

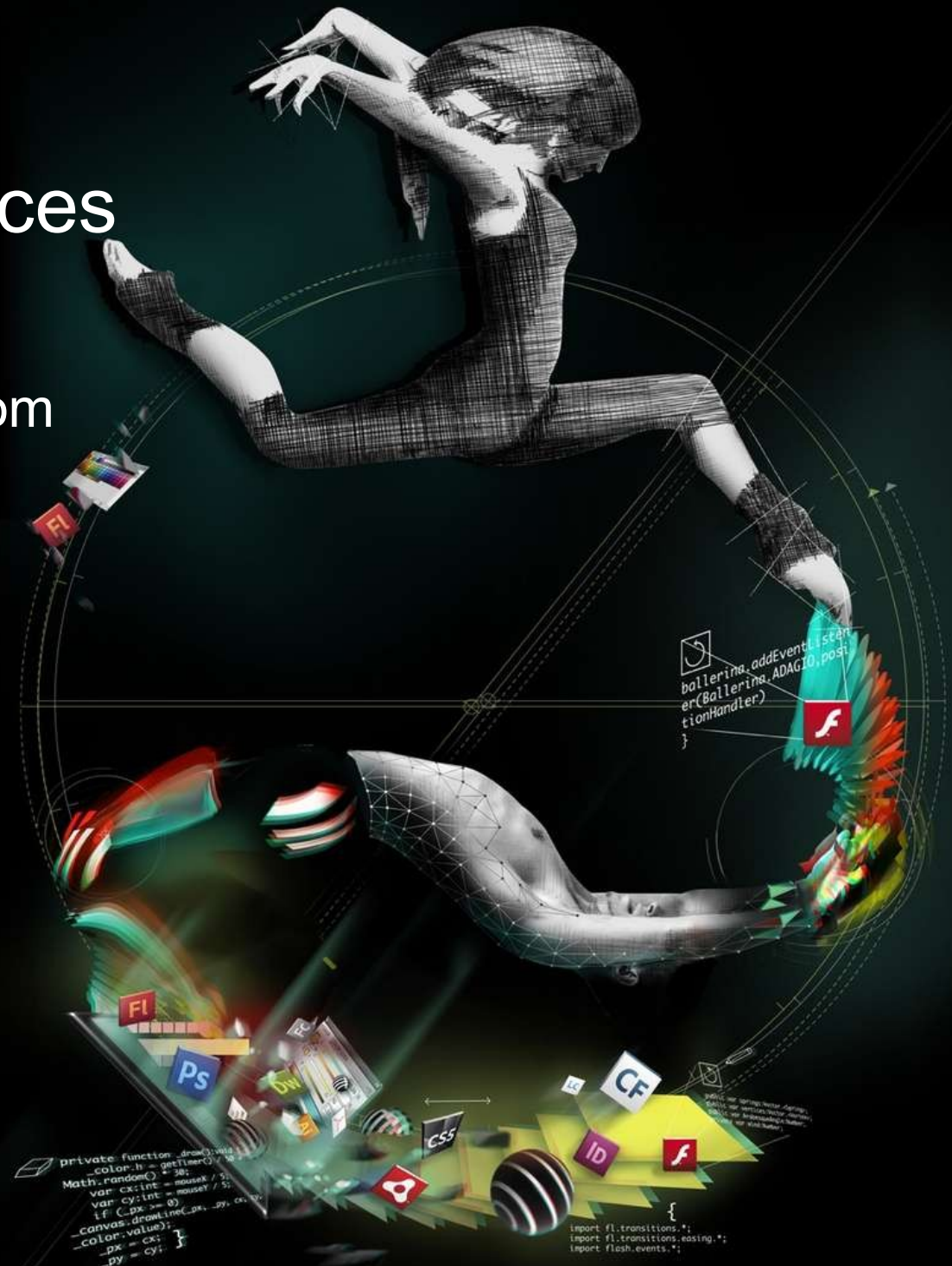


MAX

Optimizing sites for mobile devices with Dreamweaver CS5.5

James Williamson | Senior Author, lynda.com

@jameswillweb twitter



Hello there...

I want to talk about optimizing sites
for mobile devices

Why?

Because we now live in the mobile age

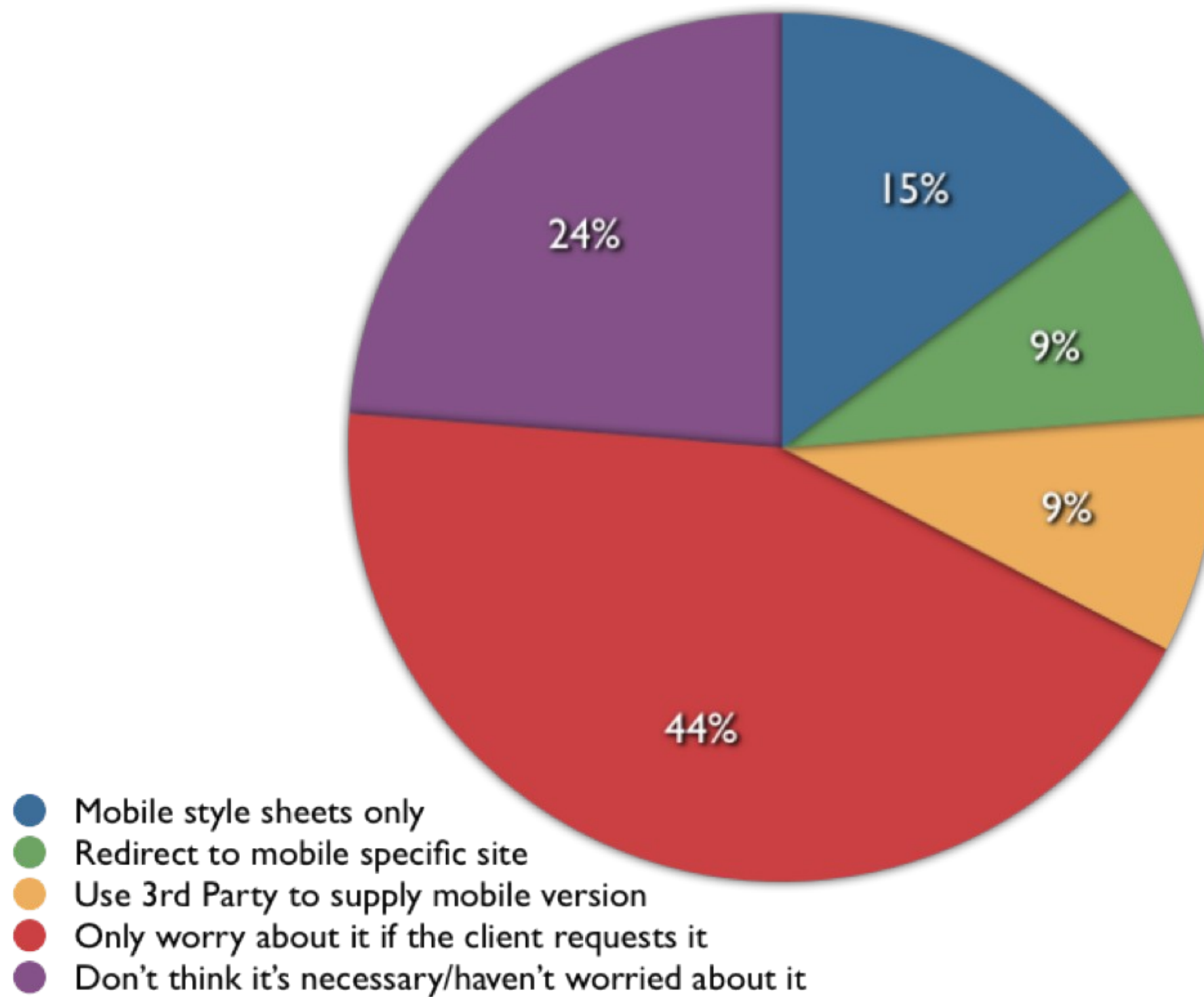
In 2010: 940 million global mobile broadband subscribers
226 million in the Americas
62 million in the US

-mobithinking.com

Quick question:

- When you are designing or planning a new site, what do you do about delivering content for mobile?
 - A) Design a separate mobile site and redirect users based on device type
 - B) Build mobile stylesheets to control layout on mobile devices
 - C) Only worry about it if the client requests it
 - D) What do you mean by mobile?

Recent sampling of web designers



Failing to develop a mobile strategy means you're
ignoring a large portion of your users

- **Develop a native app**
 - Good for targeted, task-based or platform-specific situations
 - Time and budget intensive, unique skill-set
- **Develop a separate mobile site**
 - Gives you the freedom to focus on designing for the mobile experience
 - Can use mobile-specific frameworks
 - Time and budget intensive
 - Often utilizes device sniffing & redirects
- **Serve mobile-specific CSS through media queries**
 - Good way for making sure content remains easy to access
 - Often the only choice due to time, budget, and client constraints
 - Not as robust as other choices

Often, it's not really a choice you get to make

L e t ' s b e h o n e s t

In many cases you're not going to be able to design a separate mobile site, or develop a mobile app.

W h a t t h e n ?

Developing an “mobile inclusive” strategy

- Somewhat based on Luke W’s “mobile-first” strategy
 - <http://www.lukew.com/ff/entry.asp?933>

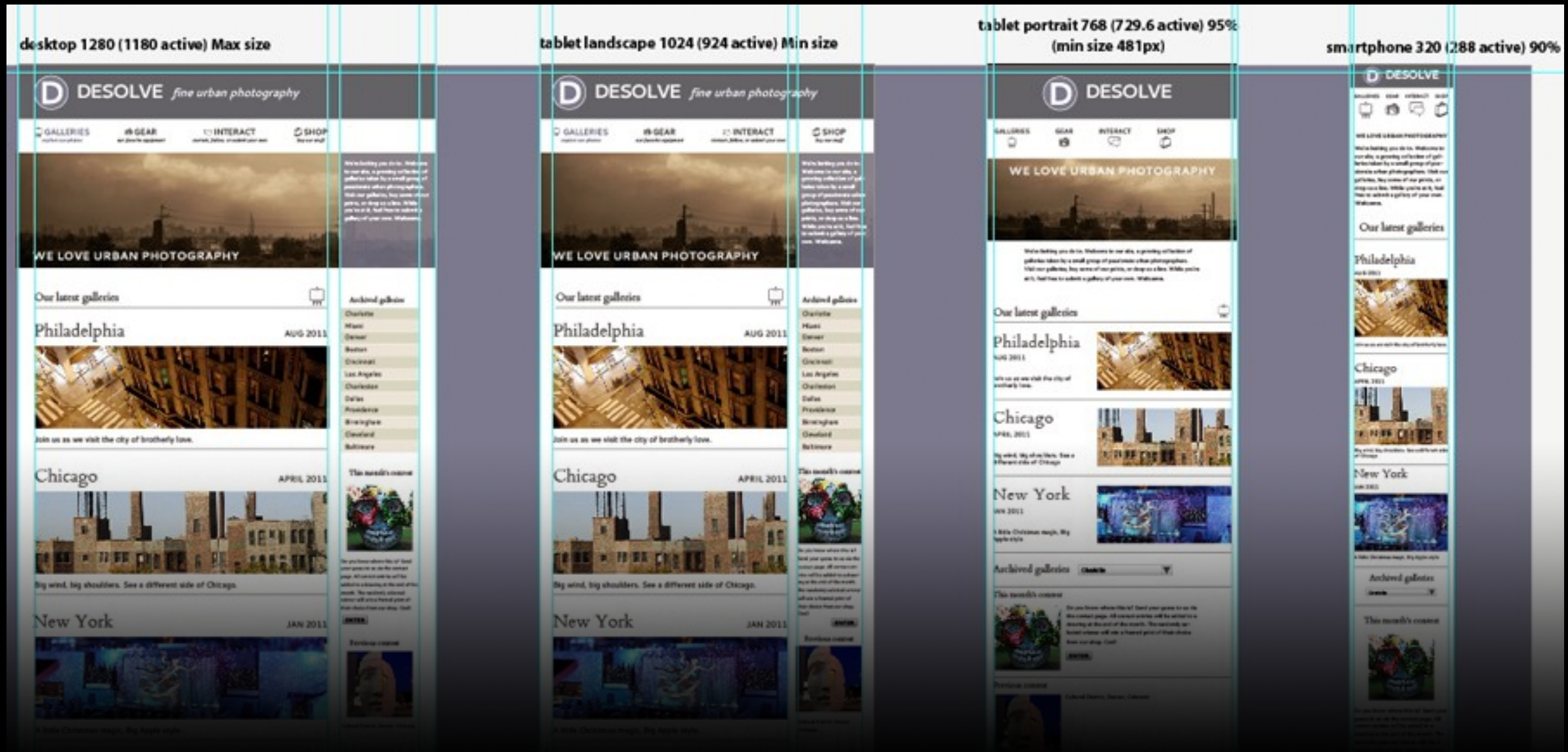
Essentially, it means treating the mobile experience as an equal during the planning and architecting of your sites.

- This requires you to go beyond merely thinking about layout and styles into thinking about user experience, and device context

- Inclusive mobile techniques

- Make tablet & mobile layouts part of the wireframing and prototyping process
- Allow mobile considerations to drive decisions on content and structure
- Embrace mobile-specific capabilities, don't run from them
- Think of device context early and often when planning
- Use media queries to serve styles based on screen size
- Use fluid layouts to deal with screen size inconsistencies
- Make images as responsive as possible, use Sprites & Data URI when possible
- Serve decorative images as backgrounds to prevent unnecessary downloads
- Conditionally load resources through Modernizr

Demo: Planning for mobile devices



```
link rel="stylesheet" href="desktop.css"  
media="[not | only ] screen [and] (expression)"
```

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

Logical operator “**and**” constrains a query based on the expression that follows it. There is no “**or**” operator, normal comma-separated lists serve as “**or**.”

Media Features

width*

height*

device-width*

device-height*

orientation

aspect-ratio*

device-aspect-ratio*

color*

color-index*

monochrome*

resolution*

scan

grid

*accepts “min-” and “max-” prefixes

Exercise: Defining Media Queries

Desolve

Fine Urban Photography

Site navigation

- [Galleries explore our photos](#)
- [Gear our favorite equipment](#)
- [Interact contact, follow, or submit your own](#)
- [Shop buy our stuff!](#)

We love urban photography

We're betting you do to. Welcome to our site, a growing collection of ga
prints, or drop us a line. While you're at it, feel free to submit a gallery of

Our latest galleries

[Philadelphia](#)

Aug 2011

<body.no-js>

PROPERTIES SEARCH REFERENCE W3C VALIDATION BROWSER COMPATIBILITY LINK

The changes you make in this dialog will affect all other pages that include the site-wide media queries file. You can select which file to use for this in the Advanced Settings section of the Site Setup dialog.

☒ This document

☒ Force devices to report actual width

Inserting a special meta tag in your document will force certain devices to report their actual width instead of reporting a false width and then scaling the page.

Description	Media Query	CSS File
mobile	only screen and (max-width:480px)	_css/mobile.css
tablet	only screen and (min-width:481px) and (max-width:768px)	_css/tablet.css
desktop	only screen and (min-width:769px)	_css/desktop.css

+ - Default Presets

Properties

Description:

Min Width: px Max Width: px

CSS File:

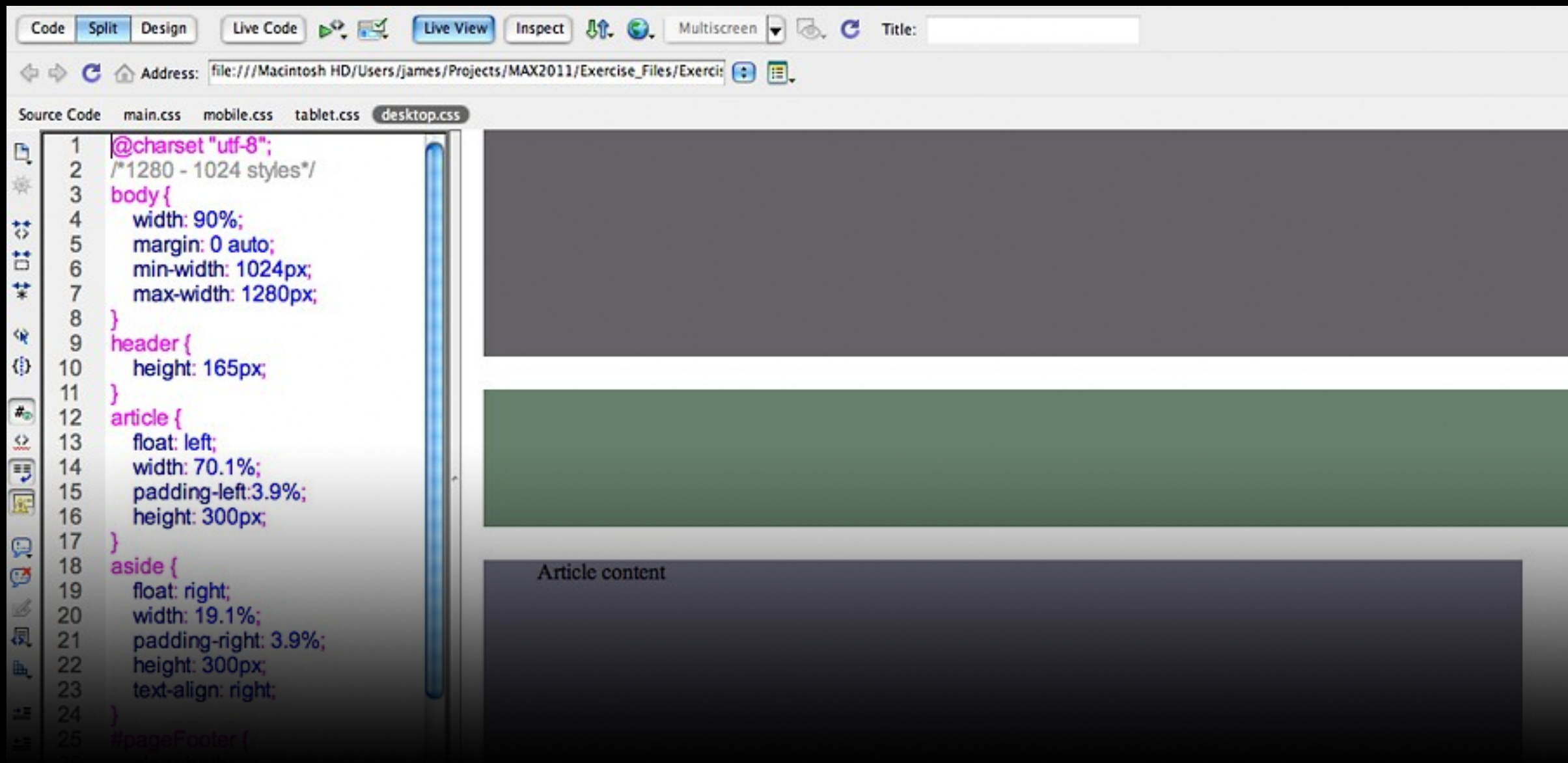
Help Cancel OK

Local F

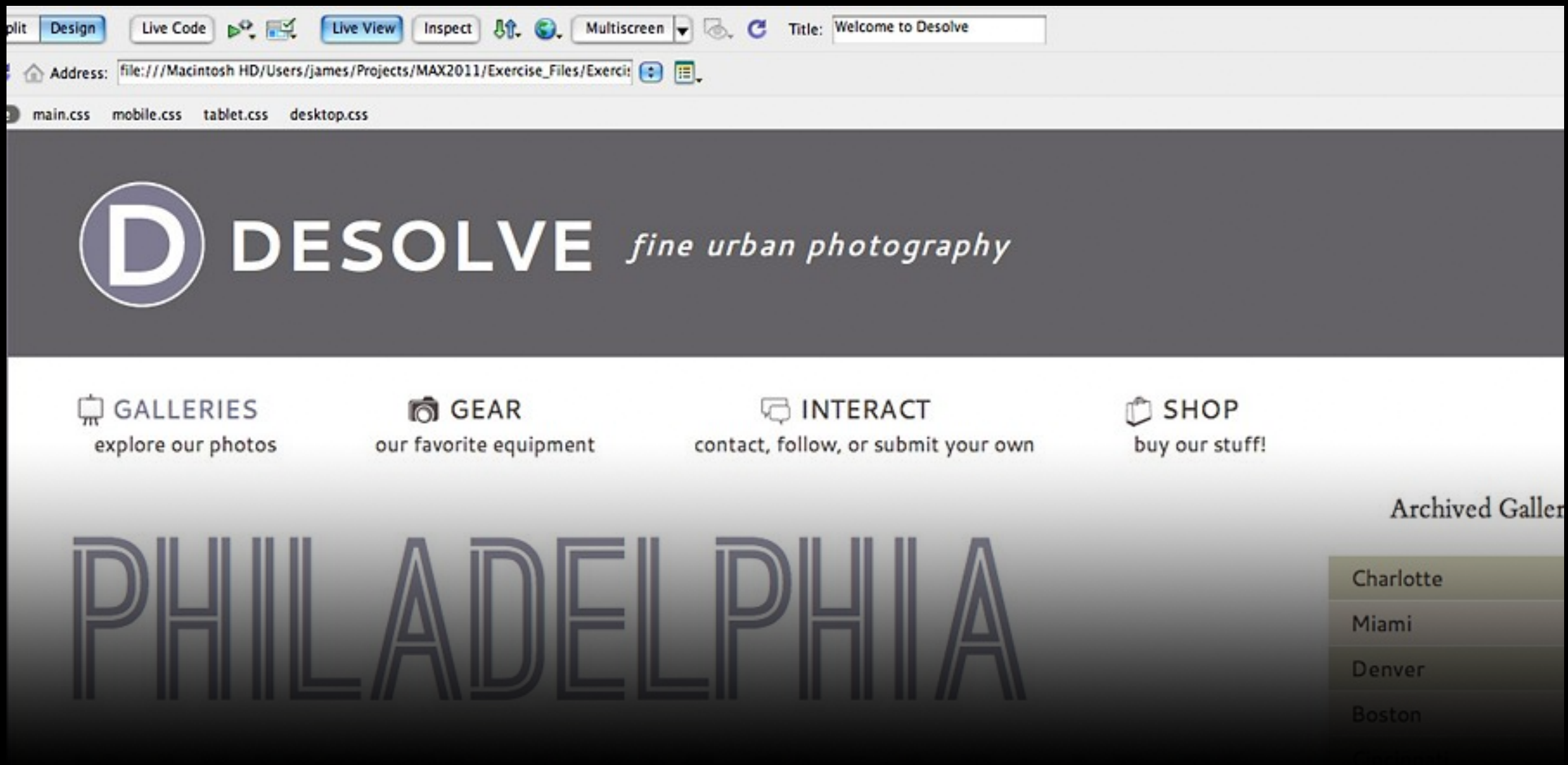
1 UTF-8

- Mobile devices come in many different screen sizes
- Fluid layouts adjust to screen sizes
- You can use 'min' and 'max' widths to make layouts fluid within a specific range
- Take advantage of new CSS capabilities such as multiple columns to make fluid layouts even more responsive

Exercise: Creating Fluid Layouts



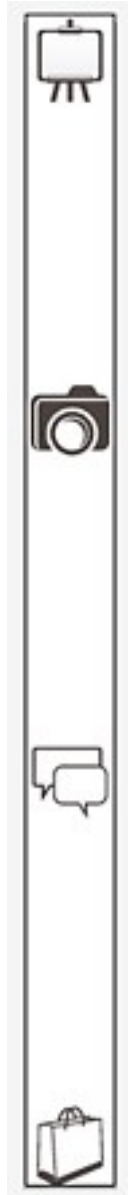
Exercise: Creating Fluid Regions



- Images can be resource intensive
- Scaling images can hurt mobile browser performance
- As much as possible, limit the amount of requests for images your pages make

- For decorative images, use background-images rather than img tags
- Placing background-image requests in media queries means the images are only requested when needed
 - Android testing suggests Android devices still request all background-images
- Combining background-images and media queries also makes it easy to swap out a large desktop image for a smaller mobile version

- Single image that contains multiple icons or UI elements
- Served as background-images
- Icon display is controlled by background-position and element width and height
- Reduces the amount of requests made to the server
- Avoid using extremely large Sprites
 - Android has rendering issues on any image larger than 520,000px
- If you're designing icons for navigation, keep in mind the average touch is around 44px x 44px



- Allow you to apply data (like images) inline in web pages rather than serving them as an external resource
- By combining them with CSS, you can store image data directly in your styles, reducing requests
- Since generated code is usually a little larger than the actual image resource, it's best suited for smaller images and icons

- CSS3 gives us new capabilities that will often allow us to replace images with pure CSS
- Rounded corners, box shadows, CSS gradients, border-images and other new techniques can often be used in place of icons, buttons, or other decorative effects
- Complex CSS drawing can often require considerable non-semantic markup
- Vendor-specific properties such as `-webkit-appearance: push-button` & `-moz-appearance: button` can be used to style UI elements

Exercise: Optimizing Images for Mobile Devices

Chicago

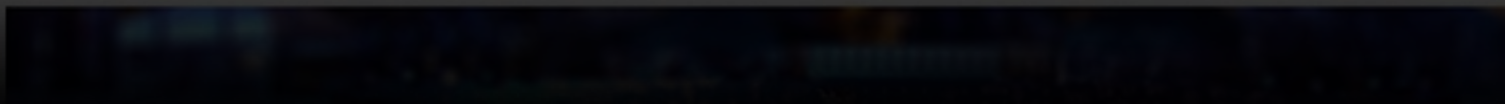
APRIL 2011



Big wind, big shoulders. See a different side of Chicago.

New York

JAN 2011



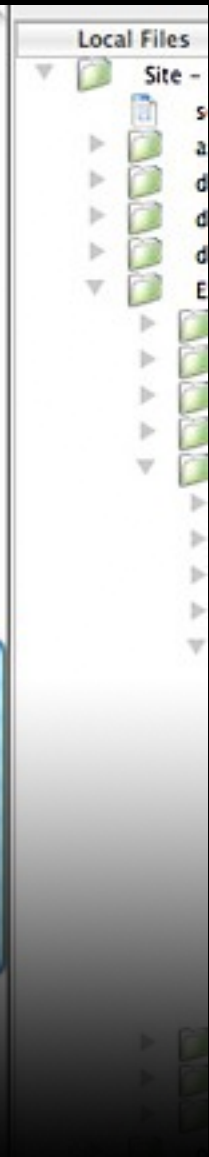
This month's contest



Do you know where this is?
Send your guess to us via the
contact page. All correct
entries will be added to a
drawing at the end of the
month. The randomly selected
winner will win a framed
print of their choice from our
shop. Cool!

Enter

Previous contest



- Certain elements, like longer lists, larger UI elements & widgets don't translate well to the mobile context
- You need to have a strategy in place that determines how to handle such cases
- Often, you can hide and show elements using CSS depending upon context
 - Keep in mind that `display: none` doesn't prevent content from downloading
- More robust solutions often use javascript to generate structure and content based on screen size or device type

- Keep in mind that layout and user interface are not the same thing
- Take care to develop a semantic structure that is flexible enough to be presented in different ways based on context
- Consider using touch events for the mobile devices
- Balance user experience, context, and resource overhead when designing for multiple devices

- Take advantage of mobile support of HTML5 form elements

New Attributes :

placeholder

autofocus

autocorrect

autocomplete

autocapitalize

New Input Types :

search

email

url

tel

number (stepper)

range (slider)

dates

- Mobile devices have capabilities you can take advantage of and develop towards
 - Geolocation
 - Touch & gesture support
 - HTML5 form support
 - Accelerometers
 - Audio and video capture
 - Phone capabilities
 - SMS
- By including mobile device planning as part of the initial process, you'll begin to think about new ways to present and interact with content

- Certain webkit-specific CSS properties can assist with designing for iOS and Android devices
 - -webkit-user-select
 - -webkit-touch-callout
 - -webkit-tap-highlight-color

Exercise: Designing for the Mobile Context

The image shows a web browser window with a code editor on the left and a web page on the right. The code editor displays the following code:

```
14 <link href="_css/tablet.css" rel="stylesheet" type="text/css" media="only screen and (min-
15 width:481px) and (max-width:768px)">
16 <!-- desktop -->
17 <link href="_css/desktop.css" rel="stylesheet" type="text/css" media="only screen and (min-
18 width:769px)">
19 <script src="_scripts/jquery-1.6.3.min.js"></script>
20 <script>
21 $(document).ready(function() {
22 //is this a touch device?
23 var hasTouch = 'ontouchend' in document ? true : false;
24 //if the device is touch enabled, add the hover effect class and strip the hover styling for
25 non-touch devices
26 if(hasTouch) {
27 $('.intro').removeClass('nonTouch');
28 $('.more').bind('touchend', function(e) { $('.intro').addClass('hover_effect'); });
29 }
30 });
31 </script>
32 </head>
33 <body class="no-js">
34 <header id="pageHeader">
35 <hgroup>
36 <h1>Desolve</h1>
37 <h2>Fine Urban Photography</h2>
38 </hgroup>
39 </header>
40 <nav id="mainNav">
41 <h2>Site navigation</h2>
42 <ul>
```

The web page on the right features the 'DESOLVE' logo at the top. Below the logo are four navigation links: 'GALLERIES', 'GEAR', 'INTERACT', and 'SHOP', each accompanied by a small icon. The main heading is 'PHILADELPHIA'. Below this heading is a paragraph of text: 'In August we traveled to Philadelphia for a weeks worth of shooting. Although we didn't find much brotherly love, we did find a lot to love about Ben Franklin's old home, From the amazing 30th Street Station to a turn of the century brothel, we cover it all.' At the bottom of the paragraph is a 'read more' button. The background of the web page is a dark, textured image.

- In many cases, resources that you'll use for desktop clients aren't needed for mobile clients
- You should carefully consider how many external javascript libraries and CSS files you need to use
- Using Modernizr and YepNope, you can add conditional loading of resources based on screen size

<http://www.modernizr.com>

- In a Mobile Inclusive site, media query support is required*

Respond.js from the Filament Group

<https://github.com/scottjehl/Respond>

(gzipped version is 1kb)

css3-mediaqueries-js by Wouter van der Graaf

<http://code.google.com/p/css3-mediaqueries-js/>

(15kb uncompressed)

*Don't use @font-face rules in IE-based media queries

Combine resources when possible

- External javascript and CSS files can be combined to limit the number of server requests
- You can use inline @media rules to combine multiple media queries in a single file

```
/*mobile styles*/@media only screen and  
(max-width:480px) {  
  
}
```

Compress external resources

- Strip extra whitespace, line breaks, and comments from resources
- Use gzip compression to decrease file size

Exercise: Managing Resources

```
Source Code  main.css  modernizr-custom.js  jquery-1.6.3.min.js
11 <script src="_scripts/modernizr-custom.js"></script>
12 <script>
13 //use Modernizr.load to determine which external resources to load and which scripts to run
14 Modernizr.load([ {
15   load: '_scripts/jquery-1.6.3.min.js',
16   callback: function (url, result, key) {
17     //bind touch events to more button
18     $(document).ready(function() {
19       //is this a touch device?
20       var hasTouch = 'ontouchend' in document ? true : false;
21       //if the device is touch enabled, add the hover effect class and strip the hover styling for
       non-touch devices
22       if(hasTouch) {
23         $('.intro').removeClass('nonTouch');
24         $('.more').bind('touchend', function(e) {
25           $('.intro').addClass('hover_effect');
26         });
27       };
28     }); //end hasTouch function
29     //load up images in gallery based on screen size
30     $(document).ready(function(){
31       if(Modernizr.mq('only all and (min-width: 481px)')){
32         //loop through gallery images and set larger path
33         var imageArray = document.getElementsByClassName('photos')[0].
getElementsByName('img');
34         for ( var i = 0; i < imageArray.length; i++ ) {
35           var newPath = '_images/philly_' + (i+1) + '.jpg';
36           imageArray[i].setAttribute('src',newPath);
37           //load image from path
```



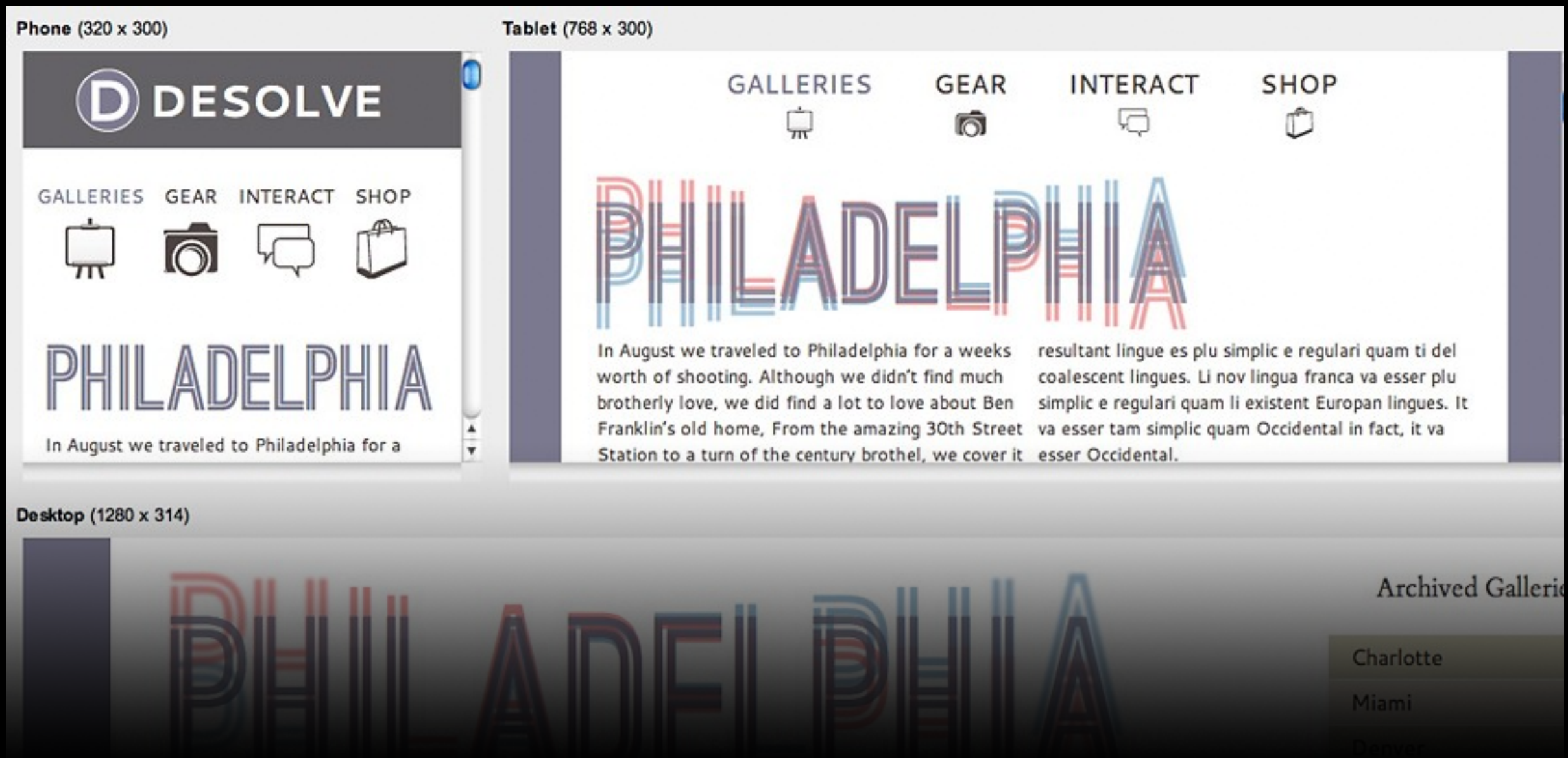
- Media queries often fail by themselves due to mobile browser viewports and reported screen width
- Pages are often initially scaled due to large viewport sizes
- The meta viewport tag allows you to control initial scaling and how the device reports its screen width

```
<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, and set the initial page scale to 100%.

- You can use the `apple-touch-icon` link to add an icon for bookmarked sites
- Using `-webkit-text-size-adjust: none` can prevent text scaling when orientation is changed

Exercise: Enhancing mobile experiences





Thank you!

@jameswillweb | twitter

www.simpleprimate.com | dead blog