

Developing Mobile Websites

Media Queries

Lesson 1, Activity 2: Media Queries

Targeting Widths and Devices with CSS3 Media Queries

Our *Jazz Calendar* site is more and more flexible: we've used a percentage-based layout to scale the site independent of the browser window's size, and we've applied CSS to scale images and other fixed-width elements. But the design still doesn't hold up when the browser window gets really narrow: elements squash together, some text might need to scale or pad differently, etc. And we've not yet addressed the specific needs of users accessing our site on a mobile device - even a scaled-down version probably won't work too well on a phone.



It would be nice not just to scale the site, but to make qualitative changes to the design based on the inherent properties of the device being used to view it, and also to adjust the design at specific browser-window widths. To address this issue, we turn to the **media**

query.

You may already be familiar with CSS media types - a CSS2 specification that offered a way to include different stylesheets based on the media type (screen, print, etc.) being employed. A print-specific stylesheet, for example, might present a page without background images, with a print-friendly font face, and with margins appropriate for printing.

CSS3 extends this concept with the media query - a way to apply CSS rules selectively based on both the type of media and the physical properties of the device (browser, phone) being used to access the page.

Each media query comprises a media type (e.g., "screen") and zero or more logical expressions - a condition evaluating to true or false based on the conditions of particular media features. We can test our user's device for screen width, device width, orientation ("portrait" or "landscape"), and other features.

Media Query Features

Feature	Possible Values	Min/Max?	Explanation
color	int	yes	bits per color component
color-index	int	yes	number of entries in color lookup table
device-aspect-ratio	int/int	yes	aspect ratio
device-height	length (pixels)	yes	height of the output device
device-width	length (pixels)	yes	width of the output device
grid	int	no	true for a grid-based device
height	length (pixels)	yes	height of the rendering surface
monochrome	int	yes	bits per pixel in a monochrome frame buffer

resolution	"dpi" or "dpcm"	yes	resolution
scan	"progressive" or "interlaced"	no	scanning process of "tv" media types
width	length (pixels)	yes	width of the rendering surface

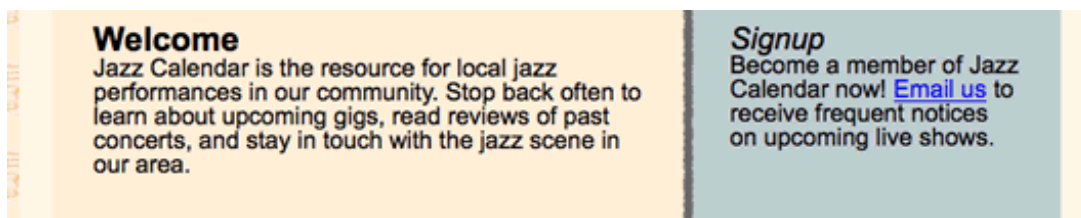
Consider a revised version of the *Jazz Calendar* home page: open <MediaQueries/Demos/noborder.html> and <MediaQueries/Demos/css/noborder.css> in your file editor. Employing the min-width query, we create a page where:

1. The graphic border between the two columns shows only when the browser is at least 768px wide.
2. The left-column gig listings line up two-across when the browser is wider than 768px; one-across when narrower than 768px.

The key here are the media queries. The following code says "include the background image only if the width of the browser is at least 768 pixels."

```
@media screen and (min-width: 768px) {
  #main {
    width:93.75%;
    margin:0 auto 5.8888889% auto;
    background: #c0cbd0 url('../images/bg_main.png') repeat-y 62.888889% 0;
  }
}
```

The screen shot below shows the *Jazz Calendar* site at a browser width of 800 pixels. Note that the graphic background shows:



The next screen shot shows the *Jazz Calendar* site at a browser width of 765 pixels. The graphic background is no longer visible:



Linearizing the Layout

As the browser narrows, it might make sense to slide the right sidebar under the left main column - a common strategy for shrinking a two-column design into the more narrow confines of a smartphone. As the screen shot below shows, the columns get pretty slim when the browser is narrowed too much, to the point where just a few words show on each line of the right sidebar:



Let's add a few more media queries to handle browser widths of less

than 520px - an arbitrary breakpoint, chosen here because it seems to make sense in the context of our page.

The page [MediaQueries/Demos/linearize.html](#) now behaves in the same manner as the previous example (graphic border and two-across gig listings disappear at browser-window widths less than 768px), but now the sidebar stops floating right (and thus slides under the main content) at widths less than 520px. Open the page in a browser and change the browser's width to check it out. Be sure to examine the media queries in [MediaQueries/Demos/css/linearize.css](#) - specifically the media queries that govern the #maincontent and #sidebar elements at widths less than 520 pixels:

```
@media screen and (max-width: 520px) {  
  #main #maincontent {  
    float:none;  
    width:92%;  
  }  
}  
  
@media screen and (max-width: 520px) {  
  #main #sidebar {  
    width:92%;  
    float:none;  
  }  
}
```

Using max-device-width

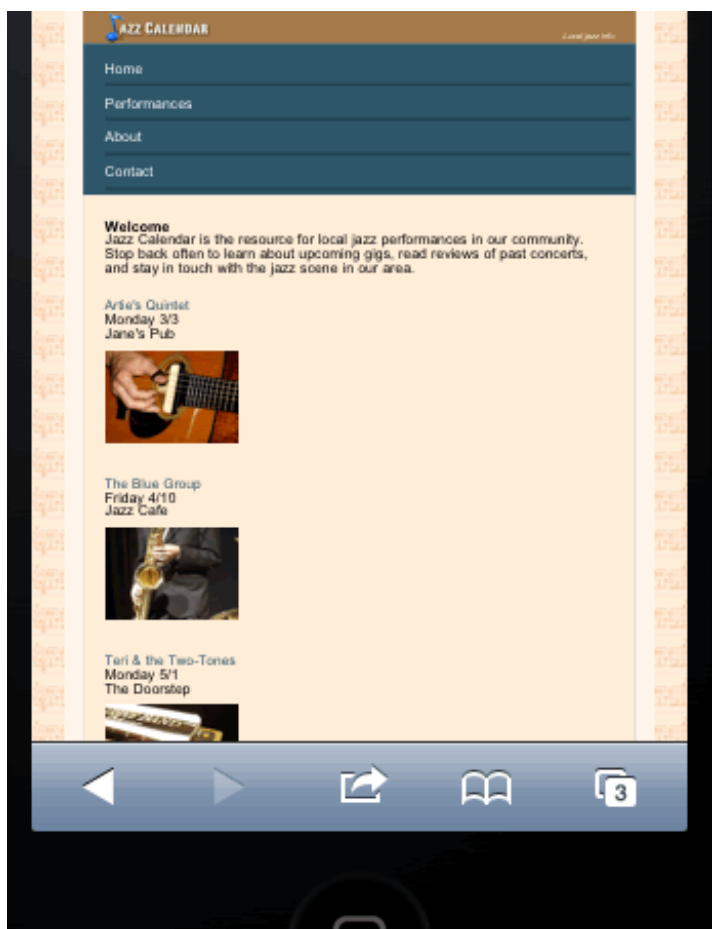
We'll now use media queries to target the properties of the device being used to view our page. The property max-device-width allows us to target the physical properties of the viewport -

max-device-width: 480px, for instance, applies not when the browser window is resized to less than 480px, but rather when the

device itself - a smartphone, say - has a screen less than 480px wide.

To view your code on a smartphone or emulator, you will need to publish to a Web server; please refer to the setup instructions for more information on how to do this.

Consider the file [MediaQueries/Demos/devicewidth.html](#). View this latest version of the *Jazz Calendar* home page on an iPhone, Android, or other smartphone device or on an emulator - you'll see that we've used media queries to selectively apply style rules just for devices with a maximum screen width less than 480px:



1. The top nav is listed vertically, with some extra top/bottom padding and bottom borders.
2. As on the small-width browser design, we don't float the sidebar

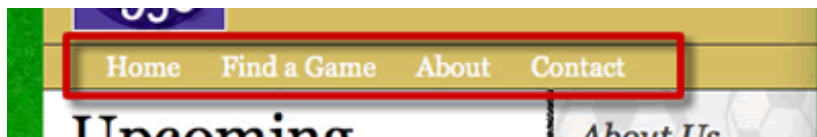
right - rather, it slides under the main content area.

3. Similarly, the gig listings appear one-on-each-line, rather than two-across.

Lesson 1, Activity 4: Adding Media Queries for Pickup Soccer

Duration: 20 to 30 minutes.

Let's use media queries to fix a few things on the Pickup Soccer site. The top (white) nav has never worked well once the browser window is resized too small (it slides under the logo when the container runs out of room to float it right). Let's change the top nav as such:



Also, the game listings get a bit difficult to read when the browser gets too narrow. Let's change this to move the titles under the images when the browser gets narrow:

Upcoming Games



Barry Park

Sundays, around 2pm; near the back of the largest field



Lakeside Park

Sundays, around 2pm; near the back of the largest field



Schiller Elementary



Central High

1. Open [MediaQueries/Exercises/topnavandtitles/css/style.css](#) in your file editor.
2. Update CSS to unfloat the top nav at widths below 600px.

- Update CSS so that, at browser-window widths below 960px, the titles for the game listing fall below (instead of to the right of) the thumbnail photos.

Solution:

MediaQueries/Solutions/topnavandtitles/css/style.css

---- C O D E O M I T T E D ----

```
nav {
  padding:0.96% 3.75% 0.96% 3.75%;
  float:right;
}

@media screen and (max-width: 600px) {
  nav {
    float:none;
    border-top:1px solid #333;
    border-bottom:1px solid #333;
  }
}
```

---- C O D E O M I T T E D ----

```
#maincontent .game {
  float:left;
  width:47.892074199%;
  line-height:14px;
  font-size:14px;
  margin:0.8431703204%;
}

@media screen and (max-width: 960px) {
  #maincontent .game {
    padding-bottom:20px;
  }
}

#maincontent .game img {
  float:left;
  margin:0 2.86% 0 0;
  max-width:100%;
}
```

```
@media screen and (max-width: 960px) {  
  #maincontent .game img {  
    float:none;  
  }  
}  
  
@media screen and (max-width: 960px) {  
  #maincontent .game h3 {  
    font-size:20px;  
    margin:3px 0;  
    line-height:22px;  
  }  
}  
  
#maincontent iframe {  
  max-width:100%;  
}  
  
---- C O D E   O M I T T E D ----
```

We use media queries to effect these changes. A query targeting a `max-width` of 600 pixels unfloats the top nav and adds borders. Another set of queries, targeting a browser width of 960 pixels, unfloats the `.game` images and modifies the `h3` titles' size.

Lesson 1, Activity 5: Linearizing the Pickup Soccer Site

Duration: 20 to 30 minutes.

In this exercise, you'll update the Pickup Soccer site again, this time presenting the sidebar under the main column at widths under 520px, as shown:



1. Open [MediaQueries/Exercises/sidebarnotfloated/index.html](#) and [MediaQueries/Exercises/sidebarnotfloated/css/style.css](#).
2. Update CSS to show the main nav without floating right and with modified background and text colors for browser widths under 520 pixels.
3. Update CSS to unfloat the sidebar, sliding it under the main column at widths under 520px.

Solution:

MediaQueries/Solutions/sidebarnotfloated/css/style.css

---- C O D E O M I T T E D ----

```
#container {
  width:85%;
  margin:0 auto;
  background: #fff url('../images/bg_soccerballs.png') repeat-y 65.520833% 0;
}
```

```
@media screen and (max-width: 520px) {
  #container {
    background: #fff;
  }
}
```

---- C O D E O M I T T E D ----

```
nav {
  padding:0.96% 3.75% 0.96% 3.75%;
  float:right;
}
```

```
@media screen and (max-width: 520px) {
  nav {
    float:none;
    background:#ccc;
  }
}
```

```
nav ul {
  display:inline;
}
```

```
nav ul li {
  display:inline;
}
```

```
nav ul li a {
  color:#fff;
  text-decoration:none;
  margin-left:20px;
}
```

```
@media screen and (max-width: 520px) {
  nav ul li a {
    color:#000;
  }
}
```

```

}
}

#maincontent {
width:61.770833333333%;
padding:0.96% 0 3.75% 3.75%;
float:left;
}

@media screen and (max-width: 520px) {
#maincontent {
width:96.25%;
padding:0.96% 0 3.75% 3.75%;
float:none;
}
}

```

---- C O D E O M I T T E D ----

```

#sidebar {
width:26.979166666667%;
float:right;
padding:3.75%;
color:#333;
line-height:20px;
}

@media screen and (max-width: 520px) {
#sidebar {
width:96.25%;
padding:3.75% 0 3.75% 3.75%;
float:none;
clear:left;
background:#ccc;
}
}

```

---- C O D E O M I T T E D ----

We use a media query to unfloat the main nav at widths under 520 pixels:

```

@media screen and (max-width: 520px) {

```

```
nav {  
  float:none;  
  background:#ccc;  
}  
}
```

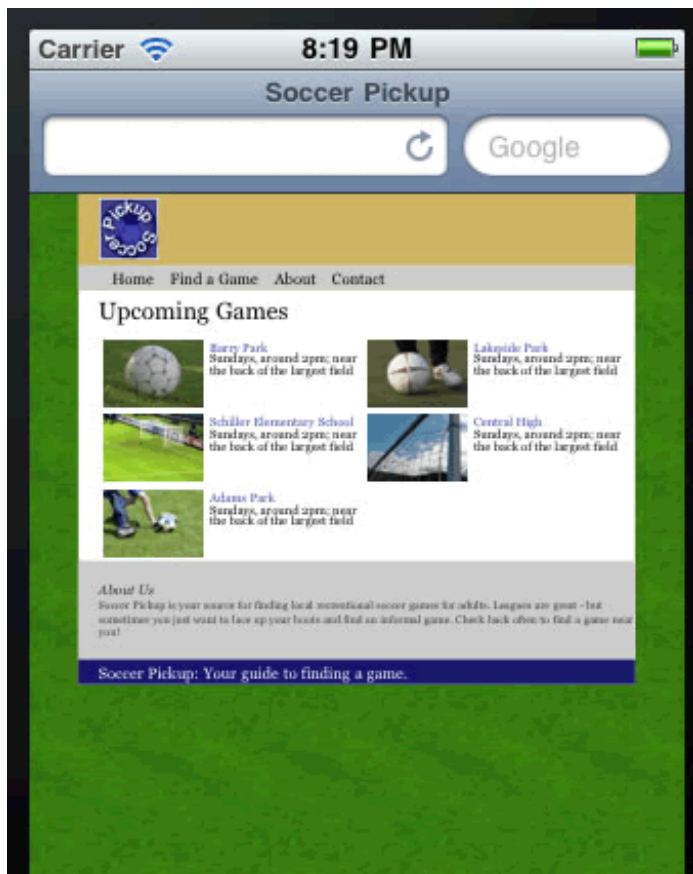
Similarly, we use several more media queries to prevent the floating left and right of the main content and sidebar elements, respectively, forcing the sidebar to fall under the main content when the browser resizes to less than 520 pixels.

```
@media screen and (max-width: 520px) {  
  nav {  
    float:none;  
    background:#ccc;  
  }  
}  
  
@media screen and (max-width: 520px) {  
  #maincontent {  
    width:96.25%;  
    padding:0.96% 0 3.75% 3.75%;  
    float:none;  
  }  
}
```

Lesson 1, Activity 7: Pickup Soccer - For Phones

Duration: 20 to 30 minutes.

In this exercise, you will update the *Pickup Soccer* site for mobile devices - we'll use media queries to display the site like this on a phone:



1. Open [MediaQueries/Exercises/devicewidth/index.html](#) and [MediaQueries/Exercises/devicewidth/css/style.css](#) in your file editor.
2. Write CSS to display the main nav below the header (rather than floated within).
3. Write CSS to display the sidebar below the main column (rather than floated to the right of).

Solution:

[MediaQueries/Solutions/devicewidth/css/style.css](#)

----- C O D E O M I T T E D -----

```
#container {
  width:85%;
  margin:0 auto;
  background: #fff url('../images/bg_soccerballs.png') repeat-y 65.520833% 0;
}

@media screen and (max-device-width: 480px) {
  #container {
    background: #fff;
  }
}
```

----- C O D E O M I T T E D -----

```
nav {
  padding:0.96% 3.75% 0.96% 3.75%;
  float:right;
}

@media screen and (max-device-width: 480px) {
  nav {
    float:none;
    background:#ccc;
  }
}

nav ul {
  display:inline;
}

nav ul li {
  display:inline;
}

nav ul li a {
  color:#fff;
  text-decoration:none;
  margin-left:20px;
}

@media screen and (max-device-width: 480px) {
  nav ul li a {
    color:#000;
  }
}
```

```

}

#maincontent {
  width:61.7708333333%;
  padding:0.96% 0 3.75% 3.75%;
  float:left;
}

@media screen and (max-device-width: 480px) {
  #maincontent {
    width:96.25%;
    padding:0.96% 0 3.75% 3.75%;
    float:none;
  }
}

```

---- C O D E O M I T T E D ----

```

#sidebar {
  width:26.9791666667%;
  float:right;
  padding:3.75%;
  color:#333;
  line-height:20px;
}

@media screen and (max-device-width: 480px) {
  #sidebar {
    width:96.25%;
    padding:3.75% 0 3.75% 3.75%;
    float:none;
    clear:left;
    background:#ccc;
  }
}

```

---- C O D E O M I T T E D ----

We construct queries with `max-device-width: 480px` to target narrow devices, unfloating the nav:

```

@media screen and (max-device-width: 480px) {
  nav {

```

```
float:none;  
background:#ccc;  
}  
}
```

#maincontent:

```
@media screen and (max-device-width: 480px) {  
  #maincontent {  
    width:96.25%;  
    padding:0.96% 0 3.75% 3.75%;  
    float:none;  
  }  
}
```

and #sidebar elements:

```
@media screen and (max-device-width: 480px) {  
  #sidebar {  
    width:96.25%;  
    padding:3.75% 0 3.75% 3.75%;  
    float:none;  
    clear:left;  
    background:#ccc;  
  }  
}
```