MU Adobe Camp 2013

DESIGNING RESPONSIVELY with Dreamweaver CS6

james williamson | lynda.com

Howdy y’all

james williamson

lynda.com | senior author

@jameswillweb on the Twitter

Go SEC!

What is Responsive Design?

I mean, is it really a thing?

Responsive Design:

Designing with multiple screen sizes and clients in mind, so that you give the best experience to the widest array of clients and user agents possible. This includes fluid layouts, fluid media, and context-sensitive design.

Common Responsive Design concepts

Viewports

The current client window is often referred to as the "Viewport." The Viewport meta tag and the CSS @viewport rule allow us to control how devices size and initially display the viewport, which is critical for designing for mobile devices.

Common Responsive Design concepts

Fluid Grids

Layout grids that are designed to flex as screen sizes change. These are typically class-based selectors that only control layout and are defined using percentages.

Common Responsive Design concepts

Mobile First

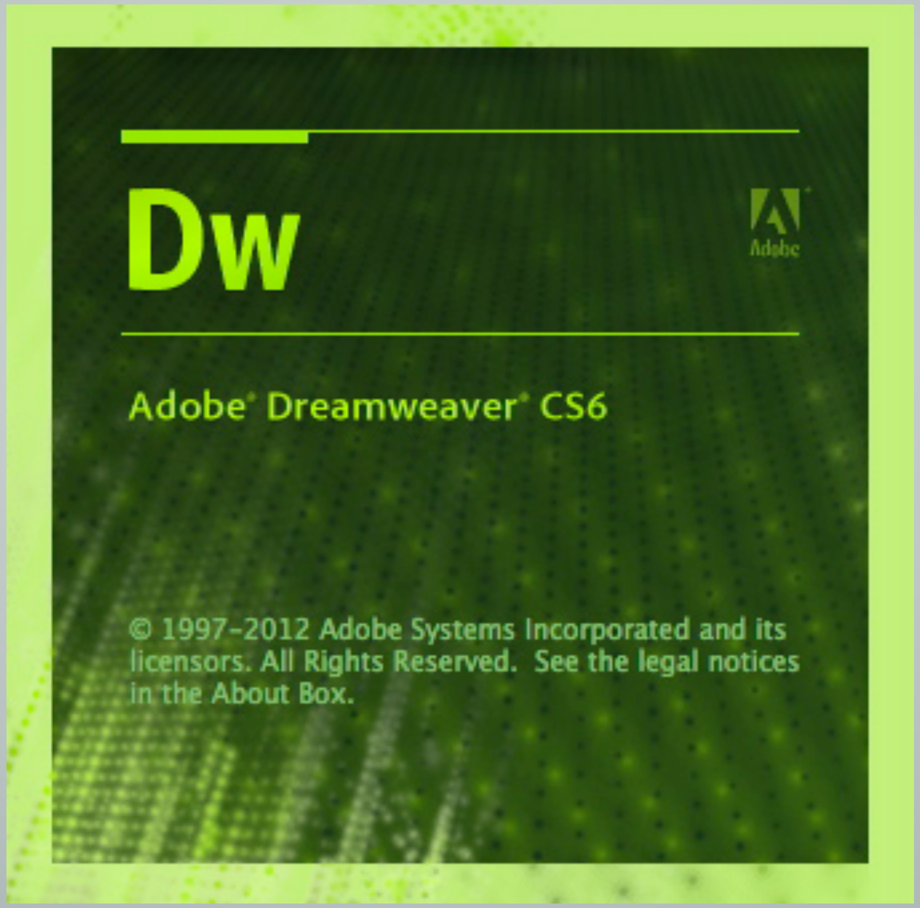
A way of approaching design that considers the mobile screen and context first as a way of focusing content and user experience. A better experience on mobile often results in a more streamlined experience on all devices.

Common Responsive Design concepts

Content Breakpoints

As screen sizes change, the available screen real estate often results in poor fitting content, broken layouts, or unintended layout errors. Content Breakpoints indicate a screen size that necessitates some change in layout or content presentation. Rather than focusing on device screen sizes, which are constantly changing, designers should rather focus on establishing content breakpoints, modifying their designs based on these breakpoints, and letting devices display whichever breakpoint they represent.

Fantastic. What does that have to do with Dreamweaver?



Dw

Adobe® Dreamweaver® CS6

© 1997-2012 Adobe Systems Incorporated and its licensors. All Rights Reserved. See the legal notices in the About Box.

Responsive Design tools in Dreamweaver

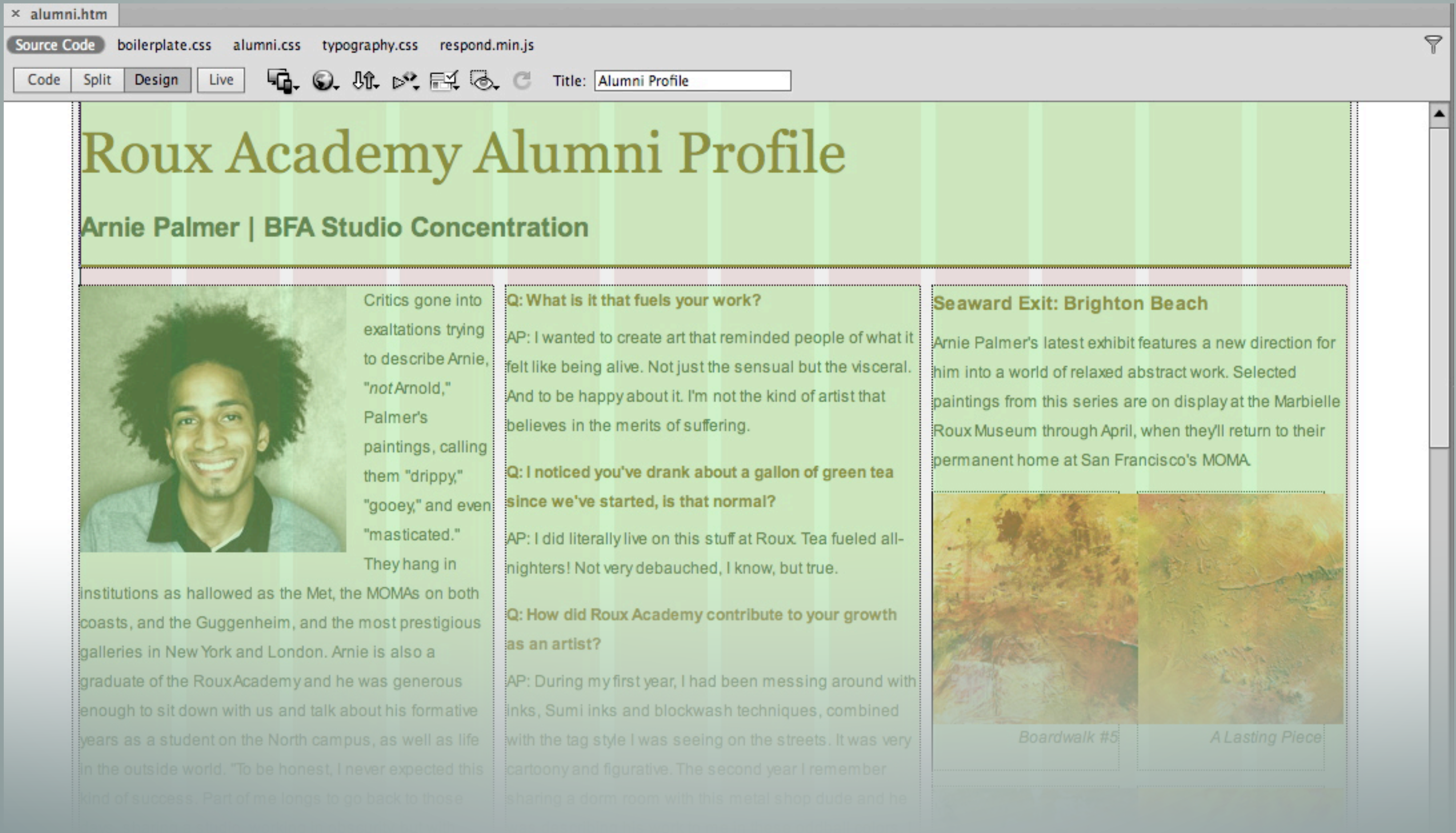
Fluid Grids

Media Query Support

Multi-Screen Preview

Responsive Live View resizing

Demo: Exploring responsive tools in Dreamweaver



Title: Alumni Profile

Roux Academy Alumni Profile

Arnie Palmer | BFA Studio Concentration

Critics gone into exaltations trying to describe Arnie, "*not* Arnold," Palmer's paintings, calling them "drippy," "gooey," and even "masticated." They hang in institutions as hallowed as the Met, the MOMAs on both coasts, and the Guggenheim, and the most prestigious galleries in New York and London. Arnie is also a graduate of the RouxAcademy and he was generous enough to sit down with us and talk about his formative years as a student on the North campus, as well as life in the outside world. "To be honest, I never expected this kind of success. Part of me longs to go back to those

Q: What is it that fuels your work?

AP: I wanted to create art that reminded people of what it felt like being alive. Not just the sensual but the visceral. And to be happy about it. I'm not the kind of artist that believes in the merits of suffering.

Q: I noticed you've drank about a gallon of green tea since we've started, is that normal?

AP: I did literally live on this stuff at Roux. Tea fueled all-nighters! Not very debauched, I know, but true.

Q: How did Roux Academy contribute to your growth as an artist?

AP: During my first year, I had been messing around with inks, Sumi inks and blockwash techniques, combined with the tag style I was seeing on the streets. It was very cartoony and figurative. The second year I remember sharing a dorm room with this metal shop dude and he

Seaward Exit: Brighton Beach

Arnie Palmer's latest exhibit features a new direction for him into a world of relaxed abstract work. Selected paintings from this series are on display at the Marbielle Roux Museum through April, when they'll return to their permanent home at San Francisco's MOMA.

Responsive Layouts

Create fluid layouts that target a range of screen sizes without targeting specific devices or screen sizes. Use media queries to respond to breakpoints that are planned around when content needs to change, not when a specific size has been reached.

The meta viewport tag

Meta viewport tags allow you to control the initial size of the viewport on mobile devices, ensuring a consistent experience

```
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

This syntax tells the device to set the viewport width to the actual device width to ensure the proper media query styles are used.

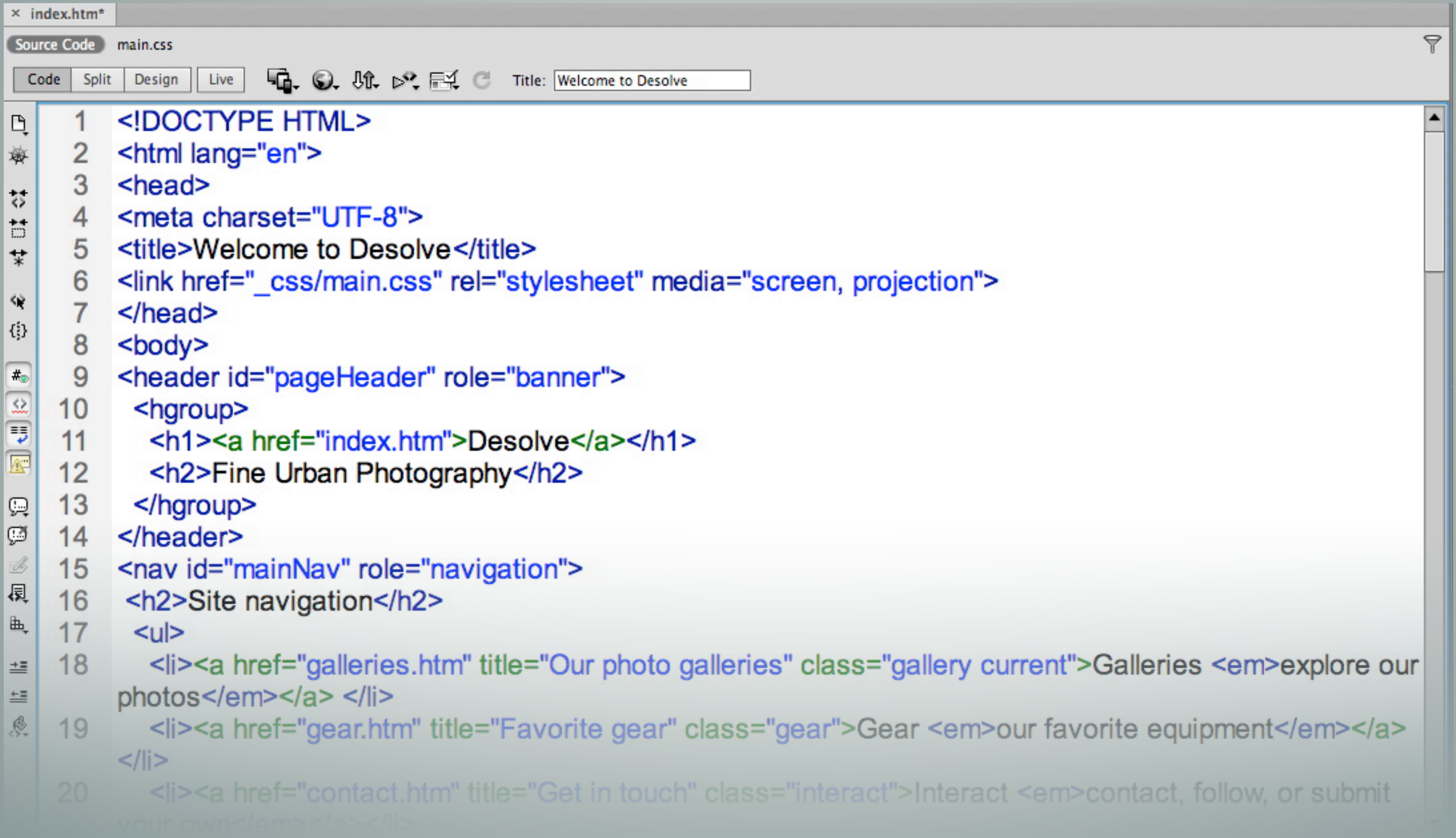
The @viewport rule

Controlling viewport functionality is coming to CSS as well. Support is increasing, but not universal.

```
@viewport {
    width: device-width;
    zoom: 1;
  }
```

This syntax also sets the viewport width to the actual device width and sets the initial scale of the page to 100%.

Demo: Setting the viewport



Title: Welcome to Desolve

```
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Welcome to Desolve</title>
<link href="_css/main.css" rel="stylesheet" media="screen, projection">
</head>
<body>
<header id="pageHeader" role="banner">
  <hgroup>
    <h1><a href="index.htm">Desolve</a></h1>
    <h2>Fine Urban Photography</h2>
  </hgroup>
</header>
<nav id="mainNav" role="navigation">
 <h2>Site navigation</h2>
  <ul>
    <li><a href="galleries.htm" title="Our photo galleries" class="gallery current">Galleries <em>explore our photos</em></a> </li>
    <li><a href="gear.htm" title="Favorite gear" class="gear">Gear <em>our favorite equipment</em></a></li>
    <li><a href="contact.htm" title="Get in touch" class="interact">Interact <em>contact, follow, or submit your own</em></a></li>
```

Media Queries

CSS Media Queries give us a way to control the application of styles based on the presence or absence of specific media features

Media query syntax

```
link rel="stylesheet" href="desktop.css"
media="screen and (min-width:500px)"


@media screen and (min-width:500px) {
  //styles go here
}
```

Media query syntax can be applied to media attributes in link tags, or inline within an existing set of styles

Media features

width*

height*

device-width*

device-height*

orientation

aspect-ratio*

device-aspect-ratio*

color*

color-index*

monochrome*

resolution*

scan

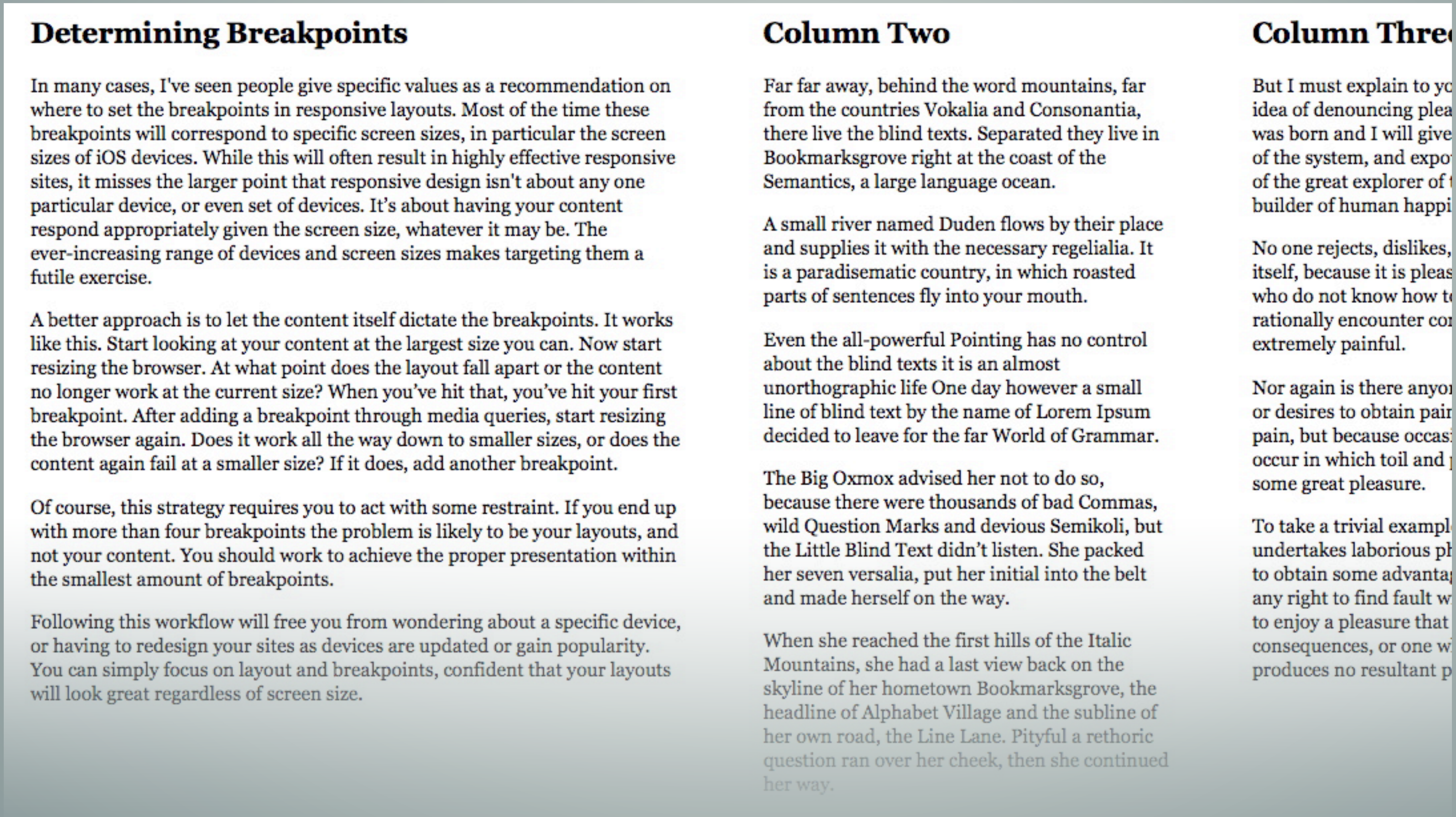
grid

*accepts “min-” & “max-” prefixes

Content Breakpoints

Content breakpoints are determined by using a fluid layout and then resizing the browser window. When the layout reaches a less than optimal state, or breaks, you assign a content breakpoint to that size. New layouts are constructed within that breakpoint, and the process continues until the desired range of screen sizes are accounted for.

Demo: Media queries and breakpoints



Determining Breakpoints

In many cases, I've seen people give specific values as a recommendation on where to set the breakpoints in responsive layouts. Most of the time these breakpoints will correspond to specific screen sizes, in particular the screen sizes of iOS devices. While this will often result in highly effective responsive sites, it misses the larger point that responsive design isn't about any one particular device, or even set of devices. It's about having your content respond appropriately given the screen size, whatever it may be. The ever-increasing range of devices and screen sizes makes targeting them a futile exercise.

A better approach is to let the content itself dictate the breakpoints. It works like this. Start looking at your content at the largest size you can. Now start resizing the browser. At what point does the layout fall apart or the content no longer work at the current size? When you've hit that, you've hit your first breakpoint. After adding a breakpoint through media queries, start resizing the browser again. Does it work all the way down to smaller sizes, or does the content again fail at a smaller size? If it does, add another breakpoint.

Of course, this strategy requires you to act with some restraint. If you end up with more than four breakpoints the problem is likely to be your layouts, and not your content. You should work to achieve the proper presentation within the smallest amount of breakpoints.

Following this workflow will free you from wondering about a specific device, or having to redesign your sites as devices are updated or gain popularity. You can simply focus on layout and breakpoints, confident that your layouts will look great regardless of screen size.

Column Two

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean.

A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth.

Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar.

The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way.

When she reached the first hills of the Italic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then she continued her way.

Column Thre

But I must explain to yo
idea of denouncing plea
was born and I will give
of the system, and expo
of the great explorer of t
builder of human happi

No one rejects, dislikes,
itself, because it is pleas
who do not know how t
rationally encounter co
extremely painful.

Nor again is there anyo
or desires to obtain pai
pain, but because occas
occur in which toil and
some great pleasure.

To take a trivial exampl
undertakes laborious p
to obtain some advanta
any right to find fault w
to enjoy a pleasure that
consequences, or one w
produces no resultant p

Fluid Layouts

Fluid layouts use percentage-based widths to control the width of page elements. This allows the page layout to adapt to the screen size, regardless of how large or small the screen.

Fluid Layout guidelines

Only the most basic layouts will work across all screen sizes

You can use **min** and **max** width values to establish ranges for fluid layouts

Designing for a range of screen sizes allows you to ignore specific devices or screen sizes

Fixed-width assets like images and video can complicate fluid layouts by causing them to break at specific sizes

Fluid Grids

Fluid grids are typically groups of class-based selectors that contain floats, varying percentage-based widths, and clearing properties. By creating a grid system layouts can be created by the simple process of applying class attributes to page elements.

Fluid Grids pros and cons

Easy to use

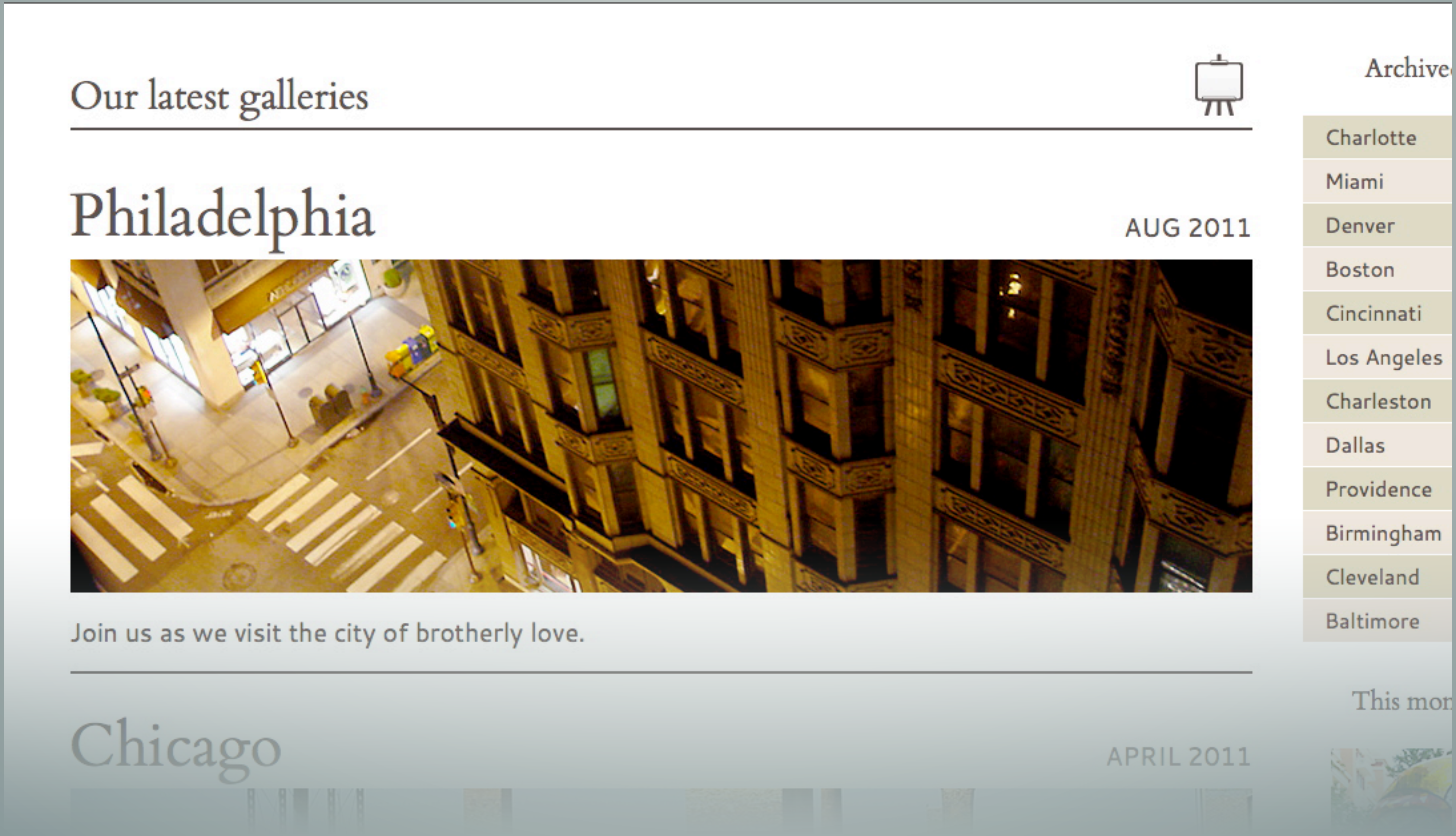
Many fluid grids already available

Can reuse them over and over

May add unnecessary weight

Use non-semantic class names

Demo: Building fluid layouts



Our latest galleries

Philadelphia

AUG 2011

Join us as we visit the city of brotherly love.

Chicago

APRIL 2011

Archive

- Charlotte
- Miami
- Denver
- Boston
- Cincinnati
- Los Angeles
- Charleston
- Dallas
- Providence
- Birmingham
- Cleveland
- Baltimore

This mon

Responsive Media

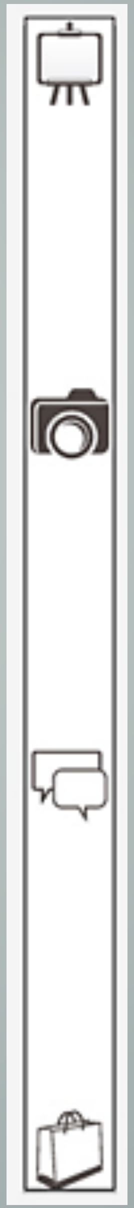
Fixed-width assets like media can be very challenging when designing responsive sites. Image or video sizes can break fluid layouts, and server requests, file size, and formats can make dealing with media in responsive designs tricky. While no definitive solution has been reached yet allowing for effortless responsive media, several techniques have evolved that allow media to work well within responsive designs.

Using background-images

If an image is purely decorative use a background-image instead, as requests for these can be filtered through media queries and only requested when needed.

Using CSS Sprites

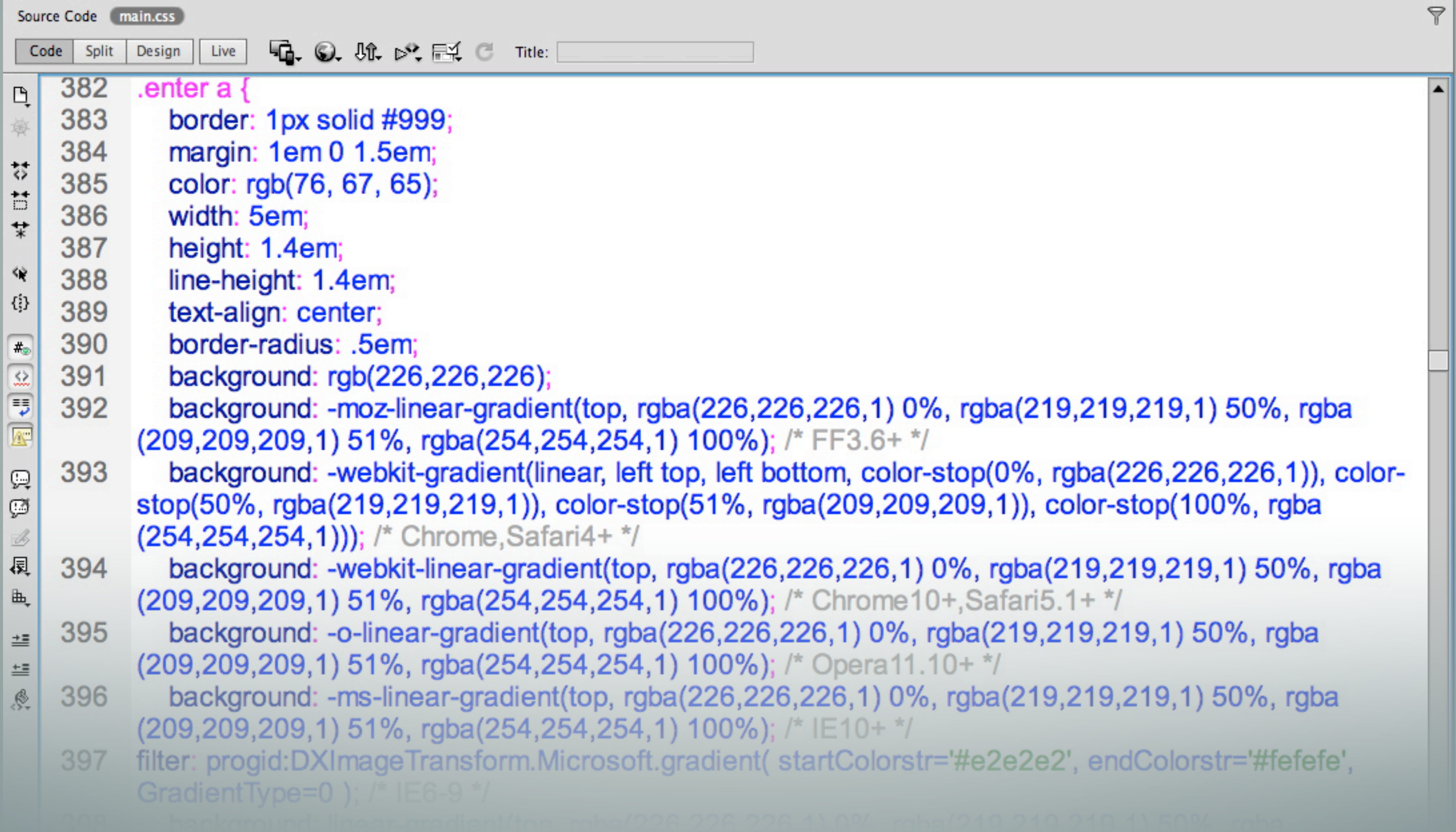
Using CSS Sprites can reduce http requests and make it easier to design assets



Using CSS in place of images

Often the best choice is not to use an image at all. CSS3 presents us with new tools that allow us to draw elements like background patterns, buttons, and icons that used to require images.

Demo: Using CSS for graphics



```css
.enter a {
    border: 1px solid #999;
    margin: 1em 0 1.5em;
    color: rgb(76, 67, 65);
    width: 5em;
    height: 1.4em;
    line-height: 1.4em;
    text-align: center;
    border-radius: .5em;
    background: rgb(226,226,226);
    background: -moz-linear-gradient(top, rgba(226,226,226,1) 0%, rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%); /* FF3.6+ */
    background: -webkit-gradient(linear, left top, left bottom, color-stop(0%, rgba(226,226,226,1)), color-stop(50%, rgba(219,219,219,1)), color-stop(51%, rgba(209,209,209,1)), color-stop(100%, rgba(254,254,254,1))); /* Chrome,Safari4+ */
    background: -webkit-linear-gradient(top, rgba(226,226,226,1) 0%, rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%); /* Chrome10+,Safari5.1+ */
    background: -o-linear-gradient(top, rgba(226,226,226,1) 0%, rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%); /* Opera11.10+ */
    background: -ms-linear-gradient(top, rgba(226,226,226,1) 0%, rgba(219,219,219,1) 50%, rgba(209,209,209,1) 51%, rgba(254,254,254,1) 100%); /* IE10+ */
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#e2e2e2', endColorstr='#fefefe', GradientType=0 ); /* IE6-9 */
    background: linear-gradient(top, rgba(226,226,226,1) 0%, rgba(219,219,219,1) 50%, rgba
```

Crafting flexible media

If you remove specific height and width values from images and video, you can set percentage based widths using CSS and allow the browser to scale the media appropriately.

This approach is not without it’s problems, and images and video often require different approaches.

Responsive image techniques

Remove height and width properties from HTML

Use percentage-based width values for scaling, browsers will automatically scale height based on aspect ratio.

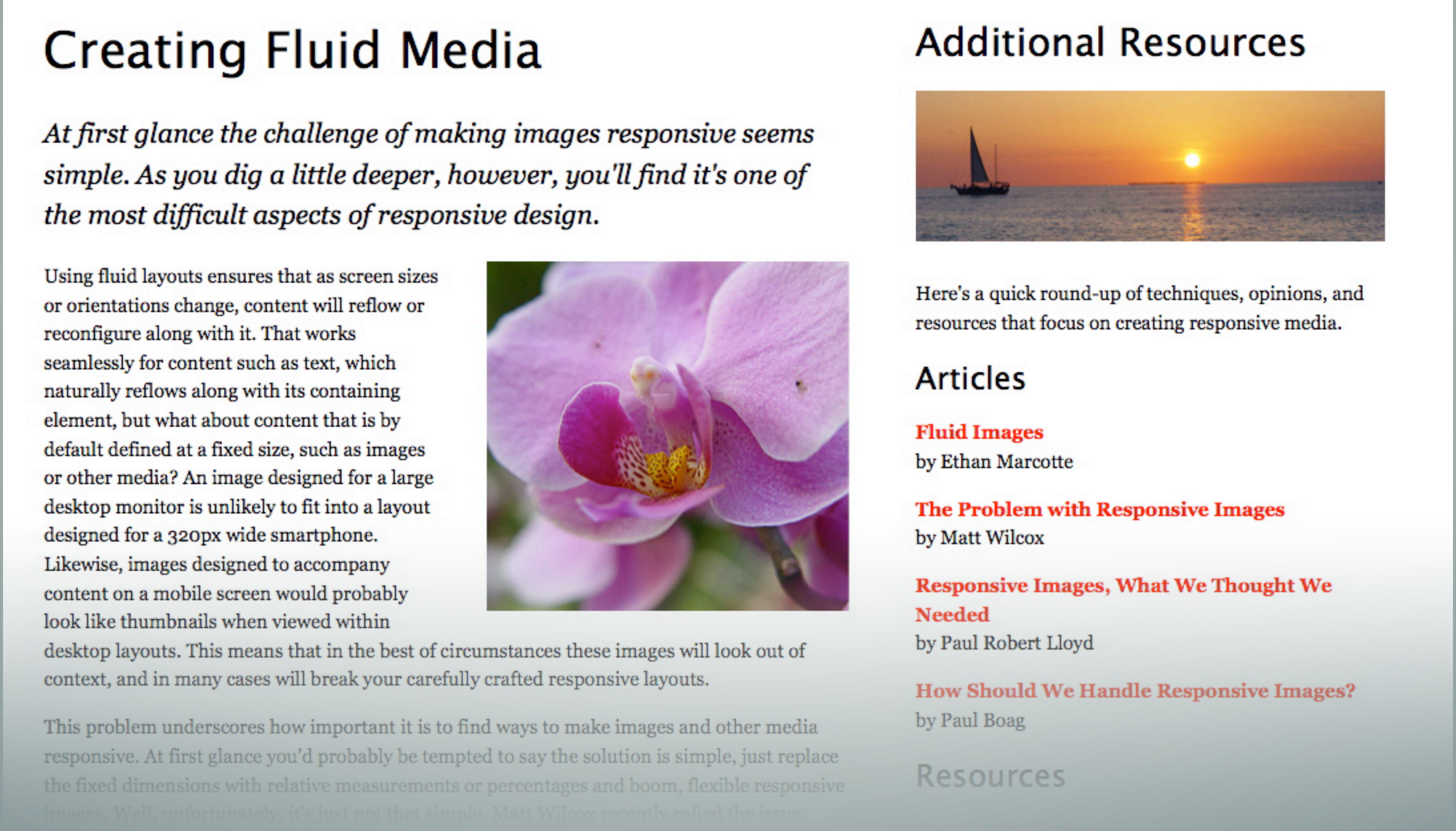
You can use min-width and max-width to set ranges for images

By placing the image in a flexible container and setting it’s max-width to 100% you allow the container to control the scaling

Since scaling images up degrades image quality you need to start with the largest size you need

This approach often requires mobile devices to download larger images than they need

Demo: Creating responsive images



Creating Fluid Media

At first glance the challenge of making images responsive seems simple. As you dig a little deeper, however, you'll find it's one of the most difficult aspects of responsive design.

Using fluid layouts ensures that as screen sizes or orientations change, content will reflow or reconfigure along with it. That works seamlessly for content such as text, which naturally reflows along with its containing element, but what about content that is by default defined at a fixed size, such as images or other media? An image designed for a large desktop monitor is unlikely to fit into a layout designed for a 320px wide smartphone. Likewise, images designed to accompany content on a mobile screen would probably look like thumbnails when viewed within desktop layouts. This means that in the best of circumstances these images will look out of context, and in many cases will break your carefully crafted responsive layouts.

This problem underscores how important it is to find ways to make images and other media responsive. At first glance you'd probably be tempted to say the solution is simple, just replace the fixed dimensions with relative measurements or percentages and boom, flexible responsive images. Well, unfortunately, it's just not that simple. Matt Wilcox recently called the issue

Additional Resources

Here's a quick round-up of techniques, opinions, and resources that focus on creating responsive media.

Articles

Fluid Images
by Ethan Marcotte

The Problem with Responsive Images
by Matt Wilcox

Responsive Images, What We Thought We Needed
by Paul Robert Lloyd

How Should We Handle Responsive Images?
by Paul Boag

Resources

Responsive video techniques

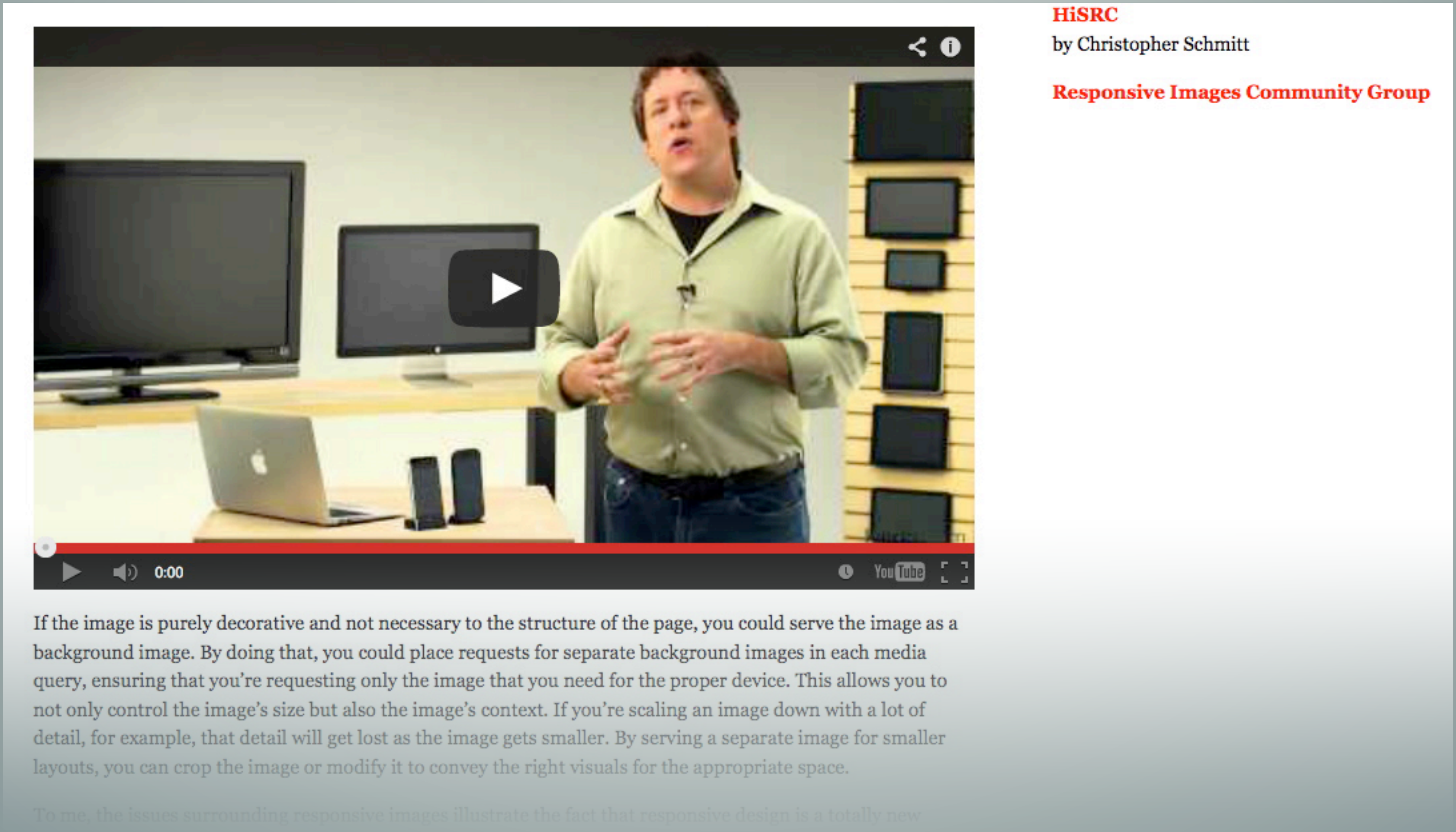
When dealing with native HTML5 video, you can often use the same techniques as images

When embedding video in iFrames, you'll need to adjust techniques

By wrapping the iFrame in a relatively positioned DIV, you can force the iFrame to scale along with the DIV

To give the DIV the same aspect ratio as the video, you can use a percentage based bottom padding (56.25% gives you 16:9)

Demo: Creating responsive video



HiSRC
by Christopher Schmitt

Responsive Images Community Group

If the image is purely decorative and not necessary to the structure of the page, you could serve the image as a background image. By doing that, you could place requests for separate background images in each media query, ensuring that you're requesting only the image that you need for the proper device. This allows you to not only control the image's size but also the image's context. If you're scaling an image down with a lot of detail, for example, that detail will get lost as the image gets smaller. By serving a separate image for smaller layouts, you can crop the image or modify it to convey the right visuals for the appropriate space.

To me, the issues surrounding responsive images illustrate the fact that responsive design is a totally new

Responsive Navigation

Perhaps the most challenging aspect of responsive design is that of navigation. Large menus can be unwieldy at smaller screen sizes and how the user expects to interact with navigation can change from one device to the other. These problems illustrate why having a clear navigation strategy is crucial to the success of any responsive project.

Responsive navigation strategies

Always build your menus with clean, well-structured code

Determine early on how complex your navigation will need to be

Think carefully about how users will want to interact with your navigation across various contexts

Think mobile navigation first

Pay attention to emerging mobile and touch navigation patterns

Realize there is not a one-size fits all solution

Styling responsive navigation

For simple menus, HTML and CSS alone will often suffice. By changing the orientation of menus for small screens you can often resize menus within acceptable ranges.

Keep in mind that touch areas need to be roughly 44px x 44px, make sure you resize icons or menu items accordingly

CSS Transitions can animate menu items and make it easier to hide and expand menu items on smaller screens. Be aware that hover events can cause problems on touch devices, and that transitions often take performance hits on mobile devices

JavaScript and responsive navigation

Often HTML and CSS alone will not be able to handle all of the requirements of responsive navigation. For many tasks you'll need to turn to JavaScript to add functionality and handle changing the menu over different devices.

Repositioning menus

Hiding/showing menus based on screen sizes

Adding/removing navigation elements

Responding to touch events & screen resizing

Replacing menus with **select** elements

The problem with touch events

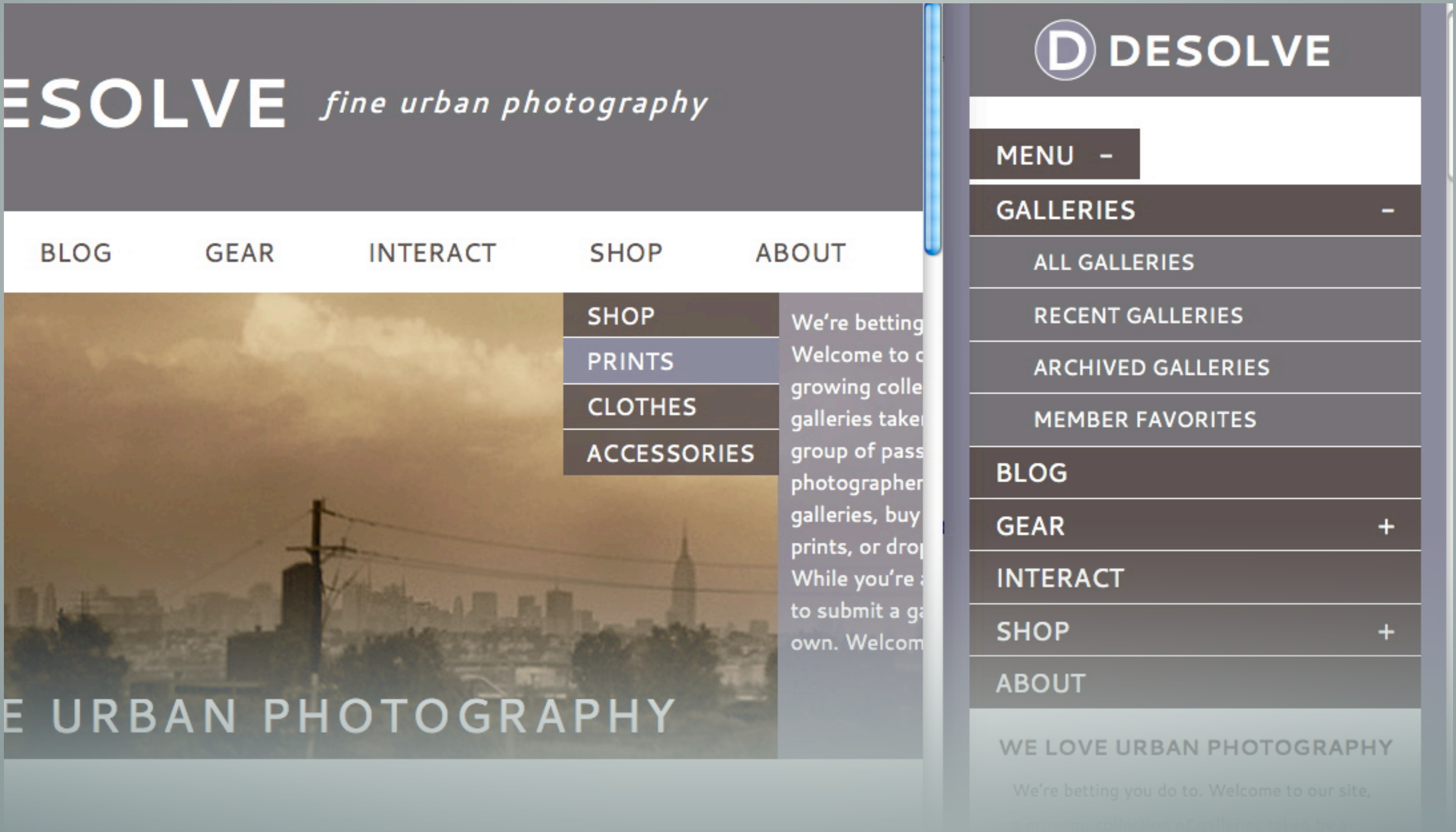
The presence of touch events no longer means mobile, so you can’t use the presence of touch events as a way of triggering mobile menus. Also, Windows 8 does not support the touch event standard, meaning you’ll have to write separate code to support touch for Windows 8 devices.

(for Windows 8 you use window.navigator.msPointerEnabled)

Be sure to provide fallbacks

Give users the best default experience first, and make sure that you’re not requiring JavaScript for menu functionality. Provide fallbacks for older browsers and non-supporting devices. Even if you can’t provide those users with an optimal experience, they still need to be able to access your site’s navigation.

Demo: Creating responsive navigation



We're betting
Welcome to c
growing colle
galleries take
group of pass
photographer
galleries, buy
prints, or dro
While you're
to submit a g
own. Welcom

We're betting you do to. Welcome to our site.

Taking advantage of mobile

Responsive design should be about more than just dealing with screen layout. Realizing that our sites will be viewed across multiple devices gives us the opportunity to take advantage of those device’s specific capabilities. Considering this very early in the planning stages of the site will allow you to provide the best possible user experience.

Designing for the mobile context

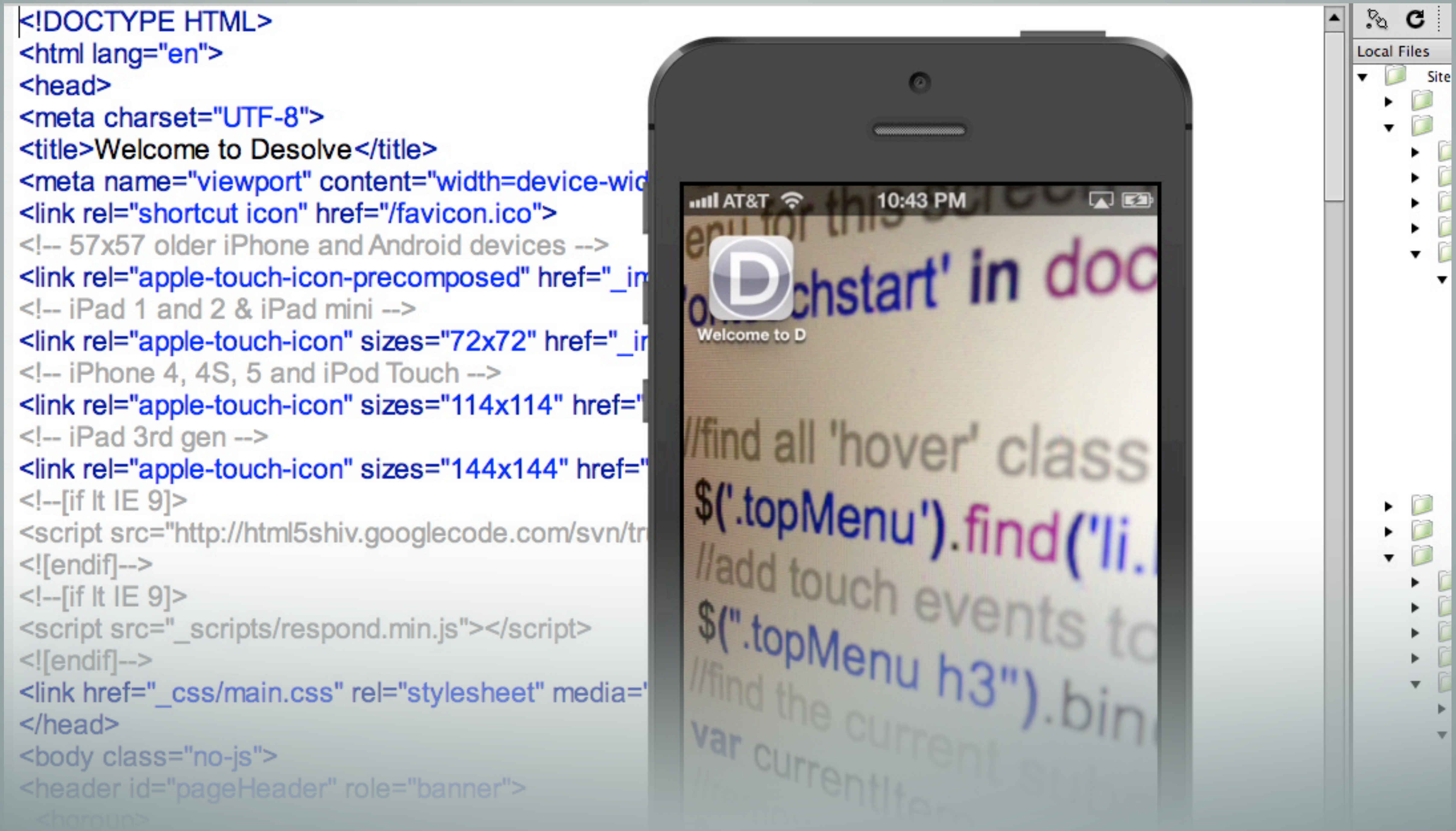
Use HTML5 form elements to take advantage of mobile contexts

Use the **tel** URL and SMS formatting to add phone functionality to your sites

Providing users with home screen icons allows them to load shortcuts to your site directly on mobile home screens

Using features like **Geolocation** and **localStorage** allows you to take further advantage of mobile functionality

Demo: Designing for mobile



```
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Welcome to Desolve</title>
<meta name="viewport" content="width=device-wid
<link rel="shortcut icon" href="/favicon.ico">
<!-- 57x57 older iPhone and Android devices -->
<link rel="apple-touch-icon-precomposed" href="_im
<!-- iPad 1 and 2 & iPad mini -->
<link rel="apple-touch-icon" sizes="72x72" href="_im
<!-- iPhone 4, 4S, 5 and iPod Touch -->
<link rel="apple-touch-icon" sizes="114x114" href="
<!-- iPad 3rd gen -->
<link rel="apple-touch-icon" sizes="144x144" href="
<!--[if lt IE 9]>
<script src="http://html5shiv.googlecode.com/svn/tr
<![endif]-->
<!--[if lt IE 9]>
<script src="_scripts/respond.min.js"></script>
<![endif]-->
<link href="_css/main.css" rel="stylesheet" media="
</head>
<body class="no-js">
<header id="pageHeader" role="banner">
  <hgroup>
```

Managing resources

Although you want to be device-agnostic when creating responsive layouts, the truth is you'll need to account for mobile devices when allocating resources. Having a strategy in place for how to load resources efficiently for mobile devices is critical to good responsive design.

Limit HTTP requests

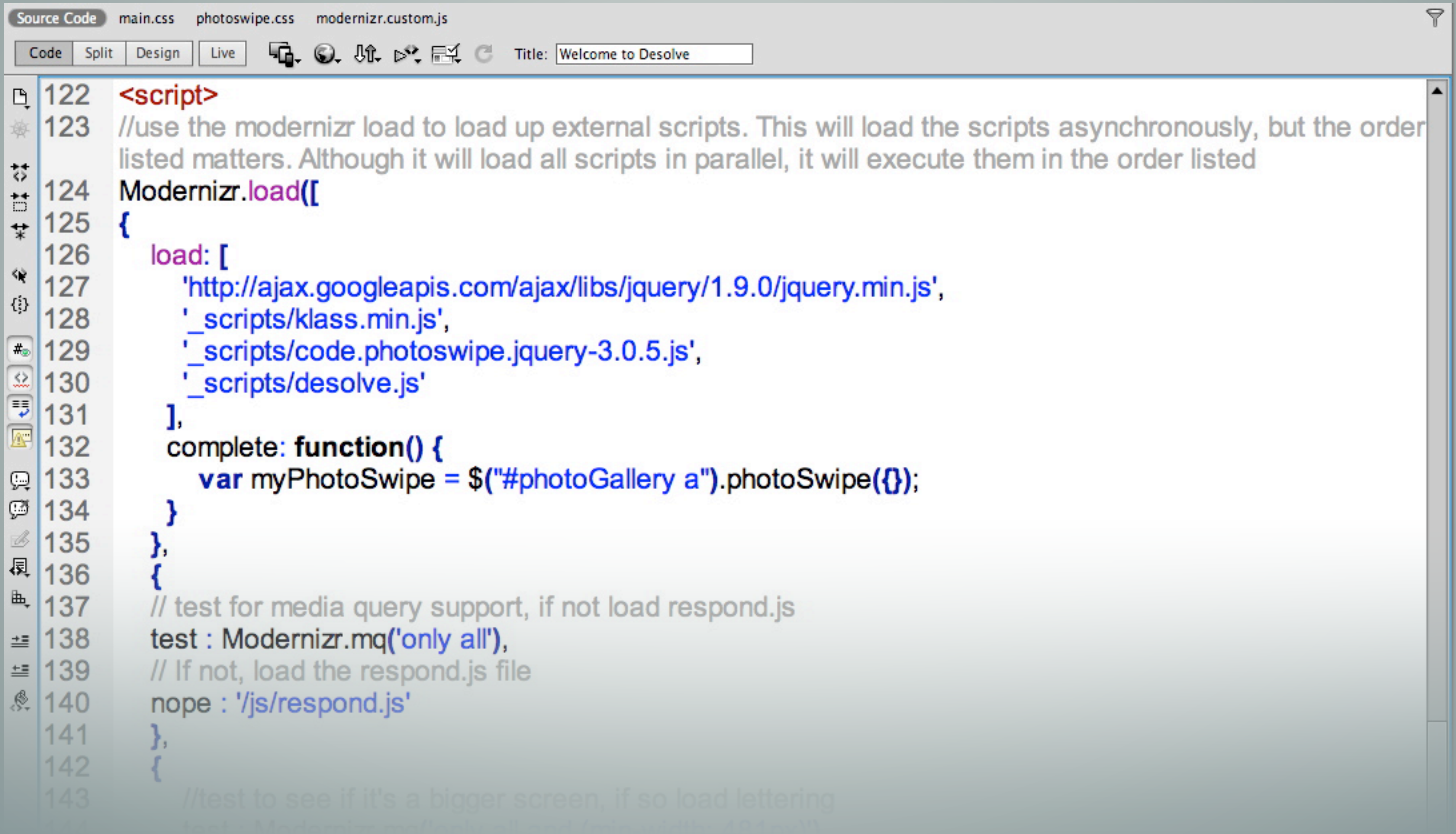
Multiple HTTP requests can hinder the performance of sites on mobile networks far more than large graphics or other assets. Finding ways to limit requests like using CSS Sprites and merging script files together is a critical part of any responsive strategy.

Conditional resource loading

http://www.modernizr.com/

Allows you to check for many CSS3 and HTML5 features, including media queries. Using the built-in **yepnope.js** you can conditionally load resources if a media query is triggered

Demo: Managing resources



Title: Welcome to Desolve

```
<script>
//use the modernizr load to load up external scripts. This will load the scripts asynchronously, but the order listed matters. Although it will load all scripts in parallel, it will execute them in the order listed
Modernizr.load([
{
   load: [
      'http://ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js',
      '_scripts/klass.min.js',
      '_scripts/code.photoswipe.jquery-3.0.5.js',
      '_scripts/desolve.js'
    ],
    complete: function() {
       var myPhotoSwipe = $("#photoGallery a").photoSwipe({});
    }
   },
   {
   // test for media query support, if not load respond.js
   test : Modernizr.mq('only all'),
   // If not, load the respond.js file
   nope : '/js/respond.js'
   },
   {
      //test to see if it's a bigger screen, if so load lettering
      test : Modernizr.mq('only all and (min-width: 481px)')
```

Embrace the user

Don’t expect your site to look or act the same across all screens and devices. Embrace the differences in capabilities and design towards effective user experiences across platforms.

THANK YOU

james williamson | lynda.com

jwilliamson@lynda.com

@jameswillweb on the Twitter

www.simpleprimate.com

download these slides:
slideshare.net/jameswillweb