AQUENT
GYMNASIUM

RESPONSIVE WEB DESIGN:
BUILD A PORTFOLIO FOR ALL DEVICES

LESSON 7

ABOUT THIS DOCUMENT

This handout is an edited transcript of the Responsive Web Design lecture videos. There's nothing in this handout that isn't also in the videos, and vice versa. Some students work better with written material than by watching videos alone, so we're offering this handout 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. We encourage you to use this handout alongside the videos, rather than as a replacement of them.

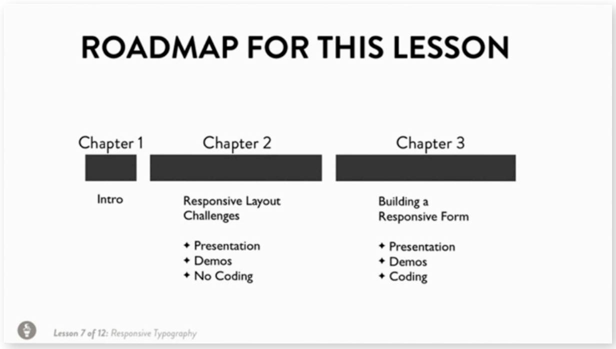
CHAPTER 1: RESPONSIVE LAYOUTS: STAY FLEXIBLE

GYMNASIUM

7. RESPONSIVE LAYOUT: STAY FLEXIBLE

Welcome to Responsive Web Design, a Gymnasium course brought to you by Aquent and Vitamin T. Responsive Web Design-- or Promote Yourself Responsively: Build A Portfolio For All Devices. This is lesson 7, Responsive Layouts, Stay Flexible.

As always, at the end of the lesson, there'll be an assignment and a brief quiz. Additionally, at any given point, be sure to use the pause button so that you can stop the demos and presentations and coding. And additionally, if you have any questions, be sure to hit the Forum. There we have TAs and instructors waiting to answer your questions. And of course, you have your other classmates as well.


ROADMAP FOR THIS LESSON

Chapter 1 — Intro

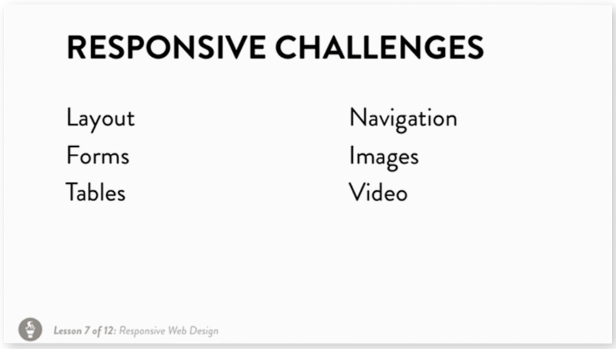
Chapter 2 — Responsive Layout Challenges
- Presentation
- Demos
- No Coding

Chapter 3 — Building a Responsive Form
- Presentation
- Demos
- Coding

Lesson 7 of 12: Responsive Typography

Let's talk a little bit about the road map for this lesson. We're currently in Chapter 1, the intro. Chapter 2, we're going to be discussing responsive layout challenges, and inside that chapter, you'll be looking at presentations and demos, but there will be no coding. Chapter 3, however, we'll be looking a little bit at presentation and demos, and there will be more coding. This is Responsive Web Design.

CHAPTER 2: RESPONSIVE LAYOUT CHALLENGES


RESPONSIVE CHALLENGES

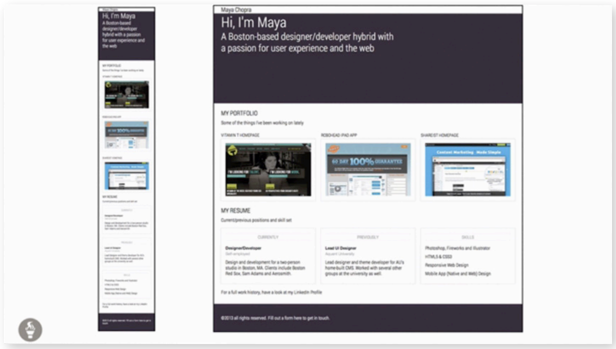
Layout	Navigation
Forms	Images
Tables	Video

Lesson 7 of 12: Responsive Web Design

In this section, we'll take a look at responsive layout challenges. The objective of this lesson is to present some strategies for the different layout challenges you will be encountering within responsive design. So first, let me just set the bar here since we're in a gymnasium. I want to talk a little bit about what we can achieve within this 30 minutes.

So of course, responsive challenges span the gamut of layout, forms, tables, navigation, images, and video. There's a lot of challenges that you're going to encounter as you begin to work with responsive websites. Now navigation, images, and video we can actually tuck away just a little bit. We're going to be covering these in upcoming lessons, specifically lessons 8 and 9. On the left-hand side here, we have some other challenges, layout, forms, and tables. In fact, these are all connected.

We're going to start with layout and we're going to talk about it in the context of what we've done already. So, here we have a snapshot of the designs that we've been creating up to this point. On the left-hand side, I have the thumbnail view of our mobile or smartphone view, and it's a single column, of course. And on the right-hand side we have our three column layout view for the portfolio.



Maya Chopra

Hi, I'm Maya

A Boston-based designer/developer hybrid with a passion for user experience and the web

MY PORTFOLIO

Some of the things I've been working on lately

VITAMIN T HOMEPAGE

ROBOHEAD IPAD APP

SHAREIST HOMEPAGE

MY RESUME

Current/previous positions and skill set

CURRENTLY	PREVIOUSLY	SKILLS
Designer/Developer Self-employed Design and development for a two-person studio in Boston, MA. Clients include Boston Red Sox, Sam Adams and Aerosmith.	**Lead UI Designer** Aquent University Lead designer and theme developer for AU's home-built CMS. Worked with several other groups at the university as well.	Photoshop, Fireworks and Illustrator HTML5 & CSS3 Responsive Web Design Mobile App (Native and Web) Design

For a full work history, have a look at my LinkedIn Profile

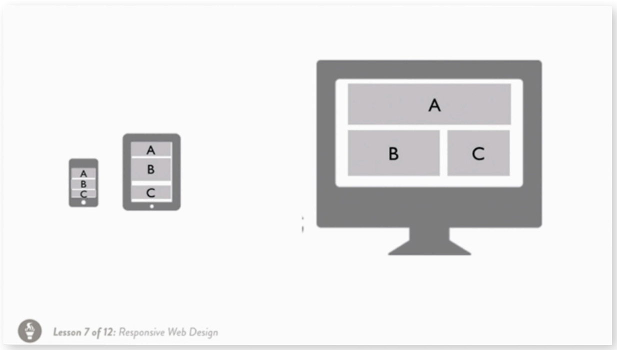
©2013 all rights reserved. Fill out a form here to get in touch.

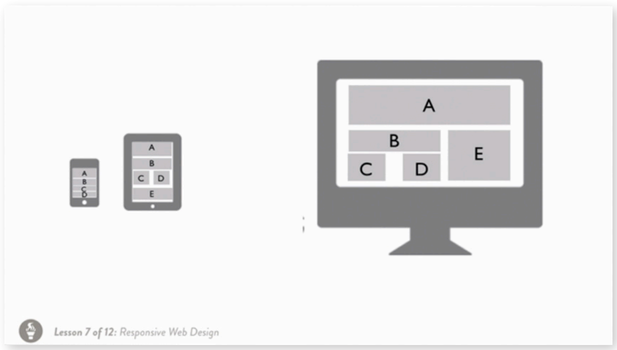
Of course, in between here we have some flexible layout going on. But the point is, this is one specific layout, and you'll be encountering different layout challenges down the road depending on different configurations of your page. So, I want to address that reality and give you some resources and ways to think about any sort of challenge that you encounter down the road. So we're going to start with some general layout solutions first, things that you'll likely encounter on virtually all responsive projects or decisions that you have to make. And then, in the next chapter, we're going to drill down to a specific layout solution, one in which we need to add a form to our pre-existing layout.

So, having said that, let's start a discussion about what a general layout problem in responsive design looks like. So, here in the right-hand side we have our desktop layout. We've got three modules here, A, B, and C. A might be the header, B might be the main content, C the sidebar. The question is how does this translate to a tablet and a smartphone view? The answer, in this case, might be pretty simple. We take A, B, and C, we put them in a single column, and this is going to work for both a mid-size screen, like a tablet, and the smallest screen, the smartphone.

But let's take this up a notch, and what about this layout? So A, B, C, D, E. This is much more complex layout, and let's think about how we would translate that. Well, on the tablet view, we might be able to get away with putting C and D next to each other because there's enough room, but on the smartphone layout, we're going to have to go back to that single column. And then we start to run into issues about priority and which content is the most important. So, all those things we have to start thinking about when we're dealing with different sized screens as well as different devices.

So, the good news is I'm not the first person to have thought of this, and there's a nice site that we can take a look at. This is at thismanslife.co.uk, Responsive Layouts, Responsibly Wireframed, and what this site attempts to do is to begin thinking about how desktop and mobile layouts might interact with each other, especially in terms of those content blocks that we've talked about.

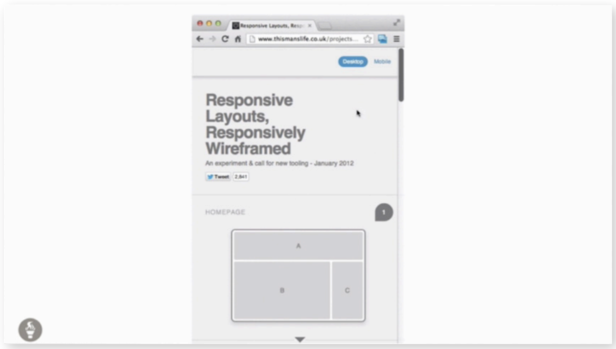




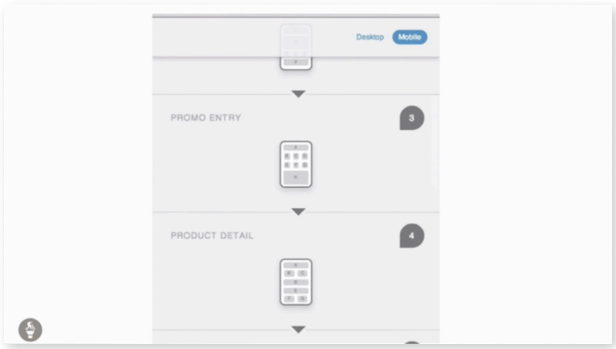
So, let's take a look at how the site works. We'll go ahead here and let's just take a look at the homepage, number one. So, we have A, B, and C, which should look very familiar. We'll click the Mobile button, and it drills down to a single column layout. And this is going to give us some idea of how we might be able to order our content. And then the nice thing about the site is it begins to increase the complexity pretty quickly. So we have different types of patterns, such as the promo entry. So you might find this on a larger site, not necessarily a portfolio site.

So the question is, how does this translate to mobile? Well, let's go ahead and toggle our button here, and we can see one solution. So, we have A as the header and then we could put B, C, D, E, F, G as thumbnails, and then H as the last content block. So, there's a couple other solutions on the page and I recommend you take a look at them. It's a nice way to start thinking about how to interpret desktop and mobile layouts.

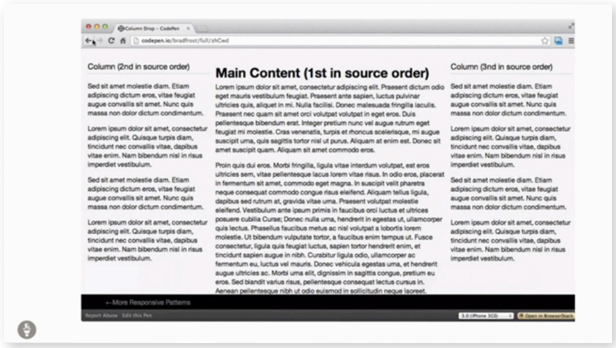
Let me point you to another resource. This is called Responsive Patterns. So, design patterns are a familiar term if you've worked in user experience or just in design and development. These are a collection of patterns that you can use to begin thinking about how to structure your pages. The URL for this particular page is there at the bottom. This is hosted by Brad Frost who is well known for his work in responsive design. It's a great site. Not only does it host patterns, but it also has resources, links, and news for all things responsive Web. So definitely check this out. Put it on your bookmark list.


Responsive Layouts, Resp
www.thismanslife.co.uk/projects...
Desktop
Mobile
Responsive Layouts, Responsively Wireframed
An experiment & call for new tooling - January 2012
Tweet
HOMEPAGE
1
A
B
C

I'll click on the first option here and, of course, the nice thing is that we can go ahead and begin to resize our browser window and take a look at how this content gets reordered. So, our main content is first, and then the rest of the content becomes a single column. So, this is fairly familiar, but let's go ahead and take a look at another one. This one we might term the column drop, and you'll see why in a second. We have two columns flanking our main column, and then what happens to that when we begin to resize?



Again, you have some choices here, but in this case the pattern says, well, the second column should stay on the right as the first column drops. Eventually, as this screen gets narrow enough, let's go ahead and drop them all to a single column. So, the point here is there are multiple patterns for layout and clearly we're not going to take a look at them all. This is for you. You can go ahead and take a look at these patterns, open up the code, get some developer tools, see what's going on, and let's just take a look at one more here.



Column (2nd in source order)

Sed sit amet molestie diam. Etiam adipiscing dictum eros, vitae feugiat augue convallis sit amet. Nunc quis massa non dolor dictum condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque turpis diam, tincidunt nec convallis vitae, dapibus vitae enim. Nam bibendum nisl in risus imperdiet vestibulum.

Sed sit amet molestie diam. Etiam adipiscing dictum eros, vitae feugiat augue convallis sit amet. Nunc quis massa non dolor dictum condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque turpis diam, tincidunt nec convallis vitae, dapibus vitae enim. Nam bibendum nisl in risus imperdiet vestibulum.

Main Content (1st in source order)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent dictum odio eget mauris vestibulum feugiat. Praesent ante sapien, luctus pulvinar ultricies quis, aliquet in mi. Nulla facilisi. Donec malesuada fringilla iaculis. Praesent nec quam sit amet orci volutpat volutpat in eget eros. Duis pellentesque bibendum erat. Integer pretium nunc vel augue rutrum eget feugiat mi molestie. Cras venenatis, turpis et rhoncus scelerisque, mi augue suscipit urna, quis sagittis tortor nisl ut purus. Aliquam at enim est. Donec sit amet suscipit quam. Aliquam sit amet commodo eros.

Proin quis dui eros. Morbi fringilla, ligula vitae interdum volutpat, est eros ultricies sem, vitae pellentesque lacus lorem vitae risus. In odio eros, placerat in fermentum sit amet, commodo eget magna. In suscipit velit pharetra neque consequat commodo congue risus eleifend. Aliquam tellus ligula, dapibus sed rutrum at, gravida vitae urna. Praesent volutpat molestie eleifend. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec nulla urna, hendrerit in egestas ut, ullamcorper quis lectus. Phasellus faucibus metus ac nisl volutpat a lobortis lorem molestie. Ut bibendum vulputate tortor, a faucibus enim tempus ut. Fusce consectetur, ligula quis feugiat luctus, sapien tortor hendrerit enim, et tincidunt sapien augue in nibh. Curabitur ligula odio, ullamcorper ac fermentum eu, luctus vel mauris. Donec vehicula egestas urna, et hendrerit augue ultricies ac. Morbi urna elit, dignissim in sagittis congue, pretium eu eros. Sed blandit varius risus, pellentesque consequat lectus cursus in. Aenean pellentesque nibh ut odio euismod in sollicitudin neque laoreet.

Column (3nd in source order)

Sed sit amet molestie diam. Etiam adipiscing dictum eros, vitae feugiat augue convallis sit amet. Nunc quis massa non dolor dictum condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque turpis diam, tincidunt nec convallis vitae, dapibus vitae enim. Nam bibendum nisl in risus imperdiet vestibulum.

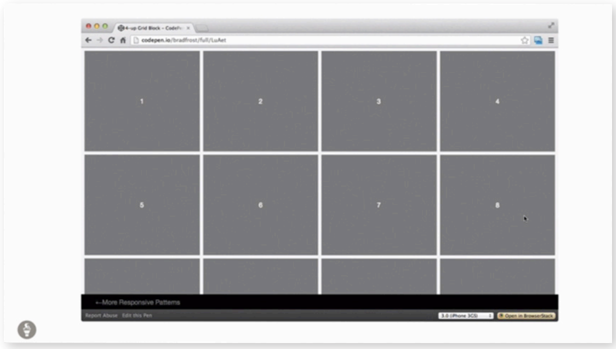
Sed sit amet molestie diam. Etiam adipiscing dictum eros, vitae feugiat augue convallis sit amet. Nunc quis massa non dolor dictum condimentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque turpis diam, tincidunt nec convallis vitae, dapibus vitae enim. Nam bibendum nisl in risus imperdiet vestibulum.

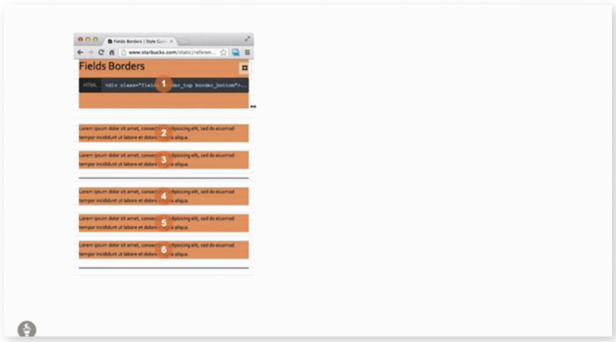
←More Responsive Patterns

Report Abuse Edit this Pen 3.0 (iPhone 3GS) Open in BrowserStack

So, in this case, we'll take a look at how to deal with a four-up gridblock. So, we have four columns here in our widest view. Of course, these are all flexible. At a certain point they jump to three, jump to two, jump to one, and, again, we could take a look at the code for that and see how it works. OK, so that's one path for solving layout problems. Let's take a look at another.



This is the online style guide. Now, style guides have been around for a while in the world of design, and if you haven't worked with a style guide before, it looks something like this. This is for Adobe, and this would show you, for example, where exactly to put the Adobe logo, how much distance it should be from the top of the page or the bottom of the page, or the space between the different logos, and it gives you all those specifications.



We have something similar for responsive sites online. So, this is not the only one, but this is actually a pretty good one. Pretty impressive. This is for Starbucks. So, Starbucks has an excellent responsive site, and they've put up their code with some visual aids to help everyone learn a little bit about how to make better responsive sites. That's pretty cool. Let's go ahead and take a look at how they do that.

So, I clicked on one link here for the field borders, and this is a multiple column layout that changes to a single column layout. This is kind of nice, but what's really nice is this button up top. So if you click here, you can then begin to toggle off and on some visual aids, such as, let's go ahead and take a look at baseline for this page, or we can turn on the boxes. I really like this feature here because now we can see these boxes-- one, two, three, four, five, six-- if we expand that browser window, we'll see how they re-flow, and we can keep track of them.

Additionally, let's go back up and let's turn on the grid, so we can see the grid structure for this. And we'll turn off the boxes in the baseline here, so we can just see the grid. So, we can now see that this is a six column grid. This looks familiar. We can now go ahead and resize this and see does this have fixed gutters or not? How does the content change? At what point does it change? If we want even more information, let's turn on Window Size, and for Window Size, we can begin to resize the window. We can see the measurements there in the top left corner, and we can even see the styles that are being used for each particular breakpoint.

So, this is a really great way to examine how someone else dealt with these issues of when to break from two column to three column, four column to five, and so forth. Now I'm really just showing you one or two here. Let's go ahead and take a look at another option which is under the promo layouts, and we'll just click on A here. So, this gives us a little more real world content. So we've got this three column layout here with the search fields, some place holders for images. If we go ahead and resize that again, we can see when it goes from three column, to two, to one. Again, some real world solutions to common layout problems.

So, coming up next, let's take a look at our layout problem or a general layout problem that we want to talk about. In this case, it has to do with tables. Responsive tables are a pain, and let me show you why. If we go ahead and begin to resize this very basic page, you'll see that at a certain point our table stops resizing. So, HTML tables have been around for a while, and they have this very stubborn property. At a certain point, they're going to stop reflowing. So, what you can see here is the user's going to have to scroll horizontally in order to see that table, and that's going to break the rest of the layout.

And even worse, let's take a look at how this works as seen by a smartphone. So depending on how your code is set up, really there's two choices. One, we are zooming all the way out so we can see the total width of the table. The problem here is the rest of the content becomes minuscule, and then you have to do the pinch and the zoom, or alternatively, you might have your code set up so that it doesn't zoom. In this case, the whole table is seen, but then we have our single-column layout at the top. Either way, this is not a good situation, so how might you deal with this layout problem?

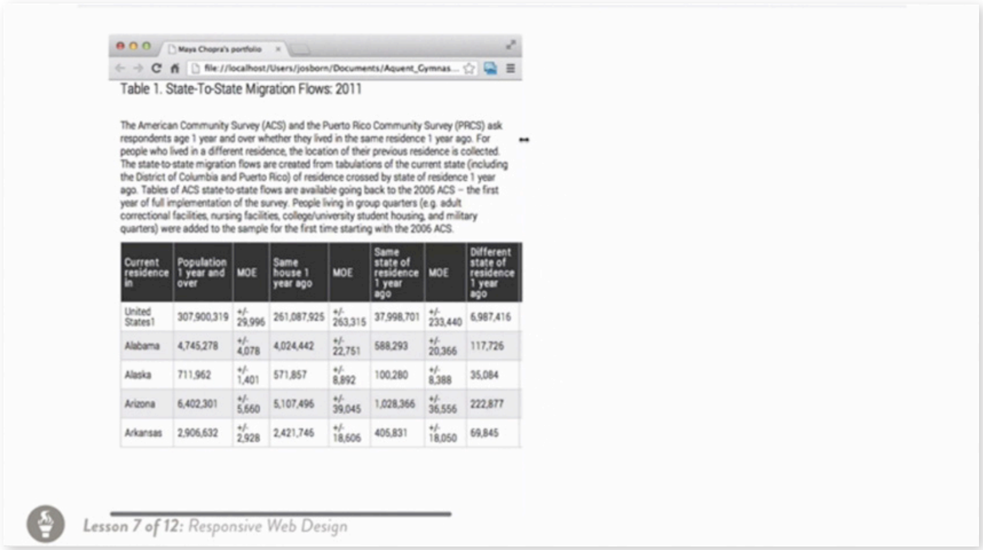


Table 1. State-To-State Migration Flows: 2011

The American Community Survey (ACS) and the Puerto Rico Community Survey (PRCS) ask respondents age 1 year and over whether they lived in the same residence 1 year ago. For people who lived in a different residence, the location of their previous residence is collected. The state-to-state migration flows are created from tabulations of the current state (including the District of Columbia and Puerto Rico) of residence crossed by state of residence 1 year ago. Tables of ACS state-to-state flows are available going back to the 2005 ACS – the first year of full implementation of the survey. People living in group quarters (e.g. adult correctional facilities, nursing facilities, college/university student housing, and military quarters) were added to the sample for the first time starting with the 2006 ACS.

Current residence in	Population 1 year and over	MOE	Same house 1 year ago	MOE	Same state of residence 1 year ago	MOE	Different state of residence 1 year ago
United States1	307,900,319	+/- 29,996	261,087,925	+/- 263,315	37,998,701	+/- 233,440	6,987,416
Alabama	4,745,278	+/- 4,078	4,024,442	+/- 22,751	588,293	+/- 20,366	117,726
Alaska	711,962	+/- 1,401	571,857	+/- 8,892	100,280	+/- 8,388	35,084
Arizona	6,402,301	+/- 5,660	5,107,496	+/- 39,045	1,028,366	+/- 36,556	222,877
Arkansas	2,906,632	+/- 2,928	2,421,746	+/- 18,606	405,831	+/- 18,050	69,845

Lesson 7 of 12: Responsive Web Design

I'm going to show you three possible solutions here. So, you're going to run into these type of solutions over and over again when you're dealing with responsive sites. So, it's not just tables, but it's virtually all layout. We have to think about these sorts of things. So, we've got lightweight, middleweight, heavyweight. Again, I want to point out that this is really not about which technique for tables is best, because fighting is bad. We don't necessarily want to do that. But what we want to do is show you, again, these three options in terms of complexity.

So the first one, the lightweight, is to simply hide less important content. Let me talk about what that means. We've got a CSS style at our disposal called display:none. And here we can see that the code, display:none, is being used for a class called table1, and this is being applied to a media query for screens less than 480 pixels. So you could think of this as, like, a smartphone view. What we're doing is we're turning off the table.

The main advantage of this is that it's really easy to do. That's all the code. You simply say display:none and the table doesn't show up. So, again, here is our table in the smartphone view. We add display:none, and it disappears. It's gone. Seems pretty easy, right? And it is, but there actually are multiple disadvantages.

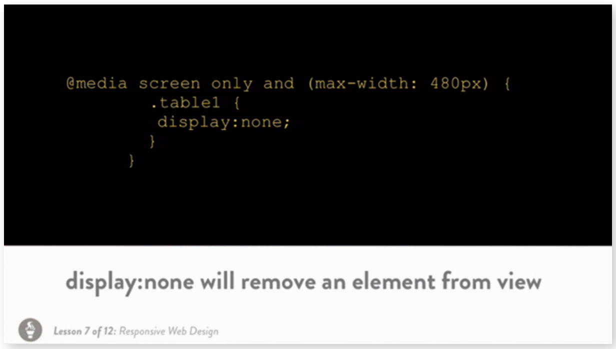
Option 1 (Lightweight):

HIDE LESS IMPORTANT CONTENT

Lesson 7 of 12: Responsive Web Design

First of all, when you use display:none, your content is still being downloaded. That's right. Display:none removes it from the CSS, but the HTML source is still present. And this potentially could affect performance, especially if you've got a lot of data inside of that content. Additionally, there are accessibility and possible SEO issues, or search engine optimization issues, when using display:none.

For accessibility, you're really removing this content from existence, and so a screenreader, for example, might totally ignore a table or other content.



```
@media screen only and (max-width: 480px) {
        .table1 {
         display:none;
        }
      }
```

display:none will remove an element from view

Lesson 7 of 12: Responsive Web Design

Possibly, the same thing with SEO, It confuses the matter if you're telling a search engine, such as Google, to ignore the content by saying hey, don't display it. But perhaps, more importantly, you could potentially frustrate your users. How? Well they're on the desktop site, they're looking at a table, and then they switch over to a smartphone or tablet and it's not there, and they're wondering what the heck is the problem.

And really, this boils down to this last point of not being mobile first. In an earlier lesson, where we talked about mobile first, we talked about setting priorities. So, if this content is truly important, then it should be in the mobile view and should we even have content that is never in the mobile view but is in the desktop?

So, let's take a look at another solution, the middleweight-- Maintain a Single Column Layout. So, you're going to see this technique used quite a bit as well, and what we do here is essentially collapse the entire table to a single column. In this case, the top row is always going to be in the first column here, and then all the subsequent rows are put into the second column. So, again, for the smallest screen or smartphone view, moving everything to a single column is a quite common tactic.

This code simplifies it somewhat. There's actually a little bit more to do, but, essentially, what you're doing is you're telling all the table elements to display as block, and this will put them into the single column. But there's other additional disadvantages. You have to take a little extra time to mark up your HTML and CSS. There is significant vertical scrolling that's now involved, and tables are usually used to compare values. If you want to learn a little bit more about the techniques of this one, you can go to CSS-tricks.com and take a look at this article, "Responsive Data Tables."

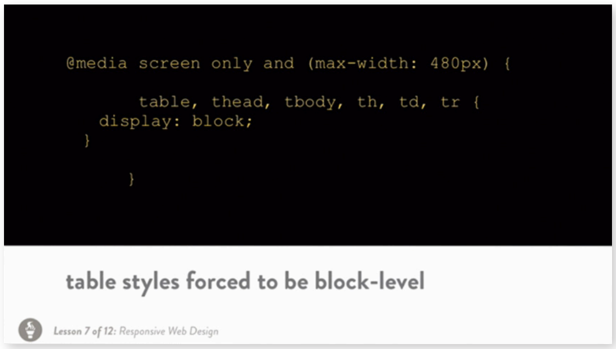
Option 2 (Middleweight):

MAINTAIN SINGLE COLUMN LAYOUT

Lesson 7 of 12: Responsive Web Design

But last, let's take a look at the heavyweight option. In this case, we're going to enable horizontal scrolling. So, again, here I'm going to show you an example. This is a table and layout solution provided at zurb.com where they're putting this code out for our use, which is very kind of them. And the way that this one works, is you simply need to provide a reference to the CSS and JavaScript here, and you can get that on the site, of course.

And then let's see how it works. We'll scroll down and we'll find this larger table. Let's resize the screen. We'll get into our smartphone view. What we can see here is that first column always stays fixed. However, we can now go in-- and there's a little horizontal scroll bar that will allow us to scroll within this frame, so this is kind of nice. This is a JavaScript solution, and again, we can see it works for a larger table as well.

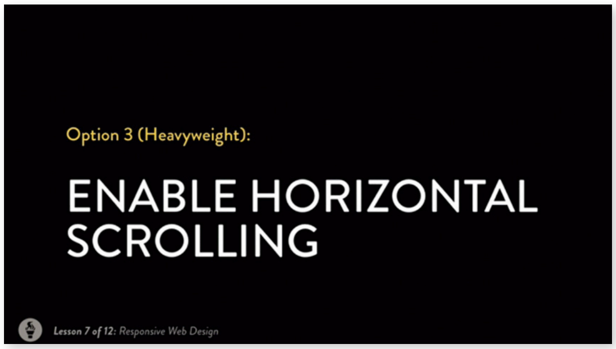


```
@media screen only and (max-width: 480px) {

        table, thead, tbody, th, td, tr {
   display: block;
  }

      }
```

table styles forced to be block-level

Lesson 7 of 12: Responsive Web Design

Well, again, a pretty acceptable solution. The disadvantages here, you need some external JavaScript and CSS files, so you're going to add extra weight to your page. And, of course, you've got that horizontal scrolling, and there's no two ways about it. Tables are going to affect your layout in some way, shape, or form, and a lot of this feels like CSS and JavaScript devilry, right? It's a lot of extra work just to make tables work. Well, this is the reality of responsive design. Right? We're going to have to use things like extra CSS and JavaScript in order to make them work in this new format.



Option 3 (Heavyweight):

ENABLE HORIZONTAL SCROLLING

Lesson 7 of 12: Responsive Web Design

And, in fact, this is really just the tip of the iceberg. For every table solution I showed you here, there's about two dozen others out there. So again, no matter what your layout challenge is, whether it be the number of columns on a page or how to deal with tables, I gave you some good resources for finding solutions. Again, one of them is here at the responsive pattern site, and this one happens to be for tables, specifically.

And next up we're going to take it out of the realm of abstract into specific. I'm going to give you a hands-on layout challenge where we have to add a form, in this case, to our pre-existing layout. This is Responsive Web Design.

CHAPTER 3: BUILDING A RESPONSIVE FORM

OK, in this section, we're going to be building a responsive form, and we're doing this in order to present a new layout challenge to our portfolio page. We want to add this form to the page, make sure it works within the context of different devices and screen widths, and we're going to go back in time a little bit because we're dealing with forms.

Forms have been around since the beginning of the web, and they haven't changed a whole lot. So just like tables, they're a little inflexible. They're not well suited for responsive design particularly, and so this is one of the challenges that we have to deal with. I should point out that we're really interested in the front end of the form. If you need to learn more about the nature of forms and the different components, I point you to this presentation by Aaron Gustafson: " Learning to Love Forms" a presentation that he did in 2007. It's a very excellent, in-depth presentation. We refer you to that because we do assume that you have some knowledge of the HTML and CSS styling of forms in general.

So having said that, let's talk about what we're going to do specifically. We're going to be adding some HTML markup to add the form to our page, and then we'll be adding some CSS styles to different media queries-- so one for our single column view or smartphone, another for the wider or tablet view, and then the third for the widest or the desktop view. As always, you're going to open your text editor, and then take this file, portfolio_start.html and choose File, Save As. And let's rename this portfolio_work.html. And this gives us a backup, so let's go ahead and save that.

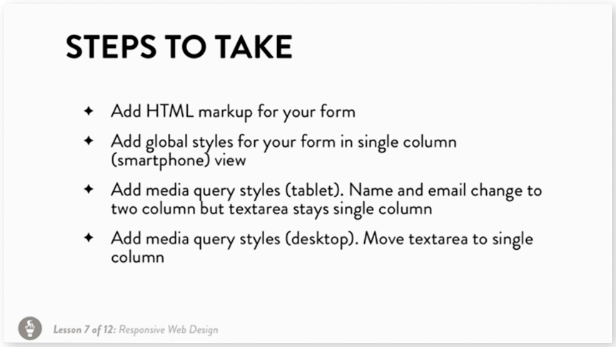
A couple of things, as always, between these lessons, I've done a few housekeeping duties here. I've taken all of the internal styles from the last lesson, and I put them into the style.css external stylesheet so that we can have a clean slate. And again, at a later point, we'll move these new styles to that external stylesheet. So having said that, we actually have two documents-- form.html and form.css that we'll be working with.

But first, let's open up this in a browser and talk about where we're going to put the form. So in our single column view, we'll go down and we're going to put the form right here above the copyright-- so right here where it says 2013 all rights reserved. As I mentioned before, we've got some code for you to copy and paste. Let's open up form.html. I'll go ahead and quickly show you this in the browser as well. It's a very simple form. It has name, email, and a small text area to say hello and then a send button.

LEARNING
TO LOVE
FORMS
WEB DIRECTIONS SOUTH '07

http://www.slideshare.net/AaronGustafson/learning-to-love-forms-web-directions-south-07

So you want to copy all of this code, and scroll down, locate this div id="container-footer", and Paste it, and Save your document. Additionally, we want to open form.css, and you want to take all this code. Although, be sure that you stop right here around line 82, before the media query. Copy that code, and put it in the internal stylesheet.

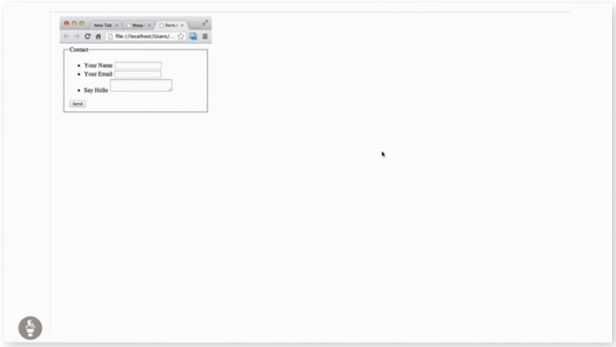


STEPS TO TAKE

- Add HTML markup for your form
- Add global styles for your form in single column (smartphone) view
- Add media query styles (tablet). Name and email change to two column but textarea stays single column
- Add media query styles (desktop). Move textarea to single column

Lesson 7 of 12: Responsive Web Design

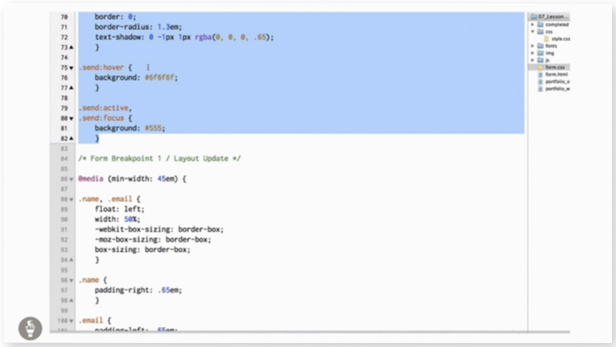
So again, there's a lot of things going on here, and we're going to talk about some of it, although not every single line. One thing I would like to point out here is, notice the values for padding for things such as the form and the list form. Remember that number, 1.3 and 0.65? That comes back to our magic number of 1.3 and the type of graphic scale that we talked about in previous lessons, but let's go ahead and save, and take a look at what this does in the browser.



So we'll save it and scroll down. And we've got a nice single column layout, so it's got a contact header, your name, your email, say hello, a Send button. Looks pretty good, right? We don't need our forms to be super complicated in this context. Let's just type in a name here, and we can see that this text input has been styled, so we're in pretty good shape. Let's go ahead, though, and expand our browser window. OK, this is not looking very good. In fact, this is looking ridiculous. So this form, not working so well in the widest desktop view.

So let's just take a step back for a second and talk about break points and how we want to address them. When responsive design first came out, we had these traditional breakpoints, for lack of a better word. These were generally related to devices that were popular when responsive design first came out. So the numbers associated here have everything to do with the width of devices, 320, 768, and then everything greater than 768.

In fact, if you go into a program like Dreamweaver-- so recent versions of Dreamweaver have added support for media queries, which is nice. However, they can actually give you some false expectations. So look what happens here when I click Default Presets. So, Dreamweaver is going to create these three different media query files, and they actually call them phone, tablet, and desktop, and they give them specific max- and min-width numbers, so in a weird way, an attempt to be useful, what this is actually doing is reinforcing what could be some bad habits.

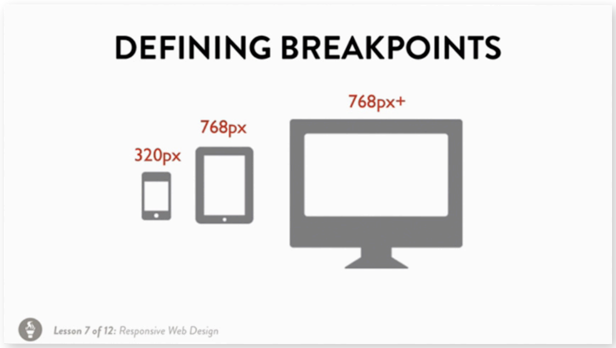


```
    border: 0;
    border-radius: 1.3em;
    text-shadow: 0 -1px 1px rgba(0, 0, 0, .65);
    }

.send:hover {
    background: #6f6f6f;
    }

.send:active,
.send:focus {
    background: #555;
    }

/* Form Breakpoint 1 / Layout Update */

@media (min-width: 45em) {

.name, .email {
    float: left;
    width: 50%;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
    }

.name {
    padding-right: .65em;
    }

.email {
```

The problem is we have all of these different devices now, and it's not really fair to say, well, phones should be 320, and a tablet should be 768. So, instead of using these fixed numbers, what we really try to do is look for organic breakpoints. At what point does the layout look crummy? And it's that point that we should change the layout. So, you might actually have three, four, or five media queries depending on what you're targeting, not just these two specific ones, tablet and phone.

So, having said that, let's go back to our code and add a media query. So, we'll go into form.css, and you'll need to scroll down a little bit if necessary. I want you to find this media query for min-width 45em, and we don't need to copy the whole media query. I do want to point out what's happening here. So, we have these two classes, name and email, and these are for the two text input fields, and they're floating left. And they both have a width of 50%, and they're using box-sizing border-box, which is a CSS3 property that we talked about in a previous lesson.

So this style block here is the core of what we're doing, and let's just flip over to the HTML for a second just so you can see what's happening. We have these two list items, and that's what this form really is, is a list item. And we've got the class="name" and the class="email", and these are the two things that are going to float. Additionally, we have this other list item for the message, and finally, we have the Send button at the bottom, which is a Submit button.



DEFINING BREAKPOINTS

Lesson 7 of 12: Responsive Web Design

So let's go ahead and copy all those styles within that media section block. And again, we don't have to copy the whole media query section because we're going to go to our external stylesheet. And we're going to find this section of styles.css for the media queries of screens wider than 45em or 720 pixels. So, let's go ahead and paste that code here after our other styles. Let's Save our document, and take a look at what this does.

OK, so we've got our browser window here, and again, this is the widest view and it's not looking so great, but notice that it begins to look better the more narrow it gets. So again, let's start from the single column smartphone view, slowly expand, and at a certain point, at 720 pixels or 45em, we're going to flip over here to the two column name and email. That text area section stays at 100% width, which makes sense because we might have people who are going in and typing long blocks of text, but let's go ahead and make this wider and wider. At a certain point, that text area looks a little ridiculous. It might be fine on a tablet, but not so much on the desktop.

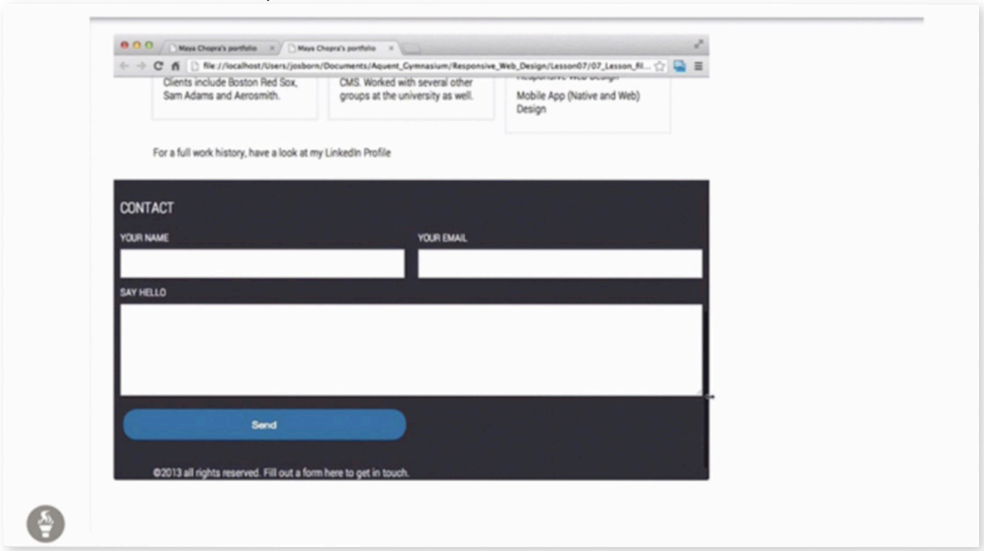
So we're now going to add a new media query specifically for the form and the form only. To do that, flip back to form.css, scroll all the way to the bottom, and let's take this entire block now, because we don't have the pre-existing media query for this min-width. So go ahead and copy all that code, flip over to your style.css, and let's paste it after the first media query.

Now, just for good measure, let's scroll up. Let's take this commenting code here for the first media query, copy it and paste it above here as a header. We just want to keep things nice and labeled and organized. Let's update this, so for screens wider than 60em, which is 960 pixels. Now, what's happening here is in addition to having padding, we're also adding clear: both; to this style rule for name and email. And just as importantly, we're putting a width of 50% on our text area. So let's go ahead and Save this page. Open this up in a new tab.

And what we can see now is that name and email are essentially on their own in a single column, and the text area is now on the right-hand side of those. So I feel like this is a pretty acceptable solution for our portfolio page. There's one little icing on the cake here. As I resize the window, I actually want you to look at the Send button in the lower left hand corner, because there's a certain style that's happening there that I want to talk about. So again, as we narrow that screen, see how that Send button will scale slightly. And, of course, in the single column view, it's at 100%.

So let's just examine the style of that Send button. In the widest screen layout, we're telling about Send button to have a 50% margin on the left and a width of 25%. In the middle media query, we also have the width of roughly 50%, so this is going to keep that Send button approximately half the size of the screen width no matter what. Of course, as we saw in the smartphone view, it's always 100% or approximately 100%.

So that's it. We've successfully integrated a form into our layout that looks pretty good. For the next lesson, we get graphic. When I say graphic, I mean images, of course, and specifically, responsive images. We're going to tackle what it means to be a responsive image, and how that affects things such as performance and editorial content, all sorts of cool things. As always, however, there's homework first. There's a short quiz for assignment number one in order to test your knowledge of this lesson and to reinforce the concepts.


Maya Chopra's portfolio
Maya Chopra's portfolio
file://localhost/Users/josborn/Documents/Aquent_Gymnasium/Responsive_Web_Design/Lesson07/07_Lesson_fi...
CONTACT
YOUR NAME
YOUR EMAIL
SAY HELLO
Send

Clients include Boston Red Sox, Sam Adams and Aerosmith.

CMS. Worked with several other groups at the university as well.

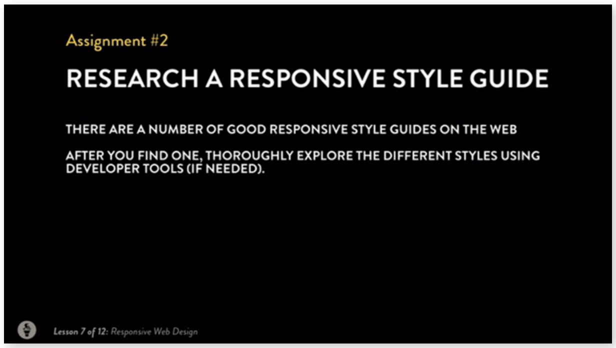
Mobile App (Native and Web) Design

For a full work history, have a look at my LinkedIn Profile

©2013 all rights reserved. Fill out a form here to get in touch.

Assignment number 2, I want you to research a responsive style guide. So we took a look at the Starbucks responsive style guides. There's a lot of other good ones on the web. After you find one of those, just take some time to explore that style guide. Look at it using developer tools. Figure out how it's working, and if you're feeling adventurous, go ahead and take some of those concepts and apply them to a copy of your portfolio page.

However, assignment number 3, the big one, we want you to add a contact form to your responsive site. So add support for at least two screen widths, just as we did here. Be sure to experiment with things like the form width, the input field width, the Send button width. Some of the things we talked about may not necessarily fit your layout, the one that you've been working on up to this point. Again, if you have questions with any of this, be sure to hit the Forum, and I'll see you in the next session.



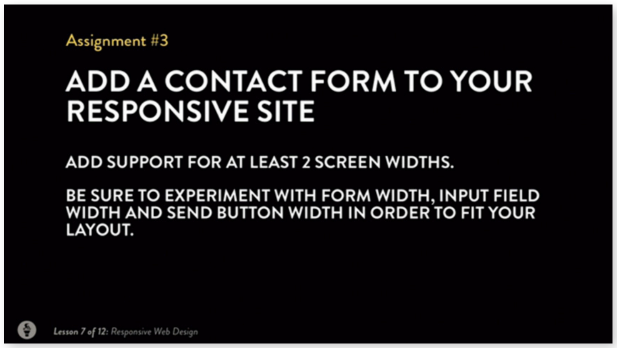
Assignment #2

RESEARCH A RESPONSIVE STYLE GUIDE

THERE ARE A NUMBER OF GOOD RESPONSIVE STYLE GUIDES ON THE WEB

AFTER YOU FIND ONE, THOROUGHLY EXPLORE THE DIFFERENT STYLES USING DEVELOPER TOOLS (IF NEEDED).

Lesson 7 of 12: Responsive Web Design



Assignment #3

ADD A CONTACT FORM TO YOUR RESPONSIVE SITE

ADD SUPPORT FOR AT LEAST 2 SCREEN WIDTHS.

BE SURE TO EXPERIMENT WITH FORM WIDTH, INPUT FIELD WIDTH AND SEND BUTTON WIDTH IN ORDER TO FIT YOUR LAYOUT.

Lesson 7 of 12: Responsive Web Design