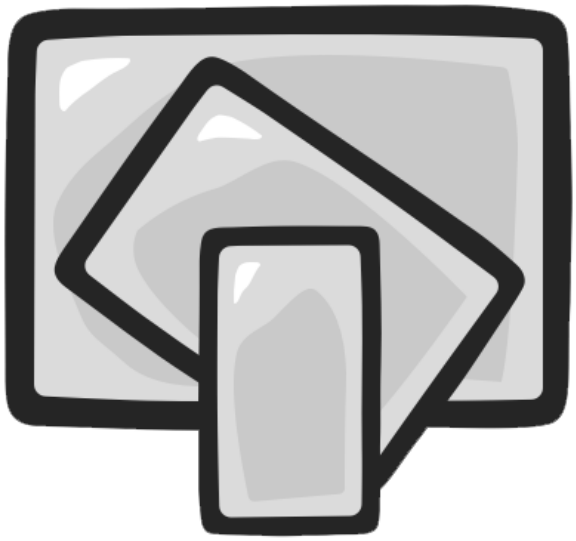


GYMNASIUM

RESPONSIVE WEB DESIGN FUNDAMENTALS

Lesson 1 Handout

Introduction: Setting the Stage

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. As defined by Ethan Marcotte in his 2010 article, a web page or web site built with Responsive Web Design principles consists of Fluid Grids, Flexible Media and Media Queries.
2. One of the primary benefits of Responsive Web Design is having one set of code that's renders the page differently on different devices, such as desktop, phone, and tablet.
3. Building your websites responsively helps your users in many ways, but it also helps your website perform well in Search Engine rankings, specifically in Google, which in 2015 began using an algorithim that favored "mobile-friendly" websites over "non mobile-friendly" websites. You can use this tool from Google to check if your existing website is mobile-friendly.
4. As you begin building and testing responsive websites, be sure to install multiple browsers on your system and get into the habit of checking your pages and functionality on them. Use browser developer tools to help you troubleshoot your HTML, CSS and JavaScript. Use the website caniuse.com,to help you identify the level of browser support for any given feature you might want to use in your site, such as Flexbox or Grid Layout.
5. Front-end development and responsive web design require a certain fluency in basic math and understanding how to use common formulas such as "Target divided by Context = Result". This formula, for example, could help you how to convert a pixel-based measurement to an em-based measurement.
6. As designers and developers it is important to remember that we are not our users. Factors such as whether or not a user is on a mobile device, or has a slow connection or is using an older computer system all need to be considered. Additionally, you should be aware of whether or not your site has accessibility barriers and adapt your content accordingly.

ASSIGNMENTS

- Quiz

- This assignment will require a web browser, which you should have, and a smartphone, which you most likely have (if you don't have access to a smartphone you can use the Device Mode or Responsive Design Mode in the Developer Tools of your browser.)

  Go online and find at least one portfolio site for a web designer. You can do this by going to Google and using basic search terms such as CSS portfolio sites or web designer portfolio sites. Once you've found one you like, click around a few pages and note things such as layout, animations, how fast the page loads, and anything else that comes to mind.

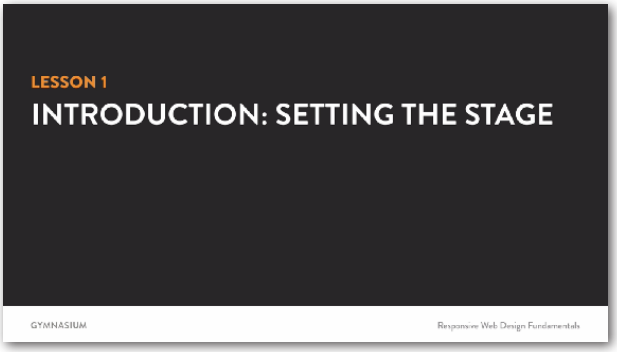
  Next, using your smartphone, or the Device Mode of your browser, go to the same website. How did the experience change? Is there a difference in the layout or how fast the pages load? Note down a few thoughts, and when you're done, head over to the forum for this class and post your comments in a new thread with the subject line "Lesson 1 Assignment".

  Additionally, go into the thread for another student's assignment, read what they had to say, and give them feedback on their submission.

INTRODUCTION

(Note: This is an edited transcript of the Responsive Web Design Fundamentals 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.)

Responsive Web Design fundamentals, lesson one. Introduction, setting the stage.



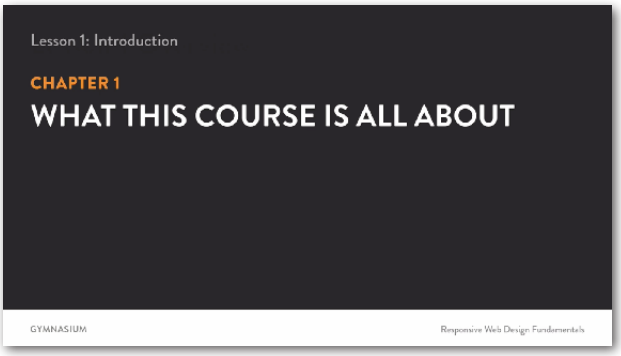
A little bit about me. My name's Jason Pamental, and I'll be your guide throughout this process. I've been working on the web for a little over 20 years in a range of different capacities, from creative director to technology manager for cell makers, NFL team, Ivy League universities, and a bunch of other corporations.

These days, I manage client strategy, design, and development over at a company called Isovera where we build websites for universities and tech corporations, mostly in the Drupal platform.

In this lesson, we're going to look at what the course is all about. Learn a little bit about what Responsive Design is. Look at some tools that we'll be using throughout the course. Talk about some technical conceptual and design considerations. And we're going to get into the idea of design translation. How do we take that static mock-up from a design and turn it into a fluid responsive website that works across all different devices and contexts?

CHAPTER ONE: WHAT THIS COURSE IS ALL ABOUT

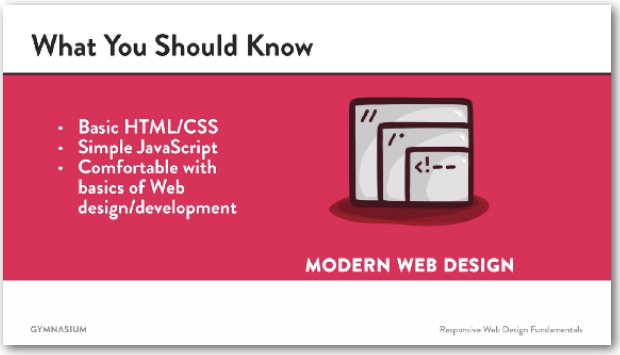
In this chapter, we're going to take a little bit of a look at you and see if you're the right person for this course. We're going to talk a little bit about, what you need to know in understanding design hierarchy. And we're going to get to know a little bit of the basics of and how they relate to a Responsive Web Design—things like HTML, markup and meaning. CSS, or the layout and style for your site. And JavaScript so we can get into behavior and finesse.



So is this class for you? Well, ask yourself, if you work on web-based projects? Are you a digital designer of some kind or do you want to be? Or are you a web developer? If you answered yes to any of those questions, then you are in the right place.

The only thing that might be a hindrance is if you have not gotten the basic understanding of HTML and CSS and JavaScript. In which case, you really should go back and take Aaron Gustavson's course on modern web design, which will be a fantastic introduction to all of the basics of HTML and CSS, getting into JavaScript, and getting comfortable with the basics of web design and development.

If you have any doubt, go check it out. It's a great class, and he's a wonderful teacher.



So what you'll learn in this class. We're going to get into a little bit of thinking in abstract relative hierarchies. We're going to talk about the basics of making your site scale and transform, and what that means. We're going to look at some new layout and performance techniques, like using Flexbox and loading web fonts for best performance. We're to look at CSS shapes, and even take a peek at CSS grid.

And finally, we're going to talk about thinking in parallel. We really want to make sure that while you're working on this, you're also thinking about your own portfolio site. You'll be making one for a fictitious person, but, obviously, we'd really love it if you end up with one for yourself at the end of this course.

Some things you won't learn. We're not going to get into using SAS, Less, or any other CSS preprocessors. We're really focusing on the fundamentals of learning to write things yourself.

We're also going to try and steer clear of JavaScript frameworks. We do use a little bit of jQuery, but mostly, we're going to be writing plain vanilla JavaScript.

We're also not going to use anything like Bootstrap or Foundation. We really want to focus on you learning things from the ground up. And we want to think about how all of these things work together before we start getting to know some of the shortcuts. Because after all, we had to roll the pixels uphill both directions when I grew up with our hands, and we liked it.

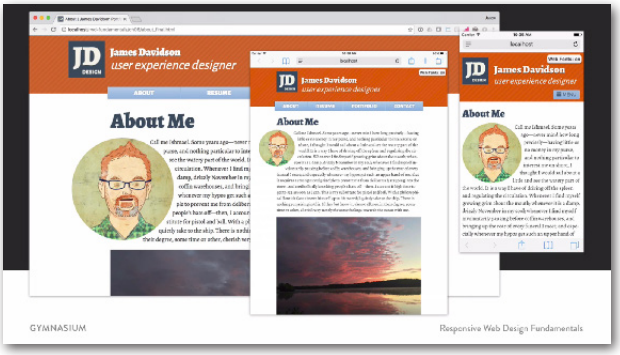
What we'll be building. We're going to make a portfolio site and that's going to have things like, a home page, a main portfolio page, a page about you, and a few other little bits and pieces here and there. And we've got some examples for you to look at from wireframes to design comps, all the way on through lots of sample code.

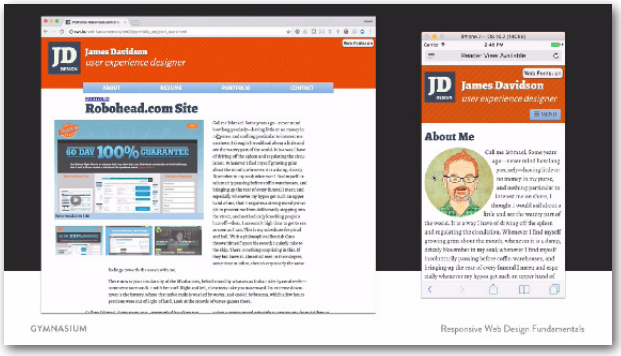
The lessons are going to be broken up. You're in lesson one right now so next, we'll have lesson two—looking at stretching and scaling, and the flow of your layout.

We'll look at going from a basic wireframe and adding in all of these different image and text elements and web fonts.

In lesson three, we're going to look at columns and Flexbox and responsive typography. And look at the future of layout on the web with some different layout ideas and techniques.

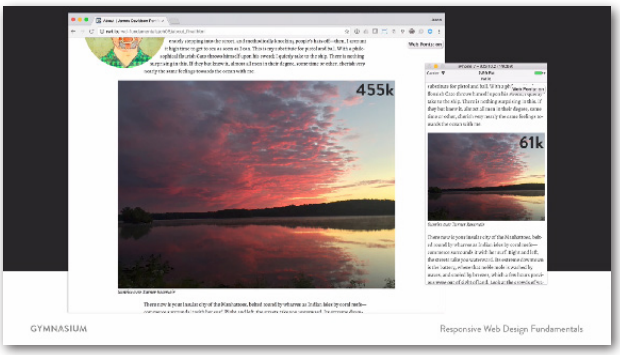
We'll be able to make sure that our type scales and our layouts reflow to work perfectly on every device you can try it on.





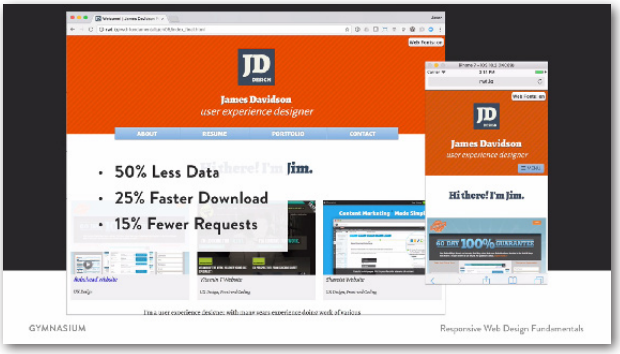
In lesson four, we're going to take a look at navigation patterns and context awareness. I want to make sure that you have the right kind of interface for each way people interact with your website. Looking at it on a desktop and what's natural and easy to use there, is not always the same as what's easiest on a phone.

In lesson five, we're going to take a look at response images, element queries, and the future of designing responsively. And what that means, and how it's different from that previous future we were talking about, which is really just focused on layout. Even something like responsive images, making sure you're serving up the right sized asset for the device that you're looking on, goes a long way to making sure that the experience actually shows up—as you can see here with that much smaller image being served on the phone.



And in lesson six, we're going to talk about the rest of the things with performance that really make a difference—both actual and perceived, and what that means. We'll talk a little bit more about mobile first. We'll talk about asset loading, and how you can control the timing of it. And then, testing. How do we make sure that it works across the widest variety of devices? And also, ensure that its living up to our expectations for speed.

When we're done, we'll actually be able to prove that our optimized site sends 50% less data down the line to the end user. It's going to load about 25% faster. And it has 15% fewer requests.

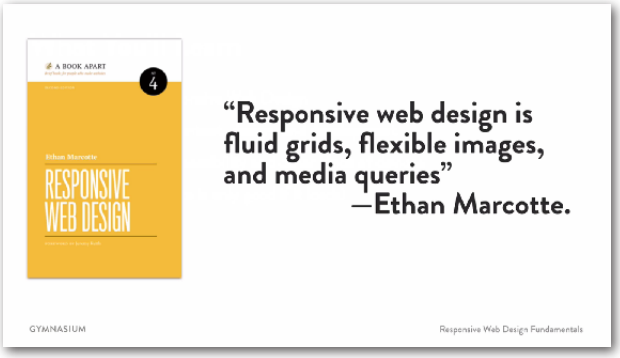


So we hope you know where we're headed, and that you're in the right place, and you're really excited and ready to go.

So throughout this course, we don't want you to forget that there's going to be a companion forum online, and they're going to be TA's there to help answer your questions. There will be quizzes and assignments as we go. And there will be a final exam. And a certificate, if you get it all done. Bonus points if you actually do this and make your own portfolio site. And hopefully, if you do, you'll share it with us so we can share it with the rest of the world.

CHAPTER 2: WHAT IS RWD ANYWAY?

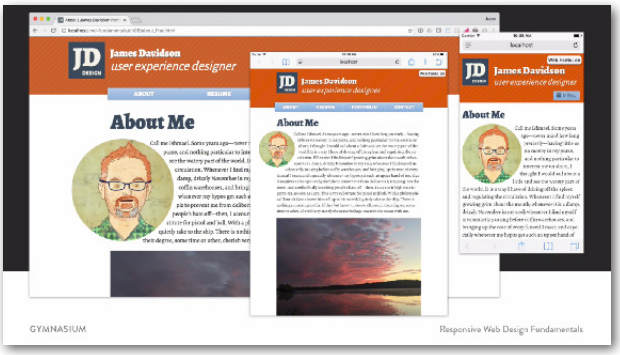
Responsive Web Design is made up of a few core tenets which we'll get into in just a moment. We're going to take a look at its impact on content design and development. Its relationship with accessibility and other kinds of devices. And performance. One of the things that's emerged in the last few years is that performance is really one of the most important design considerations when you're working on a site because it really only works well and is appreciated if it actually shows up.



So way back when, in the early depths of, I think, May of 2010, a fellow named, Ethan Marcotte, wrote an article on A List Apart about Responsive Web Design and followed it up—not long after—with this book. And he laid out the basic core tenets that really are the foundation of any responsive design.

Fluid grids, meaning that they don't have a fixed pixel width, but they scale. Flexible images that will scale with that fluid grid. And media queries which allow us to change that layout and how the website looks and behaves based on the size of the screen and the capabilities of the device upon which the site's being viewed.

And it really was a sea change in web design—at least it certainly was for me and many others. It really completely up-ended everything about how we made websites and how we thought about the web. And has been a complete revolution ever since.



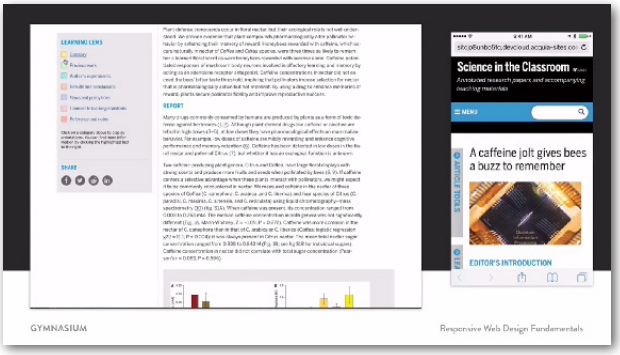
So again, coming back to the same screens we looked at before, bear in mind, it's one set of code that's rendering the page a little bit differently on each one of these different devices. We've got desktop, phone, and tablet.

And you'll notice that a few things change about how the layout, scales, and moves, and then, also, about how it behaves. You'll notice that there is a totally different menu there on the right, for the phone.

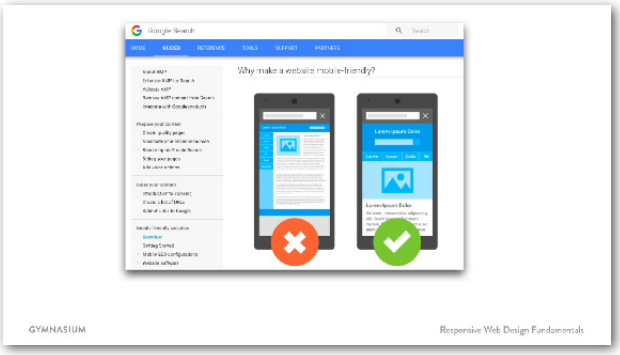
So a few things that have pushed us in this direction is the prevalence of smartphones, and how much they've grown in capability and use. The amount of time people are spending on their phones. And the number of users who are simply going to leave if it takes any great amount of time for that website to show up.

So on this website for Science in the Classroom, we have a desktop interface on the left that moves in scales and adapts to work better on the phone, to be more appropriate for the size of the screen and the way people

interact with it. So everything there is touch enabled, and also, keeps things in place for while you're browsing on the phone, but doesn't get in the way of taking in the content.



Another thing that came along just a year or two ago, was this notion of Google mobilegeddon. Getting One of the things that we know that our lives are ruled by as web designers and developers, is how our websites are viewed and ranked in Google. And when Google, a couple of years ago, decided that they were going to start changing the way websites ranked based on how well they performed—both in speed, but also, in optimization for small screens—that was really the straw that broke the camel's back in terms of acceptance of responsive design as the norm and not the exception.

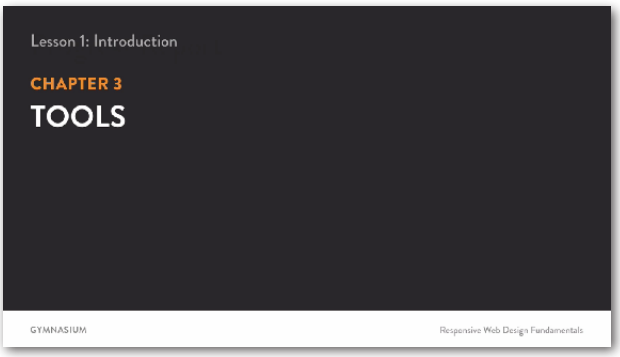


And just as further proof, there are also plenty of research studies that show that like this statistic from Google themselves, 61% of users are unlikely to return to a mobile site if they had trouble accessing it. And 40% are going directly to a competitor's site instead. If that's not a compelling statistic for you, your boss, or your client, I'm really not sure what else is.

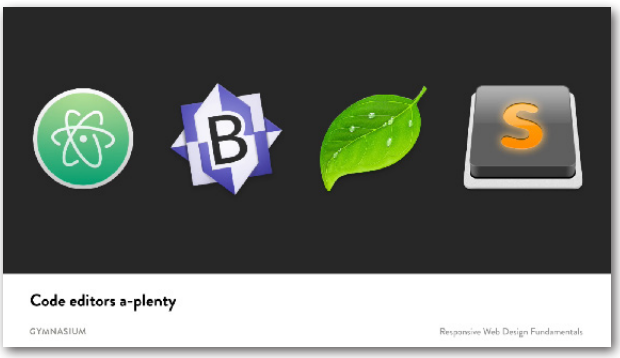
So we have a little bit of an understanding about Responsive Web Design. We've talked about its impact. We see some of the connections between Responsive Web Design, accessibility, and other devices so let's move on.

CHAPTER 3: TOOLS

What will need to work on this course. So there's a few things that you'll need. First off, code editors. Here are the icons for four different text editors I can recommend.



The first is Atom which we tend to recommend for Gymnasium courses because, A, it's free. But B, it's also available for both Mac and PC, which means the user interface is going to be very similar no matter which platform you're on. And this tends to make it a little bit easier to follow along with instructors while you're coding.



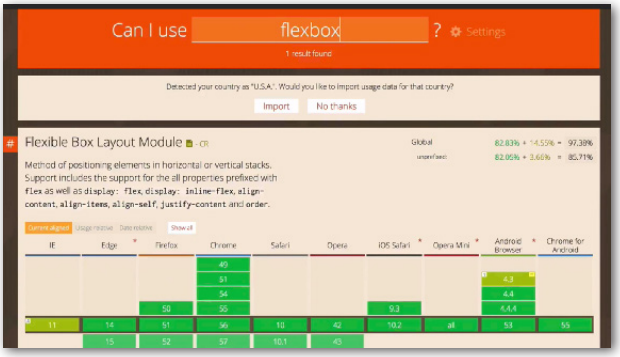
The second icon is BBEdit which is Mac only. And while there is a paid version with a full set of features, they also have a free version which has no real restrictions or nagging reminders, which is pretty generous.

Next up, there's Coda which is also Mac only, and is one of my personal favorites.

And last up is Sublime Text, which is extremely popular. Like Atom has the benefit of being cross-platform for Mac, Windows, and Linux, but unlike Atom, it does come with a price tag.

Web browsers—get all of them. What we've got featured here—Safari, Chrome, Firefox, Edge or Internet Explorer from Microsoft. The Edge browser's the newest one. It's really quite good. And opera.

The best thing to ensure that you have a website that works across all devices, is to make sure that you have at least two or three different web browsers installed, and make a habit of testing your code in all of those different browsers, just to make sure that you're not optimizing for something that works slightly differently from one to the next.

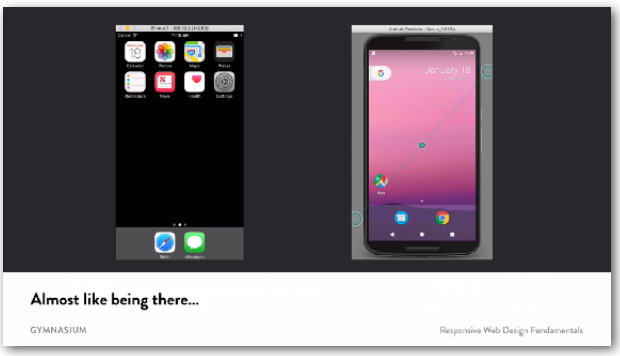


You also want to think about going to a website like, caniuse.com. So right here, I'm just taking a look at the compatibility of Flexbox. You can look up any kind of CSS attribute there and see how well it's supported. You can see that pretty much across the board, we can start to use Flexbox and be pretty confident that it's going to work.

Looks like the global compatibility with devices accessing the web is actually up over 97%. You do have to remember, that some percentage of that is with conditional support or it's only partially supported in Internet Explorer 11, but otherwise, it's in pretty good shape.

Developer tools are another thing I wanted to mention. All browsers have some version of these tools designed to make your life easier, and the general workflow is to right click or Control click on a page and this typically pops up a number of code panels which allow you to inspect the document. And do things like modify the HTML, CSS, and JavaScript. Analyze the network traffic on a page. Look at the sources of various files being used.

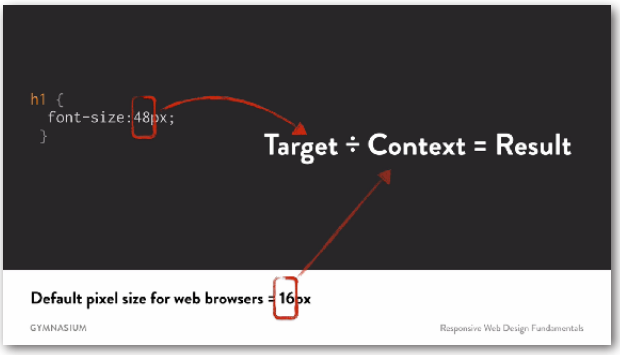
I'll be using these developer tools quite a bit in the upcoming lessons, and while I'll do my best to explain what's going on, if you want a more thorough introduction or a view of browser tools, let me plug this course that we have here on Gymnasium called, Build Better websites with Chrome Developer Tools, taught by Keith Peters. This will let you walk through a number of tips and tricks for making the most of these fantastic and quite frankly, almost necessary tools for front end web development.



Now, finally, emulators. I wanted to bring up, you can get free copies of Xcode from Apple, which comes with an iOS simulator, and you can download the Android SDK which has an Android emulator so you can actually run these right on your laptop or desktop, and open up a browser and see an exact replica of what it would look like on that device.

Finally, calculators. Here's a quick tip to get you started. Let's say, we have a heading which is set at 48 pixels and we want to convert those pixel units to ems. The code for that style might look like this.

Now we know that the default pixel size for web browsers is 16, and if you didn't know that previously, now you do. So we have two important numbers here, and the formula I refer to looks like this—target divided by context equals result.

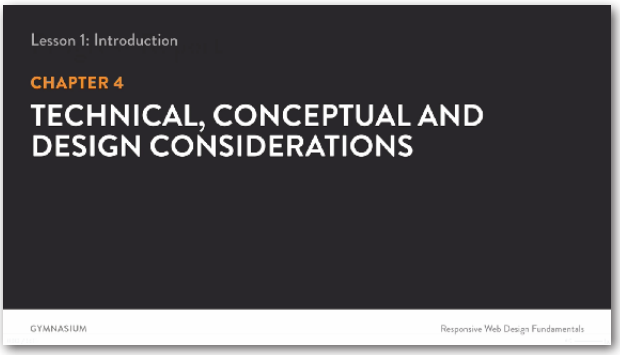


In this case, the target number is 48, and that's the number we want to convert. The context is 16, which is the standard size so the size of our body text. We plug those numbers in, we get 48 divided by 16, and the result is 3, which happens to be the value we want in ems.

Now don't worry too much about memorizing the formula. My larger point is that in Responsive Web Design, a certain facility with math is required. It's not a whole lot, but it's there. And more specifically, you should get used to having your calculator at hand. It's a tool you'll be needing every now and then in the real world.

So make sure you've got your code editor handy. Install all the browsers you can. Go grab all the emulators you can. And make sure you add a shortcut to your calculator—seriously.

CHAPTER FOUR: TECHNICAL, CONCEPTUAL, AND DESIGN CONSIDERATIONS

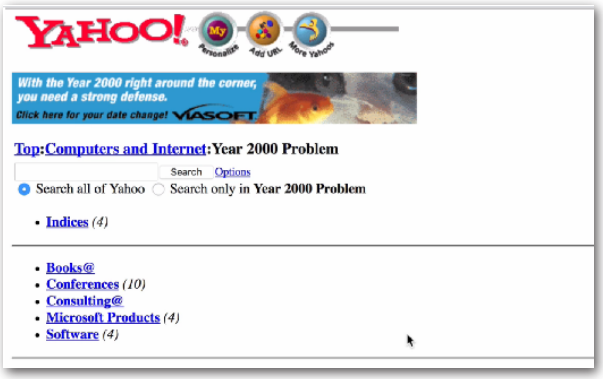


What you'll learn—first, think outside yourself. You have to remember that you're not your user. We also need to understand a new normal. One of the things that if you've come at this from more of a traditional desktop background, or you think about websites on a larger screen, designing for the web is more than just three screen sizes.

It's not just desktop, tablet, and mobile. Mobile first is increasingly mobile only and email is opened overwhelmingly on a mobile device. So if someone clicks on a link in an email, the chances are the first time they're going to see whatever you're linking them to is going to be on their phone. And then maybe if it's compelling enough, they might look at it on a larger screen.

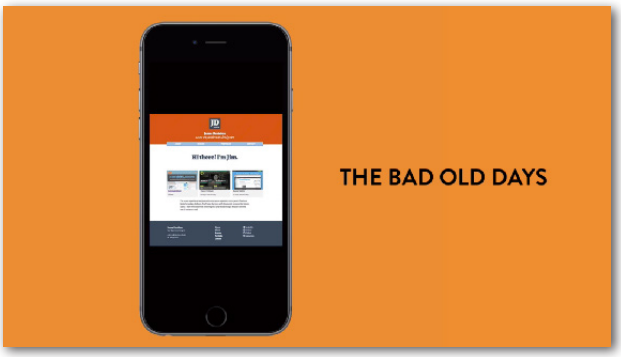
And next we're going to talk a little bit about how design really starts with markup. It actually starts with the HTML itself. And we have to think through the idea that meaning and intent have to be indicated in that markup before it can be conveyed visually. So this is something that I started doing in about 1994.

And I thought I was pretty clever, thinking I could save some space in the navigation bar by linking the logo back to the home page.



And it turns out I wasn't the only person who thought of that. And it's a pretty standard thing. Most web designers will do that instinctually. Even content management systems like Drupal do that by default out of the box. And yet, even today, after 20 years of browsing the web, 50% of the users you test probably won't know that that logo links back to the home page.

So we have to remember we are not our users. They're not always going to be as in-the-know as we are. So we have to make sure we can step outside of our own normal and put ourselves in their shoes much more thoroughly. So in the beginning with the first launch of the iPhone, there was really no notion of optimized for that small screen.



The only thing that we could do is basically just squash it down, so it shows on that small screen. And people would pinch and zoom. The advent of responsive design meant we could put something that's actually more appropriate on that screen. So not only is it scaled properly, so people can actually read what's there without pinching and zooming. But it also behaves in a way that the navigation will be a little bit more normal and a little bit more usable on that small screen.

So it's really all about context and being appropriate for the kind of device upon which it's being viewed. Another little fact is that over 31 million US internet users will only go online via a mobile device in 2016. And that number's only getting bigger. So we have—just to give you a general sense of it, that's accounting for just over 50% of the actual web access in the US. And it's even higher if you look outside in other markets around the world.

So we really have to stop and think it's not just mobile first. It may very well be mobile only. So when we're thinking about that and how we make sure that our meaning and intent is translated when the screen size keeps changing, we have to start thinking about proximity and scale, and not fixed pixel sizes. So here we've got a print publication from Monotype that's got this really wonderful lay out across this two-page spread, showing the main title of the piece.

And here are a few other spreads on the inside, so we can see just how this layout gets conveyed across these printed pages. Now, what we want to do with this printed pages is see how we can translate that design intent, and make sure it will actually work well and convey that same sense of style and importance and

meaning across headings and pull quotes and paragraph breaks and all that sort of thing, if that gets translated onto the screen. Now, here's a web page made out of that same design—the same web fonts being loaded.

And a lot of that same care to convey that same feeling expressed in those layouts, but now on the large screen and small. So we can see that we can do an awful lot to bring similar typography, similar visual elements, similar layout ideas that can translate across large and small screen, and give the same kind of feeling that we'd have across on that printed page. But we have that meaning—the H1 in the paragraph tag that's letting us know what those most important elements are. Then we layer on the typeface to bring that layout to life.

That way, once that story gets emailed, it gets posted to Facebook, or shows up on medium, we know that all of those same elements will still be just as important. The title will still be in H1. The introductory paragraph will still be just that little bit more prominent. And the rest of the article will break up in the appropriate way. So it will still be just as enjoyable wherever somebody finds it.

So for our progress report, the first thing we have to check for is our biases—make sure that we're not tripping ourselves up, making assumptions about what people know and are familiar with. We want to make sure that we understand what normal is. And that's another assumption. We can't be sure that someone is going to be behaving in some particular way just because they're holding a phone. It may be the only way they can get online.

And we want to connect visual and programmatic systems of hierarchy and meaning, so that our semantic markup, our H1s and H2s and paragraph tags are all connecting to how we visually represent things on the screen.

CHAPTER 5: DESIGN TRANSLATION

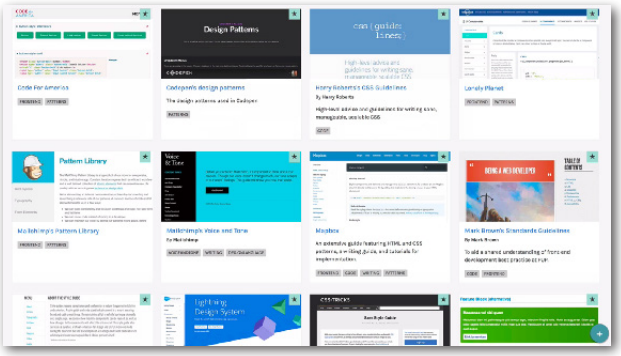
Now we go from concept to creation across all the different contexts of use.

So first we're going to review the design. We're going to translate that design from a fixed idea into a system of hierarchy. We're going to look at patterns and flow. And we're going to look from how this might move from static to dynamic, thinking about how well will this site work if the content changes tomorrow?

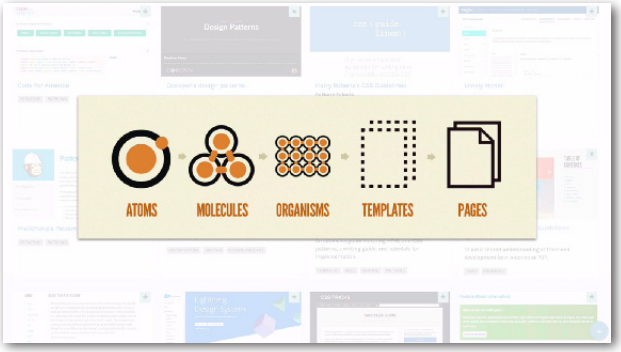
So we have some wireframes, and those will be available for you to download so you can take a look at them yourself. They're covering all the different pages on the website, from a Home page to the portfolio page to an

About page. And we have a design comp, so we have what we want it to end up looking like. And that's still something that's pretty static, but it gives us a lot of good pointers about style cues as well as how we'd end up wanting to break things up and lay them out on the page.

And then we need to get into flow. If it's going to move in scale across different screen sizes, we want to make sure that we have a good understanding of how we want that layout to move and change, just as you see it scale right here. And speaking of patterns, I want to talk a little bit about design patterns and pattern libraries, and this notion of breaking your design into different component parts and deciding how you want those component parts to behave and fit together.

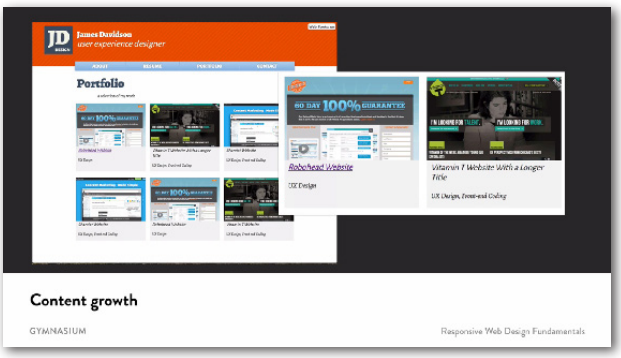


So on the screen here is a listing of different pattern libraries that were collected by Brad Frost and Anna Debenham on styleguides.io. And these are all based around a lot of similar ideas that Brad himself has written about in his system of atomic design, where he's thought about design patterns from an atomic level, where you are connecting individual elements together to form a molecule, like a form element that might have a label and a wrapping around it, pulled all together into an organism, which might be the whole form, all thrown into a template that might show you how you would apply all of those styles together for any kind of form to that specific content page that we'll have on your own site.



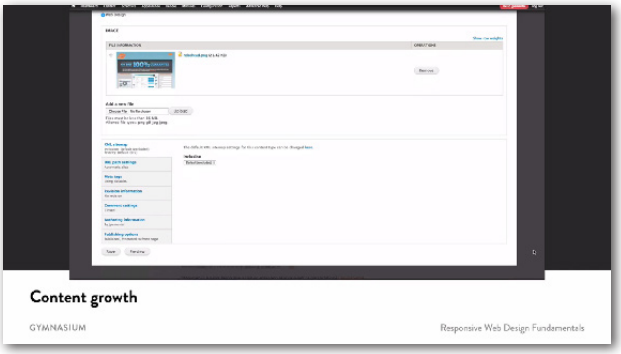
This is another example, a more detailed one, from the US Digital Service, which shows some of the UI components that they've put together for the US web design standards. So you can see a whole bunch of different examples of how buttons might behave, secondary buttons, how they would be styled, how you might style a table or different form elements. It's really an incredible library. We'll include links to some of these specifics in the notes.

Finally, content growth. Now you may be making a static website, or you might decide to put it into a content management system of some kind, or even maybe make your own. But you want to have these design patterns like we see here on the right and decide how those things come together and how they might move and change, and what are the changeable pieces as you go out building out your website.



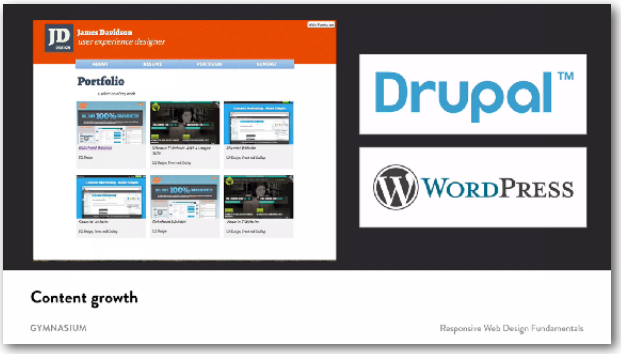
Now another thing I wanted to point out with this example is that you'll see one of these titles wraps on to a second line. One of the things that you have to anticipate on the web, especially with the way layouts move and scale, is that you're never going to create the perfect content that will always have exactly the same number of words on every line. So the reason we use techniques like flex boxes to help make our design systems more resilient. This is especially important

when you start to think about moving your content into a content management system and we'll take a look at how that might work.



And what I have here is a bit of a cheat. I've used in other content management system to flow in some of the content to show you what it might look like were you to put this into a content management system, and show how you would break it into a form that might allow you to upload that image and title and text separately, hit Save, and not have to worry about any of that markup.

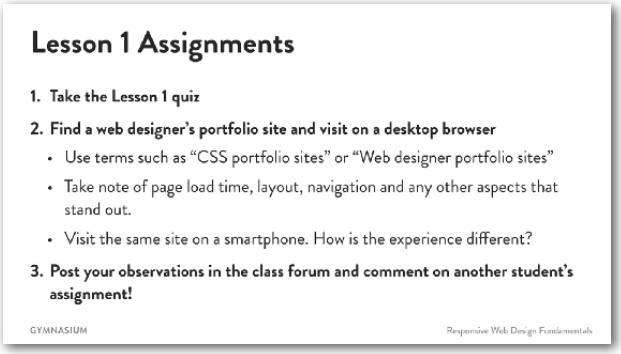
Now I happen to work with Drupal more than other content management systems. But there's also WordPress and many others out there that would be perfectly good alternatives to build a portfolio site. The whole idea with that content management system is that you have a database behind it so that like items, such as portfolio items that have images, text, a title, perhaps an external link, those things are all pretty similar. So having a system where you can add more of them and have it publish automatically for you certainly saves an awful lot of time and headache.



So there you have it. We've broken down the page. We've taken a look at some of the patterns throughout your design, and we've reviewed some of those patterns and layouts for trouble spots with changing content to make sure that we know where we might get tripped up with that extra long title.

Now let's turn our attention to assignments. In this course, the majority of your learning will take place as you follow along with my code and we build a responsive portfolio page for a fictional web designer together. In addition to that project, I'll be giving you other assignments designed to help you further your knowledge of the topics we've covered. So every lesson has at least two assignments, and we've already discussed the first assignment, which is the quiz.

Here's the second assignment for this lesson. It will require a web browser, which you should have, and a smartphone, which you most likely have. I'd like you to go online and find at least one portfolio site for a web designer. You can do this by going to Google and using basic search terms such as CSS portfolio sites or web designer portfolio sites. Once you've found one you like, click around a few pages and note things such as layout, animations, how fast the page loads, and anything else that comes to mind.



Next, using your smartphone, go to the same website. How did the experience change? Is there a difference in the layout or how fast the pages load? Note down a few thoughts, and when you're done, head over to the forum for this class and post your comments in a new thread like Lesson 1 Assignment. Go into the thread for another student's assignment, read what they had to say, and make a comment as well.

That's it for now. This will just get you started. In the next lesson, you'll begin learning the core concepts of designing modern responsive CSS layouts and start some serious responsive coding as well.