentioned in the videos by the instructor, plus bonus readings and resources hand-picked by the instructor.
- A transcript of the lecture videos for this lesson

CORE CONCEPTS

1. Understanding the scope of worldwide mobile usage for accessing the Internet and how user behavior needs to be well understood for the most effective design and development.
2. Examining the components of “Mobile First”: the practice of making mobile design a priority instead of the desktop. Specifically, we cover the concept of Progressive Enhancement, which can be summed up as creating a baseline experience for the smallest screen layout and then adding layers of enhancement or features for larger screens/devices.
3. Optimizing your primary layout for a smartphone by adding the viewport meta tag to prevent scaling of the page, as well as adding JavaScript to force the browser address bar in iOS and Android devices to disappear on page load.
4. Creating a single-column layout optimized for the small screen of a smartphone, providing structure in the form of <div> elements with ID styles and adding styles for the images to ensure the width and height responsively adapt to the size of the browser.

ASSIGNMENTS

1. Quiz
2. Download and skim Google’s New Multi-Screen World Study.
 - a. <http://www.google.com/think/research-studies/the-new-multi-screen-world-study.html>
 - b. You do not need to read this cover to cover, but at the very least read the following sections:
 - bi. The Executive summary
 - bii. The two modes of multi-screening
 - biii. Found time
 - biv. Implications for business
 - c. Note three items you found the most surprising and/or useful for you and post these in the Gymnasium Forum.

3. Create screenshots of work that represents you and add images to your Portfolio-in-Progress.
 - a. Take a minimum of three screenshots of work that you have done and add it to your portfolio. Follow the lead of the sample portfolio that is being used in the video and make these 400 pixels wide, and then add them to the Portfolio section of the page.
 - b. After you have inserted these in the document, add the `img {max-width:100%;}` code to your stylesheet.

RESOURCES

READING

- <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#subscribers>
Mobile subscribers worldwide
- <http://www.lukew.com/ff/entry.asp?933>
Luke W's original 2009 blog post on "Mobile First"
- <http://barryclark.co/adaptive-responsive-or-mobile-first/>
Adaptive, Responsive or Mobile First?

SITES MENTIONED IN PRESENTATION

- <http://www.google.com/think/research-studies/the-new-multi-screen-world-study.html>
- <http://home.web.cern.ch/>
- <http://developer.apple.com/library/ios/#documentation/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html>
- <http://24ways.org/2011/raising-the-bar-on-mobile/>

INTRODUCTION

(Note: This is an edited transcript of the Responsive Web Design lecture videos. Some students work better with written material than by watching videos alone, so we're offering this to you as an optional, helpful resource. Some elements of the instruction, like live coding, can't be recreated in a document like this one.)

Welcome to Responsive Web Design, a Gymnasium course brought to you by Aquent and Vitamin T. Responsive Web Design or Promote Yourself Responsibly-- Build a Portfolio For All Devices. This is Lesson 3, Mobile First. There will be an assignment and brief quiz at the end of the lesson. Be sure to use the pause button throughout these videos in order to review concepts or to look at code. And finally, if you have questions, be sure to hit the Forum, where we have TAs as well as your other classmates to help you get through some problems.

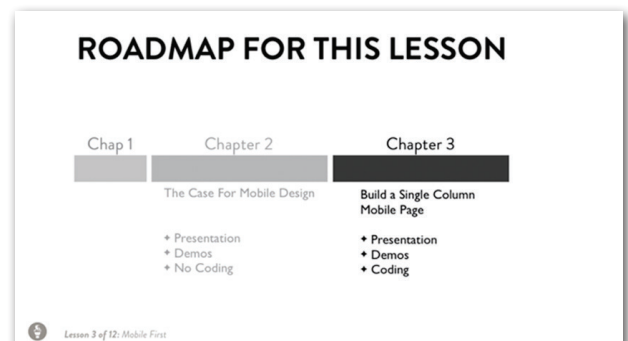
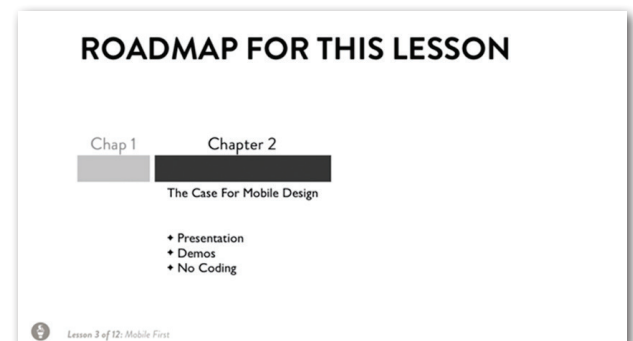
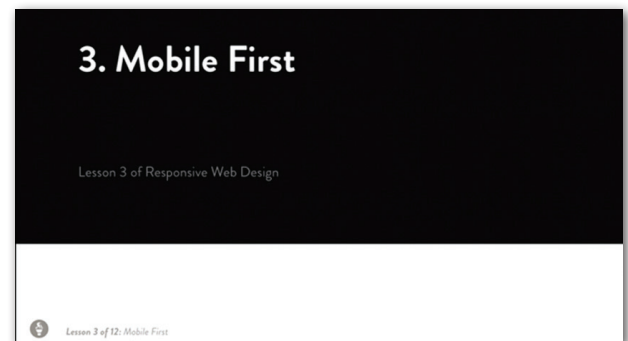
This is Responsive Web Design.

THE CASE FOR MOBILE FIRST DESIGN

Now before we begin Chapter 2, let me just briefly outline what we'll be doing for the rest of this lesson. In this section, we'll be talking about the case for mobile design. And there will be a presentation, some demos, but there will be no coding until Chapter 3. Chapter 3 will be building a single column mobile page based on some lesson files that were started in the previous lesson. And there'll be demos and coding, of course.

So this is the case for mobile first design. And we'll be taking a look at the scope of mobile coverage, why mobile design is a priority, exploring the components of a mobile-specific layout, as well as the principles of progressive enhancement. We're going to get through this quickly but the goal is to explain some of the lingo associated with these technologies. This is going to help us make sure we're all on the same page, as well as when you're talking to other clients or customers.

So let's start with some stats. We have, here, the planet Earth, 7 billion people. In year 2011, there were 6 billion mobile subscriptions. I'll let that sink in for a second. Can you guess the percentage there? 87%. 87% of the world has a mobile subscription. It's actually probably higher now since it's the year 2013.



If you needed more convincing, mobile subscriptions outnumber fixed telephone lines 5:1. Mobile broadband subscriptions outnumber fixed broadband 2:1. Now, of course, the thing to keep in mind here is that this is including developing nations which actually have a far greater number of mobile subscriptions.

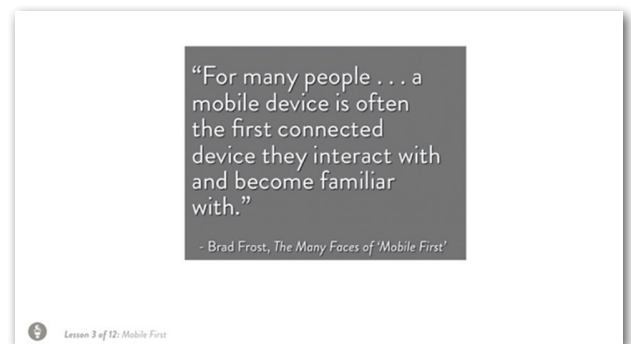


So mobile first has multiple meanings depending on who you ask, but we'll start at the beginning. In year 2009, Luke Wroblewski came out with a great article called "Mobile First" and this really set the stage for the term. And you could go ahead and read that article. I strongly recommend it. Luke has also come out with some great books. Feel free to go and explore all those. They're incredibly great resources.

Now the term mobile first has evolved over the years, and here's another take on it. Brad Frost who's written a great article called "The Many Faces of Mobile First" says that for many people, a mobile device is increasingly the first connected device they interact with and, thus, become familiar with.

The mobile ecosystem and the way we talk about it is changing so quickly that it's hard to keep up. New terms are emerging all the time. Mobile only, for example, is a relatively new term that describes individuals and even entire cultures whose only access to the Internet is through mobile devices. If someone will never see your website on a desktop monitor, you better make sure your mobile experience is good if you want them to stick around or come back.

Too often, design decisions about mobile are based on faulty predictions about user behavior. Be sure to do your due diligence and research, or you'll likely waste time and effort creating a mobile site for an audience that isn't actually relevant. You can start making better decisions about your mobile users by looking at the numbers.



And here's a company that knows a lot about numbers as well as mobile users, Google. In 2012, they released the results of a survey of over 1,600 participants on cross-platform consumer behavior. In other words, are people using multiple devices at once? And if so, what are the implications for content developers and businesses?

The answer, not surprisingly, is yes. People are definitely using multiple devices at once, and the data breaks down some common myths about mobile usage. Myth number one, smartphone users always have slower connections. Myth number two, smartphone users are always on the go and in a hurry. So the truth is 77% of the time when using a TV we're actually using another device. Specifically, a smartphone about half the time.

MYTH#1
SMARTPHONE USERS ALWAYS HAVE
SLOWER CONNECTIONS

 Lesson 3 of 12: Mobile First

MYTH#2
SMARTPHONE USERS ARE ALWAYS ON
THE GO & IN A HURRY

 Lesson 3 of 12: Mobile First

If we flip this on its head, 57% of the time when using the smartphone, we're actually using another device at the same time. Which one? Well, specifically, a television at least 29% or one third of the time.

So if we take these two facts and put them together, when people are using simultaneously a television and a smartphone that typically means that Wi-Fi is present. And if Wi-Fi is present, that means they don't have a slow connection and they're not on the go.

The point here is examine mobile data and then use that to drive your design decisions. In the end, designing for mobile first is advantageous. So let's go back, way back, to the year 2010 and 2011 when responsive design first came onto the scene. Typically, the way that a responsive site might be built was this. We have a large screen and we can make it have five columns, a smaller screen has three, and a smartphone device has a single column. They all use fluid grid, flexible media, and media queries, that's responsive design, great.

However, there are some problems with that model. One of the problems is the mobile design comes at the end. So if you're a designer or a developer, you know all about deadlines. And the truth is mobile design typically gets the short shrift at the very end of the process. But additionally, features are removed and, perhaps more importantly, there are actually performance penalties when you do it this way. We'll get into some of the technical reasons for that, but that's the fact.

If we flip it on its head and we start with the design and development of the mobile first or the smartphone and then we move up to a small screen and then a large screen, that still gives us fluid grid, flexible medias, and media queries, but now we're doing it differently. Make mobile design a priority

MAKE MOBILE
DESIGN A PRIORITY.
NOT DESKTOP*

* Regardless of platform make users a priority

 Lesson 3 of 12: Mobile First

and not the desktop. That's really what mobile first is about. Now if we're being picky, I would actually say regardless of the platform, you should be making users a priority, but hopefully you get the point.

Now at this point, I should probably talk about adaptive design. So depending on who you ask, adaptive is a subset of responsive. If you ask other people, responsive is a subset of adaptive. But let's just ignore that and let's just define what adaptive design means to me.

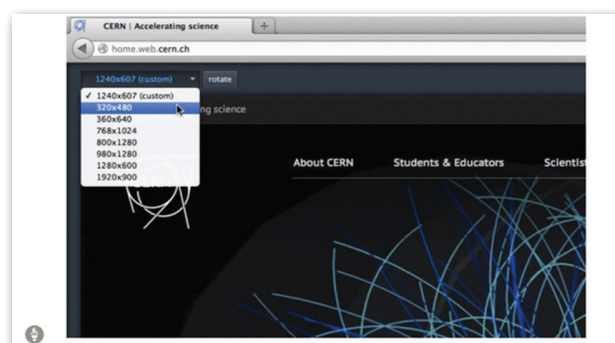
Adaptive design still has breakpoints so we can have separate layouts, however, typically adaptive design does not have a fluid grid or flexible media. It still has media queries and it might look something like this. So this site has media queries that trigger three different layouts as I change the width of the screen. Notice that the images are not resizing as the browser window changes and the layout grid is relatively static. Based on their screen size, visitors to the site will get a specifically formatted layout, but not the full range of variable layouts that fluid grids and flexible images would provide.

The point is the world is actually big enough to hold both the terms responsive and adaptive. These terms aren't as important as what they have in common. We need to use new and developing technologies to better serve our users, no matter where they are, no matter what device they're on.

So let's ask these two people what they think mobile design is. On one hand, we might say apps. On the other hand, we might say websites. For this course, we're focusing on mobile websites and not apps. However, we've got a lot of good things to learn from mobile apps. The best mobile apps force us to focus on tasks and features. And so we can take some of the same concepts and bring them over when we're using mobile web design. Here we're establishing the core experience on a small screen first.

So let's take a look at another example. This is the homepage of the recently redesigned CERN website. So you can go ahead and take a look at this as well. So one of the interesting things about this site is it's the heart of a lot of scientific exploration. They have particle accelerators and all sorts of cool things, but it was also where the first website was born. So Tim Berners-Lee created some of the first websites for CERN and this is very interesting for lots of reasons.

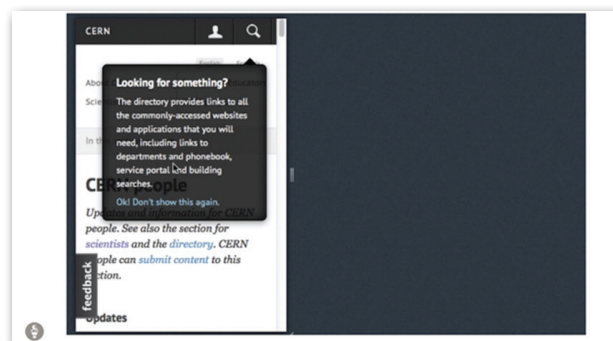
So let's take a look at how the responsive site works here. And what I want to show you actually is a neat little trick that I have in Firefox. So in Firefox, we have the ability to go into Tools, Web Developer, and then we can choose this option



Responsive Design View. And what Responsive Design View allows us to do is go in and choose certain breakpoints or certain width and height dimensions in order to reproduce the effect of a mobile phone. You can do the same thing in other browsers by simply changing the width of the browser, but this is a really nice shortcut.

So we have a 320 by 480 viewport here and let's just take a look at some of the features. I'll go ahead and click on this link and we can see that we have an expandable navigation. I'll go ahead and collapse that.

Notice as we begin to explore the navigation when I hover over a certain section, I get this option. This is a navigational aid and we'll talk a little bit more about this later. But this is something that only pops up the first time you visit the site. And what's it saying? It's basically saying, hey, if you're looking for something, try this. This magnifying glass will give you the directory. So what does the directory look like? Well let's explore that. The directory is a single column, and we have all of the high level navigation that we need here for experiments and projects. We have a phone book. We have a Search button. Let's go back to the homepage.



Now in the homepage, as we scroll down, I'm going to click here on this link. And this link is going to bring me to something that is not responsive. So this is somewhat a flaw. We can forgive them because there's a lot of pages on the site and there's a lot of challenges, but it is interesting. And we'll take a look at that later.

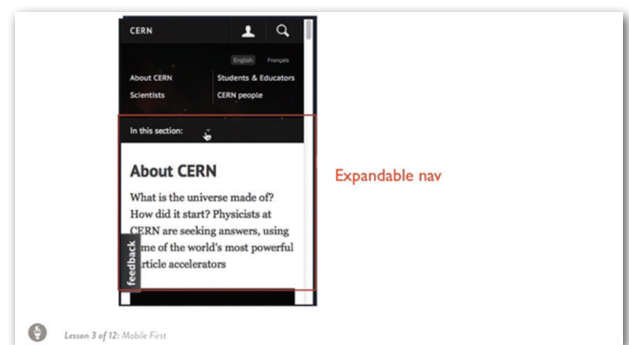
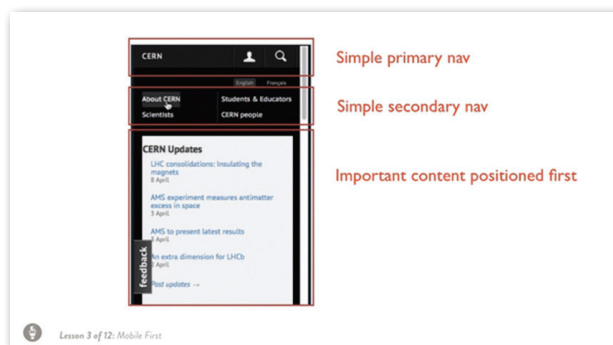


Now what I'll do here is I'll flip into the Tablet view or what we can think of as the Tablet view, 768 by 1024. Notice the layout changes. And what I'm going to do is I'm going to scroll down. I'm going to click on that same link that took me to the website earlier. Now it does something totally different. It shows me a high resolution view of this cool particle explosion that's taking place here. So this is a feature that we're adding for larger sites.

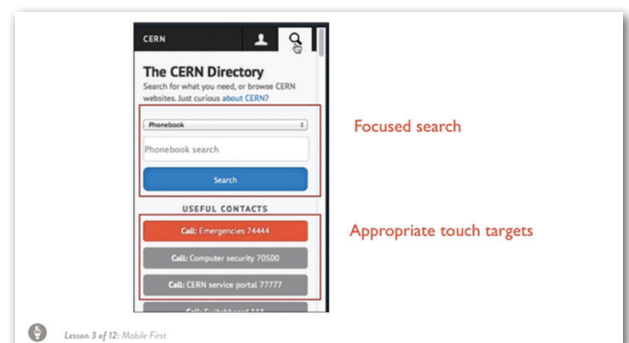
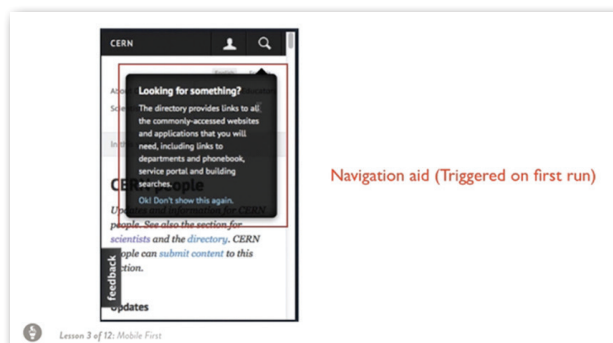
Finally, I'm going to show you this on the large Desktop view. So I'm going to go back to Firefox, and I'm going to turn off the Responsive View by unselecting it or deselecting it.

We can sum this all up as progressive enhancement, which is about creating a solid foundation for the smallest screen. And then we can add features for larger screens as needed. That's exactly what we saw there. So I'm going to break this down a little further. Let's go ahead and take a look at some of the components.

So first of all, we had the simple primary nav. We had the simple secondary nav. We had the primary content positioned first. We also had an expandable navigation. We had a navigation aid, which is only triggered on the first run. Of course, you can see the link there, it says "Don't show this again."



In the directory, we had a focused search, so the ability to look for a phone book. And notice the appropriate touch targets here. So we have nice big buttons. And in this case, one of them is colored red and that's the emergency one, which is a nice design choice.



In the next section, we'll begin to put some of these practices into motion by adding them into our portfolio page. But for now, this is Responsive Web Design.

BUILD A SINGLE-COLUMN MOBILE PAGE

In this section, you'll begin to build a single column, mobile page. Specifically, you'll learn how to control your page within the mobile viewport. You'll be adding HTML structure and CSS style for a single column small-screen layout, and you'll be adding initial support for images within that same small-screen layout.

So someone told me that our site is responsive already. Let's just check this out. What could this possibly mean? All right. Well, we have our design here. As I begin to change with the browser, you can see that text is reflowing. So that's pretty good.

Well, I mean we have some work to do. But darn it, maybe this is pretty close to responsive. High-five! Let's go check it on our phone. Hmm. This is not looking the same. What's going on here?

Well, the issue is what we're seeing is the default behavior of, in this case, the iPhone. It's attempting to scale the entire page to the size of the viewport. But we don't have to worry. There's something called the viewport metatag that we'll be using, and this will allow us to control the viewport's size and scale.

This is what the code looks like. It's a simple metatag with a few properties and values. The device width value sets the width of the screen to 100%. Then we have the initial scale property. And this has a value of 1. So this controls the zoom level when the page is first loaded.

If we go ahead and add that code, this is what we get. And this is a much better view. The viewport metatag isn't an official web standard, but it's very well-supported in most popular browsers. If you need to do a little bit more research, you should look for the Configuring the Viewport section of the Safari Developer Library. And we'll put the link in our classroom link section.

Now one other thing when it comes to that viewport and that mobile view. Does this bother anyone else? This address bar here? Wouldn't it be great if we could get rid of this, and it would give us a little more space for our design? So the good news is we can use some JavaScript in order to make the address bar disappear. And this will work in both iOS and Android. Because we do have some code that could make this work in just iOS, but we need something that works in both platforms.

WHAT WE'LL BE COVERING

- ♦ How to control your page within the mobile viewport
- ♦ Adding HTML structure and CSS style for a solid single column small-screen layout
- ♦ Adding initial support for images in a small-screen layout

Lesson 3 of 12: Mobile First

```
<meta name="viewport"
content="width=device-width,
initial-scale=1"/>
```

The device-width value sets the width of the screen in CSS pixels at a scale of 100%

Lesson 3 of 12: Mobile First

Does this address bar bother anyone else?

We can use some JavaScript in order to make the address bar disappear in both iOS and Android browsers



Lesson 3 of 12: Mobile First

So this JavaScript makes the address bar disappear. If the user begins to scroll, It will pop up momentarily and then disappear. From the HTML perspective, it's a very simple line of code to add. This is a browser agnostic approach for hiding the address bar.

If you want to learn more about the details, you should look at this article, which is incredible. It's an excellent article. And the solution to this actually took a lot of work, I'm sure. And you can go to this link and read much more about it if that is your pleasure.



So we add those two lines of code, and now our page is responsive? High-five! Yeah . . . no. The thing is, this is just the beginning. There's lots more to do. It's time to get real with code. So what you're going to need to do is open up your text editor. And we're going to walk through building the structure of this page and adding some styles.

So specifically, you're going to be creating a solid one column layout. We're not concerned about the desktop. So again, this is just our mobile view. Whatever the desktop layout looks like, we'll be addressing it later. Additionally, there's going to be some fine-tuning of typography, margins, and padding that we're going to have to do later. And we'll be adding some additional images. But when all is said and done, we're actually going to have a very solid mobile layout in about seven or eight minutes.

WHAT YOU'LL BE BUILDING

- ♦ A solid one-column layout based on our HTML5 foundation
- ♦ Not concerned about desktop layout
- ♦ Will need to return and fine-tune typography and spacing
- ♦ Will be adding additional images

Lesson 3 of 12: Mobile First

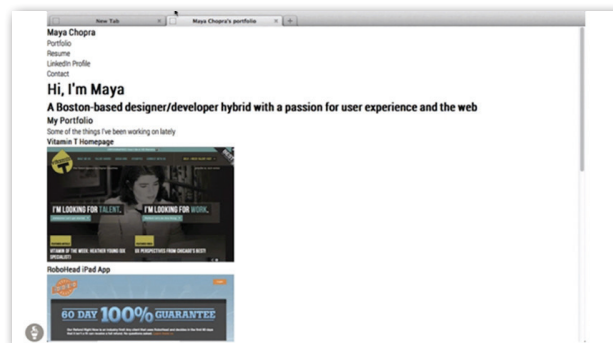
OK. To begin with, you're going to need to open this file located in your lesson folder, `portfolio_start.html`. Rename this `portfolio_work.html` and save it, because this creates a backup. Again, we want to take a look at this in the browser just so we have a sense of what it looks like.

I am using Firefox here and as I noted in a previous code example, I use this feature called the Responsive Design View. And this is going to allow me to see the page at 320 by 480. You can get the same effect if you're not using the latest version of Firefox, or Firefox at all, by simply resizing the width of your browser.

As I noted before in the presentation, I've added these two tags for you, so the metatags here and the script for the hide address bar code. Additionally, there's a style link here that you can use, and it's empty right now. But we're going to be using internal stylesheets for the majority of this lesson.

There is an external stylesheet. We're going to leave that alone for now. And eventually, we can move the internal styles to the external ones. So the first thing we're going to do is we're going to add an ID to various parts of the page. And the first ID that we're going to add here is the `id="nav"` to this first section. Then we'll say `id="masthead"` for the header. Next will be `id="content"`. And then the last one we'll add is for the footer at the very bottom of the page. And we'll call this the `id="colophon"`.

Now that we've added those IDs, let's go ahead and add some styles for those. And we're going to put these all in a group. And the goal of this is to give us some better margins for our entire layout. So in this case, we're going to say `margin`, top value and bottom value, zero. Left and right values, 18 pixels. Let's reload our page. And now we get some nice margins on the left and the right.



If you notice this navigation section, it's not working very well. This is also something that we're going to return to later. So there's an entire lesson devoted to mobile navigation. To make things easier, we're going to comment that entire section out, so that we don't have to worry about it at this point. So go ahead and add some comments here right above the nav section. And of course, you're going to need to close the comment as well.

```
22     margin: 0 18px;
23 }
24 </style>
25 </head>
26
27 <body lang="en">
28   <div id="container-nav">
29     <section id="nav">
30       <h3>Maya Chopra</h3>
31       <!-- Turn off for now
32       <nav>
33         <ul>
34           <li>Portfolio</li>
35           <li>Resume</li>
36           <li>LinkedIn Profile</li>
37           <li>Contact</li>
38         </ul>
39       </nav>
40       <!--
41     </section>
42   </div>
```

OK. Next up, we're going to be adding some div structure to various parts of our page. Now wait a minute, I can hear you say perhaps, we're using HTML5. Why are we starting to use divs? Well, these divs are going to be used primarily for the visual appearance for styling.

And this is essentially what we use divs for now in HTML5. We can add things like background images and background colors. These are primarily styling elements, not necessarily structural or semantic. So the first one is `div id="container-nav"`. This is wrapped around the first section. The second one is immediately below. So we're going to say `div id="container-header"`. And of course, this goes around the header section. We'll just close it here.

Next, let's go ahead and add some styles for this. So the first one, of course, is `#container-nav`. This will be a simple background color. In this case, it's white, which won't make a huge difference. But let's copy and paste it. Call that `#container-header`. And for this one eventually, we're going to put a background image, but we're going to use a background color. And we're going to put in a height of 350 and a width of 100%.

```
15 <!--[if lt IE 9]>
16   <script
17   src="//html5shim.googlecode.com/svn/trunk/html5.js"></script>
18 <![endif]>
19
20 <style type="text/css" media="all">
21   #nav,#masthead,#content,#colophon {
22     margin: 0 18px;
23   }
24
25   #container-nav {
26     background: #fff;
27   }
28
29   #container-header {
30     background: #352a38;
31     height: 350px;
32     width: 100%;
33   }
```


We can almost think of this as a placeholder, because eventually, we're going to be using some background images. But for now, let's just go ahead and take a look at what this does. If we reload our page, this is the result. So notice the navigation is gone. Now we have this large field of color. 350 pixels high. 100% wide. And we're going to return to that and do a little extra styling at the very end.

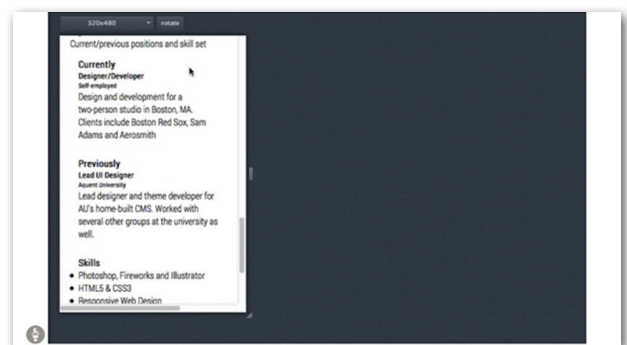
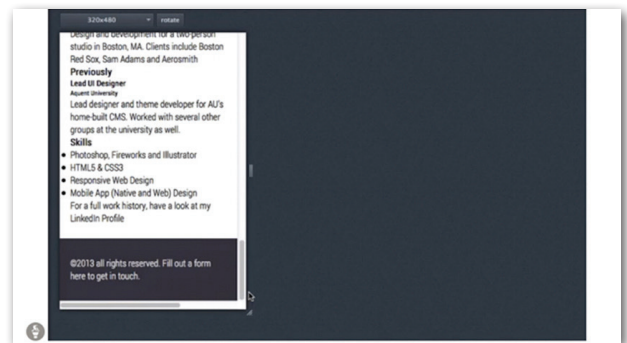
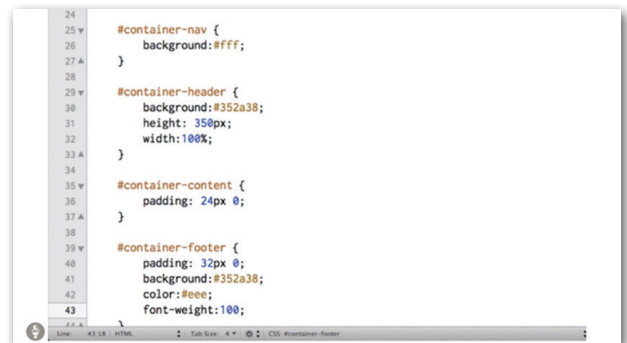
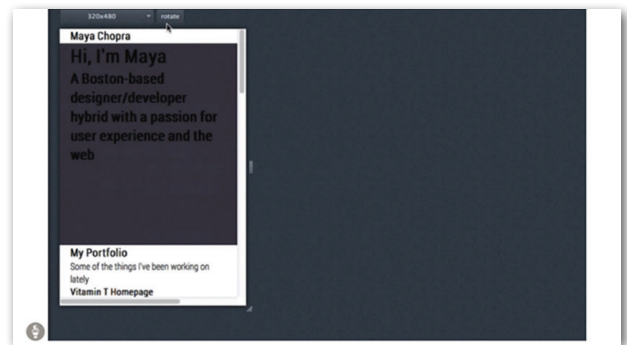
But let's keep moving on. Now just like we did before, we're going to add a few other divs. And this one is called id="container-content". We're going to wrap this around the article that we put in earlier. So that closing tag should go right after the closing article tag right around line 128.

And while we're here, let's go ahead and add another one of these divs called id="container-footer". And we're going to wrap this around the last footer on the page. Since we just put in those IDs, let's go ahead and style them. So let's start with the #container-content. And we simply want to put in some padding. And we're going to put in 24 pixels of padding from the top and the bottom. Nothing for the left and right.

Let's copy and paste that entire block. Put in the style for the #container-footer. And the padding here will be 32 pixels. We'll add the same background color that we did for that placeholder up above. So that's going to be a dark purple. And we're going to make the font color close to a white. Not quite a white, but similar. Font weight, 100 also.

Let's save our page. Take a look at what this does. And if we scroll down, you'll see the effects of the header. And now we're getting some nice spacing and padding through the rest of our page. We got a little more to do. We're going to be adding quickly three boxes here, called div class="box". And we're going to put first one around this code that starts currently right underneath the My Resume heading. That's the first box.

Let's go ahead and copy that, paste it below. Make sure that we close it. The second box is going to wrap around the section called Previously. And the third box, we're going to go ahead and put around this section here.



So we have these three boxes. Each one of them will have a padding of 14 pixels on all four sides. Let's go ahead and save our document, reload, and see what this gives us. It's looking pretty good here. So this indents everything a little bit. So we're going to ignore those ugly list styles for a moment. And we're done with this section.

So we're going to come back and do a little extra work with this code. But hold on. Yes, sir. You have a question. What is your question? Why pixels? You use ems. OK. Fair enough. I love ems too. But we're going to keep it simple for now. Later on, you'll be seeing plenty of ems. The ems, they will come. Don't you worry.

So let's quickly talk about images. So we have a 320 by 480 viewport that we've been working with. The images, however, that we're using, are 400 pixels wide. So what's happening here is that they're jutting out, or they're larger than can fit inside of our viewport.

So this brings a few problems. For example, this forces the user to scroll. It doesn't look very good. In general, it's just not a great solution. So what can we do about this? Oh, no, no, no. We don't want to reduce the size of the images. To do that would ignore our desktop mockup completely.

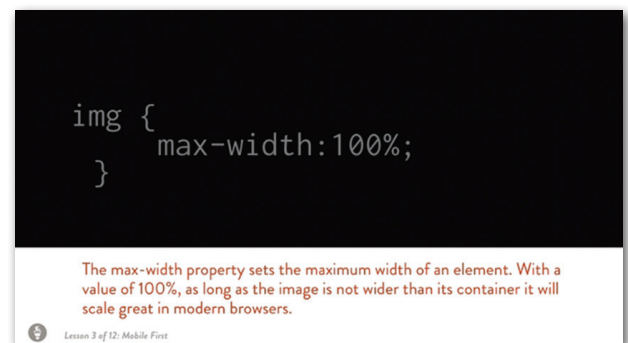
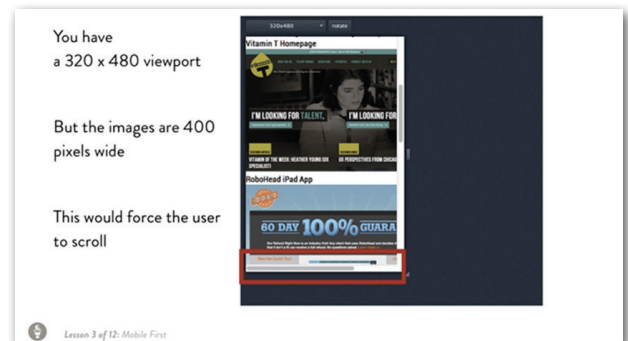
So this is our mockup. And here, it's been decided that those images have to be 400 pixels wide. So we can't go ahead resizing our mobile ones, or else this will affect the desktop layout. So it is extremely inconvenient. But there is a good solution.

This little line of code here will do a whole lot of work. The style here is for an image, and the property and value is `max-width: 100%`. The `max-width` property, if you haven't used it before, sets the maximum width of an element. When you set a value of 100%, as long as the image isn't wider than the container, it will scale great in all browsers.

So it's not going to get any larger than the native size, which is 400. But if the container of the page is less than 400, it will scale. This is great. The only place this is not great, by the way, is older versions of IE. And in later lessons, we're going to have to address this.

But for now, our images scale nicely within our single column layout. You'll now notice that there is no scroll bar, which is great. If we rotate the phone, it even looks great in landscape view. So in general, this is a pretty good solution. Later on, it will have some consequences that we'll have to deal with, but for now we're in good shape.

Let's go ahead and quickly add this to our code. Again, it's a very simple style: `img{max-width:100%;}`. So of course, this is going to affect all images on the page and we only have three of them. But let's go ahead and add that style, reload our page, and just take a look at it. This is looking nice!



So at this point, we actually have a pretty decent foundation. Again, if we rotate this and look at it in landscape, it works very well there. What's not to like? So I'm going to go ahead, and I'm going to change that text there that is black to white. I'm going to create a new style here. And this is masthead. And the color will be white. Let's go ahead and save the page. Reload. And I like that a little better. I'm going to like that a lot better when we start to add images for this section.

But for now, we're in the clear. We've created a good, solid, single column layout. In upcoming lessons, we're going to focus on better typography, and we're going to get serious about the images. And you better watch out, because media queries are on the horizon, and they're going to be lots of fun.

Yes, you may now high-five! Good job. Here's a little homework for you. Assignment number 1, as always, is a short quiz that will test your knowledge and reinforce the concepts in this lesson. Assignment number 2, I'd like you to download and skim the multiscreen study that I mentioned in chapter 2. Here's the URL here. Go ahead and skim through it. You don't want to necessarily read the whole thing unless you want to. But chances are you're going to find at least three items that you find surprising. I'd like you to note those, and then post them in the Forum. And hopefully, we can get an interesting discussion going on about some of the changes that are happening these days in the world of mobile.

Assignment number 3. Add images to your portfolio in progress. As we have said in the previous lessons, the goal here is to get you a responsive portfolio page by the end of this course. If you could take three to six screenshots of projects that you've worked on, and then use those within your code.

Again, you can use the template from this course as a model, but you want your own work in there.

Be sure to add the image with the 100% code to your stylesheet. This will make sure that it works well in a mobile browser. There's a little bit more information about this assignment in the classroom. I'll look forward to seeing you there. And I look forward to seeing you for the next session.

