

Developing Mobile Websites

Flexible Images and Other Media

Lesson 1, Activity 2: Flexible Images and Other Media

Improving the *Jazz Calendar* Site

The updates to the *Jazz Calendar* site we made in the previous lesson took a static, fixed-width, pixel-based design and turned it into a more flexible, percentage-based page. But a quick examination of the page - especially when one resizes the browser smaller and smaller - shows that the design breaks down when the width of the browser becomes too small. The images, specifically, break the design: the guitar, sax, and other photos on the home page (associated with each gig listing) remain a fixed width and height and, as such, don't receive the same "resize to match the browser" behavior. This means that, at smaller widths, the images overlap the design:

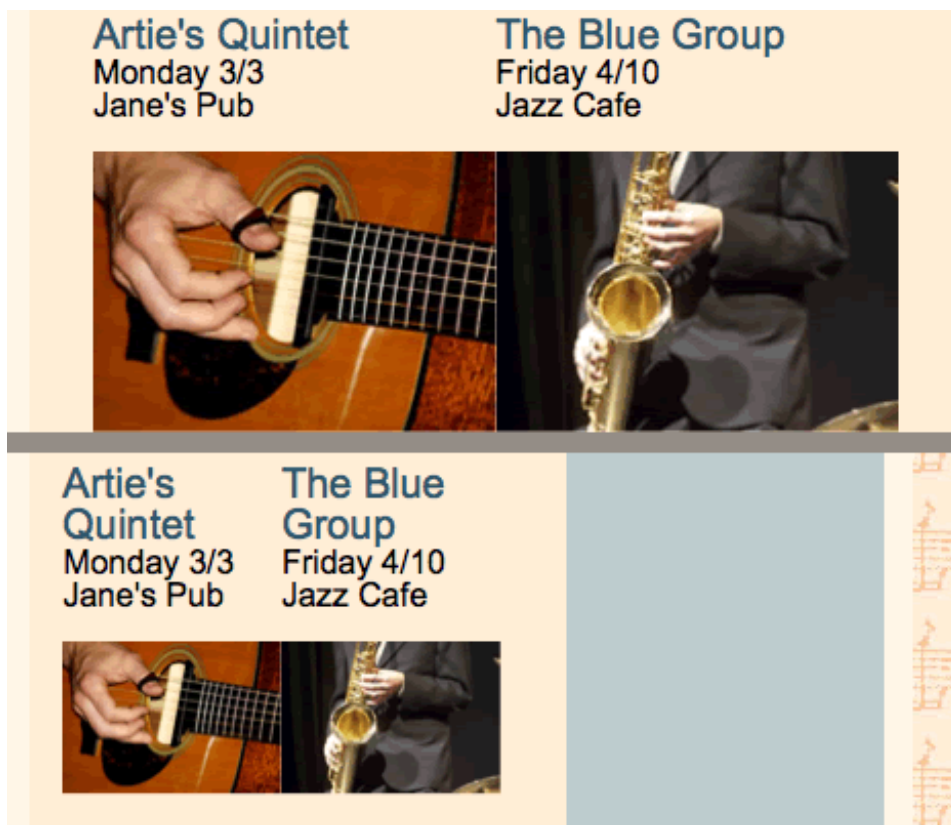


The solution to this is the magical CSS `max-width:100%`. First discovered by [Richard Rutter](#), this CSS rule constrains images (and other elements, as we'll see) to a maximum width inherited by their containing element - that is 100% (and no more) than the element's parent.

Thus, a simple update to the CSS (see [FlexibleImages/Demos/css/flexibleimages.css](#)) for the *Jazz Calendar* page - specifically, for the images associated with each gig listing - offers the user a more graceful presentation when the browser window is less wide.

```
#main #maincontent .performance img {
  margin:0 0.9% 10px 0;
  max-width:100%;
}
```

View [FlexibleImages/Demos/flexibleimages.html](#) in your browser to see the live page. Note that, as you size your browser narrowed, the gig images now get smaller to fit their container element:



Consider an interior page within the *Jazz Calendar* site - one with a

large top photo. Without addressing the width of this photo, the design would break even more readily than would our home page - one need not resize the browser too small before the fixed width of the image pushes it into the sidebar:



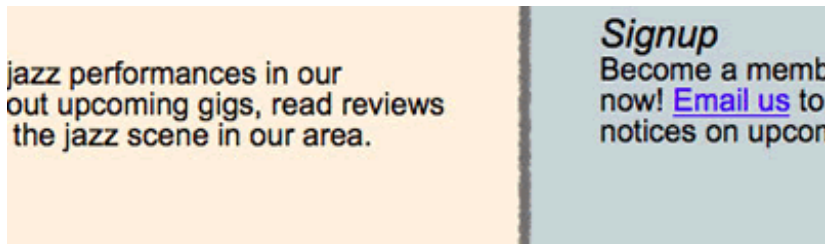
The addition of one CSS rule fixes this problem: our large top image shrinks appropriately as the browse window resizes. Check out [FlexibleImages/Demos/interiorjazzpage.html](#) and [FlexibleImages/Demos/css/interiorjazzpage.css](#)

```
#main #maincontent img.pagephoto {
  max-width:100%;
  margin-bottom:4%;
}
```

Note that the `max-width:100%` CSS rule applies for other fixed-width elements, such as video, besides images. We'll practice this in an upcoming exercise.

Flexible Background Images

Imagine that we wish to update the design of the *Jazz Calendar* site to add a graphic border between the left and right columns.



An easy way to accomplish this is to use a background image - with the cream color on the left, the dusky blue on the right, and the stylized line running vertically at the border - and tile it vertically behind the main content div. But how to do so, given that our design is flexible - with neither column a fixed width?

We start with a background image that is large enough to handle the most wide browser window; we'll use a 3000-pixel-wide image. (This is the file [FlexibleImages/Demos/images/bg_main.png](#).) To split the columns in the background graphic to fit the percentage widths of the two columns whose border we are creating, we would recall the original column pixel width as a percentage of the width of the container element (the #main div): $566\text{px} / 900\text{px} = 62.888889\%$. Thus the border in our graphic goes at $3000\text{px} * 62.888889\% = 1887\text{px}$. We can then use CSS's background-position rule to place the background image appropriately:

```
#main {
  width:93.75%;
  margin:0 auto 5.8888889% auto;
  background: #fff url('../images/bg_main.png') repeat-y 62.888889% 0;
}
```

Open the file [FlexibleImages/Demos/flexiblebg.html](#) in your browser to see the result.

Of course, we're not actually scaling the background image - just placing it appropriately for the relative widths of the columns.

Lesson 1, Activity 4: Fixing Images in Pickup Soccer

Duration: 10 to 15 minutes.

In this exercise, you will edit the Pickup Soccer site to learn how to scale image widths dynamically.

1. Open [FlexibleImages/Exercises/flexiblesoccerimages/index.html](#) and [FlexibleImages/Exercises/flexiblesoccerimages/css/style.css](#).
2. Edit the CSS file to make the soccer-game images' widths scale dynamically.

Solution:

[FlexibleImages/Solutions/flexiblesoccerimages/css/style.css](#)

```
body {  
    font-family:Georgia,serif;  
    background-image:url('../images/bg_grass.jpg');  
}
```

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

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

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

We've used `max-width:100%` to scale the game images dynamically, applying the style rule to any image inside of an element of class `game`.

Lesson 1, Activity 6: Flexible Widths for Embedded Video

Duration: 10 to 15 minutes.

We consider now an interior page for the Pickup Soccer site - the "About Us" page. mocked up with some greeking text. We include a video - an embedded YouTube video of cool soccer tricks, appropriate for our audience. Of course, in the absence of any extra CSS, the embedded video would stay a fixed width, regardless of the size of the browser window, and would thus break the design:



1. Open [FlexibleImages/Exercises/flexiblevideo/css/style.css](#) in your file editor;
2. Add an appropriate CSS rule to make the embedded video scale in width as its container element changes;
3. View [FlexibleImages/Exercises/flexiblevideo/index.html](#) to test your work.

Solution:

FlexibleImages/Solutions/flexiblevideo/css/style.css

```
body {  
  font-family:Georgia,serif;  
  background-image:url('../images/bg_grass.jpg');  
}
```

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

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

```
#maincontent p {  
  margin:0 0 15px 0;  
}
```

```
#maincontent iframe {  
  max-width:100%;  
}
```

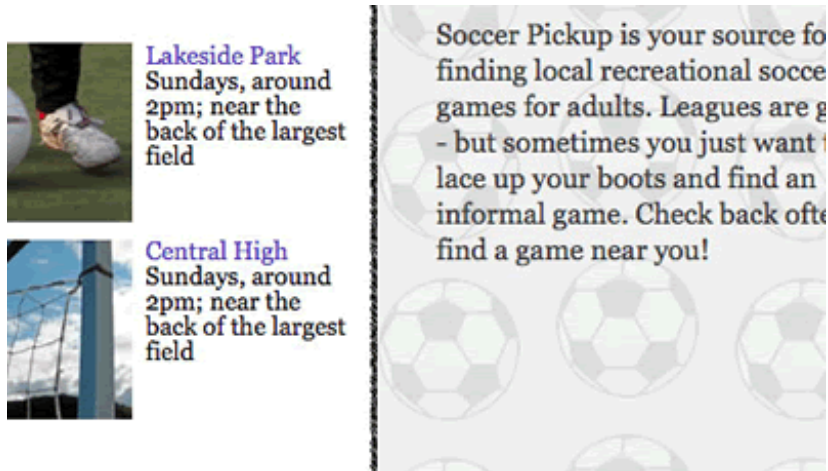
---- C O D E O M I T T E D ----

We use `max-width:100%` to scale the `iframe` used to embed the video.

Lesson 1, Activity 7: Flexible Background Image in the Pickup Soccer Site

Duration: 10 to 15 minutes.

In this exercise, you will add a graphic background for the right column of the Pickup Soccer site to match this design:



1. Open the files [FlexibleImages/Exercises/flexiblebg/index.html](#) and [FlexibleImages/Exercises/flexiblebg/css/style.css](#) in your file editor.
2. If you have access to and are comfortable with using Photoshop or some other image editor, create an appropriate background image for the right column; if not, use this file: [FlexibleImages/Exercises/flexiblebg/images/bg_soccerballs.png](#) - be sure to create the file with a large width.
3. Edit CSS to place the background image so that it appears correctly regardless of the scale of the page.

Solution:

[FlexibleImages/Solutions/flexiblebg/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;  
}
```

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

We use a `background-position` value of `65.520833%` to position the background image to align properly as the columns' widths change.