

# Lecture 5: Introduction to Images

Dr. Jason Schanker

# Introducing Images

- Images are placed on the page using the `img` void element with a `src` attribute whose value is the path to the image.
  - ❑ Example (from `elixir.html`): `<img src = "../images/green.jpg">`  
(Note no closing tag or content)
- Images are requested from the server when processing the HTML: When the browser encounters the `img` element, it looks at the `src` attribute so it can send another request to the server for the image it needs. Thus images can load *after* some of the page is displayed.
  - ❑ This was more noticeable back in the days of dialup modems where you might load a lower resolution image via an additional `img` attribute `lowsrc` \* before the actual one you wanted to show so the user wouldn't have to wait to see something!
- The main image formats to choose from are GIF (.gif), JPEG (.jpg), and PNG (.png).

**\* Don't use `lowsrc` now, it's considered an obsolete or deprecated attribute.**

# Image formats: Which one to choose?

- **GIF (Graphics Interchange Format)**: The original image format for the web; supports animations and one-color transparency; limited to 256 colors; good for logos, text, and simple drawings.
- **JPEG**: Good for photos; uses lossy (as opposed to loseless) compression meaning that some parts of the original image are lost to reduce the space requirement of the image.
- **PNG (Portable Network Graphics)**: Newer image format that's good for logos, text and simple drawings; supports images of varying complexity: PNG-8 (256 colors), PNG-24 (16,777,216 colors), and PNG-32 (4,294,967,296 colors); Supports multiple transparent colors for anti-aliasing (allowing smoother edges)

# Book's Trivia Question exercise

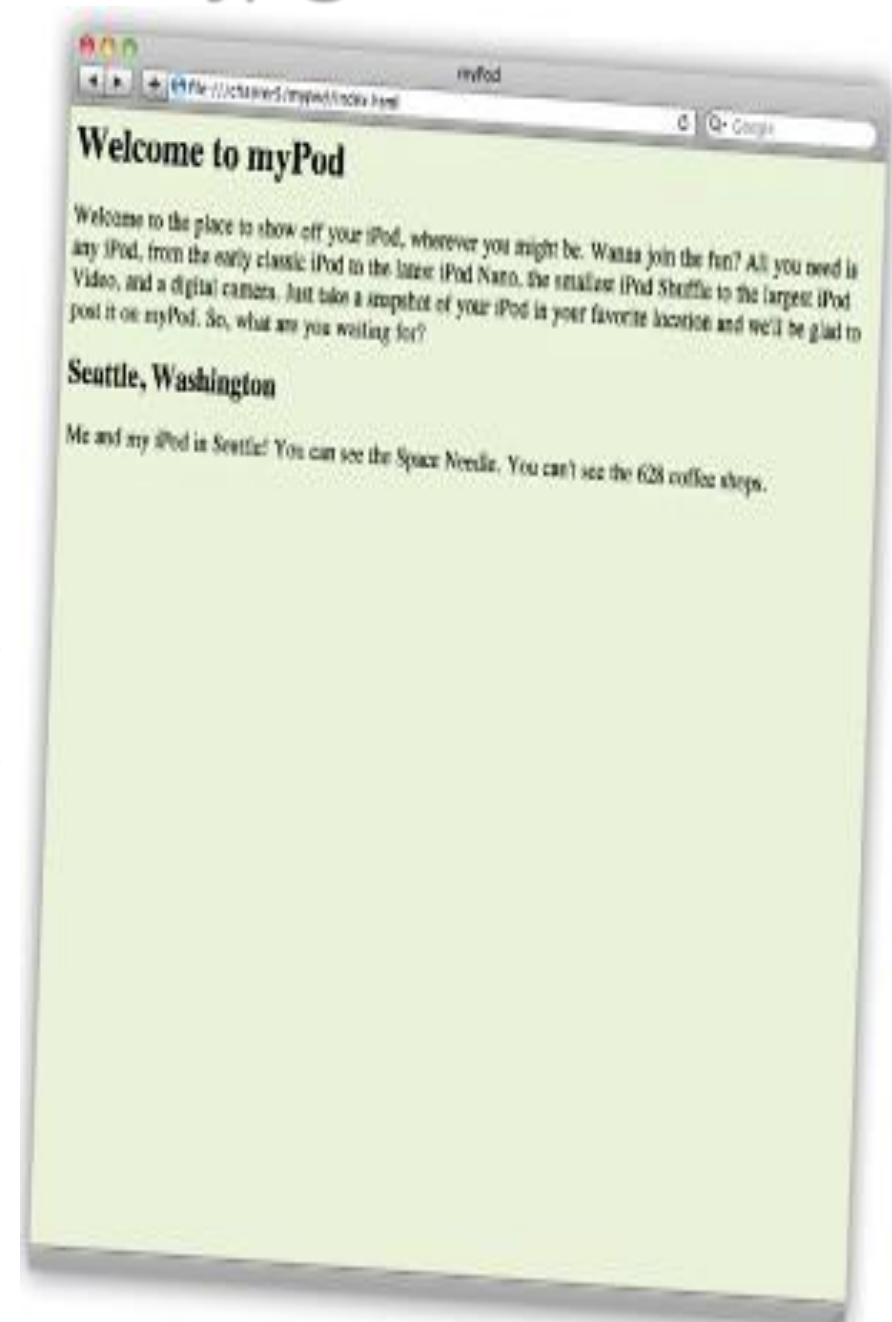
```
<html>
  <head>
    <title>Sharpen your pencil trivia</title>
  </head>
  <body>
    <p>How long a line can you draw with the typical
      pencil?</p>
    <p>
      Fill in HTML to display image with answer at the URL:
      http://wickedlysmart.com/hfhtmlcss/trivia/pencil.png here.
    </p>
  </body>
</html>
```

# Couple of image attributes

- Always use the `alt` attribute of the `img` element to supply a text alternative for the image. If you have a broken image because e.g., the image can't be found or if the user stopped the page from loading before the image could be retrieved, this is the text that will be displayed. It's also useful for the visually impaired since it can be read by the computer.
  - ❑ Example: `<img src = "logo.gif" alt = "Company logo">`
- The `width` and `height` attributes can be used to specify the desired dimensions of the image in pixels, scaling the image if necessary. Setting these to be the actual dimensions of the image is a good use of them as in this case, you are specifying structure and allow the rest of the page to layout correctly around it before the image loads.
  - ❑ Using it for scaling purposes is not recommended because...
    - A. Then you are specifying presentation in HTML, which should be kept separate in the CSS;
    - B. If you are scaling to larger dimensions, the image will look grainy.
    - C. If you are scaling to smaller dimensions, then you are missing out on an alternative to decrease its file size (and hence loading time) by supplying a separate smaller version of the image.

# myPod Code: Add image seattle\_video.jpg to bottom.

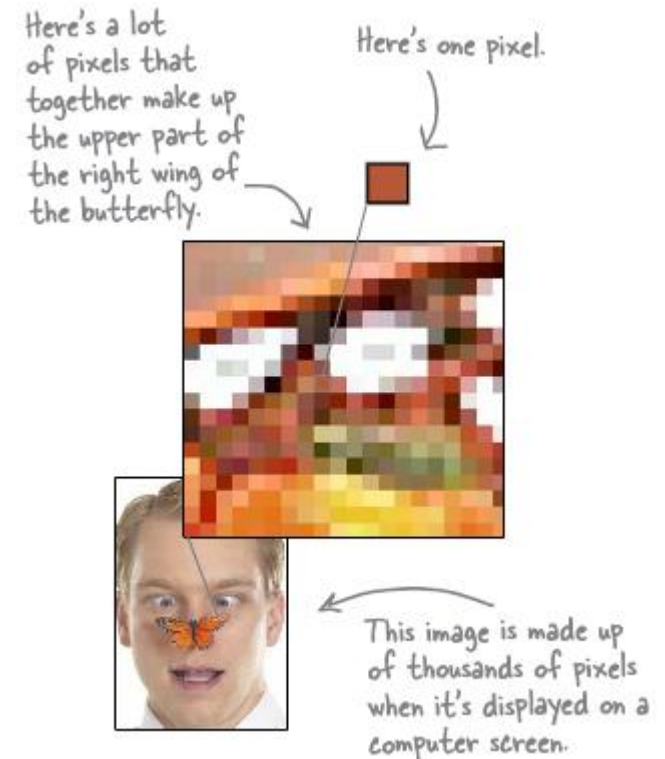
```
<html>
  <head>
    <title>myPod</title>
    <style type="text/css">
      body { background-color: #eaf3da;}
    </style>
  </head>
  <body>
    <h1>Welcome to myPod</h1>
    <p>Welcome to the place to show off your iPod, wherever you might be.
      Wanna join the fun? All you need is any iPod from the early classic
      iPod to the latest iPod Nano, the smallest iPod Shuffle to the largest
      iPod Video, and a digital camera. Just take a snapshot of your iPod in
      your favorite location and we'll be glad to post it on myPod. So, what
      are you waiting for?</p>
    <h2>Seattle, Washington</h2>
    <p>Me and my iPod in Seattle! You can see the Space Needle. You can't see
      the 628 coffee shops.</p>
    <p><img src = "photos/seattle_video.jpg"></p>
  </body>
</html>
```



# Pixels

- The computer screen is made up of a grid of pixels, which you can think of as tiny rectangles that can only display a single color at a time.\* In most cases, you can't even tell that images are made up of these rectangular blocks of single colors unless you zoom in. Higher resolutions mean more pixels per square inch.
- Most (desktop) browser windows span 800 – 1200 pixels so you should keep your image widths at a maximum of 800 pixels so that horizontal scrolling isn't necessary. But this is different for mobile (more on adjustments later).
- An estimate is 96 pixels per inch (ppi), but with higher resolution monitors and retinal displays, this number goes up.

\* Actually, there are subpixels but you generally only set colors of whole pixels.



# Exercise: myPod (cont.)

- I. Use an image editing application such as Paint on Windows (in Accessories) to open the `seattle_video.jpg` image and resize it to half of its size (600 X 400). Then Save **AS** `seattle_video_med.jpg`.
- II. Change the image source to this new image and set the `alt` attribute to the value, "My video iPod in Seattle, WA" and set the width and height attributes to 600 and 400, respectively. Then reload the page.
- III. Then add the images, `seattle_classic.jpg`, `seattle_shuffle.jpg`, and `seattle_downtown.jpg` with the respective alternates of "A classic iPod in Seattle, WA", "An iPod Shuffle in Seattle, Wa" and "An iPod in downtown Seattle, WA" below the first. Surround the text with the `<p>` and `</p>` tags.



# Exercise: myPod (cont.)

- IV. Below the added pictures, add the heading, “Birmingham, England” and the paragraph text:  
“Here are some iPod photos around Birmingham. We've obviously got some passionate folks over here who love their iPods. Check out the classic red British telephone box!”
- V. Then below this, add a new paragraph element with the two photos, `britain.jpg` and `applestore.jpg` in the photos directory. Include appropriate width and height attributes and alt text of, “An iPod in Birmingham at a telephone box” and “An iPod at the Birmingham Apple store”.

# Exercise with thumbnails

- Thumbnails of images are smaller versions of them.
- I. Create a new thumbnails directory and resize the first four images to 150 X 100 pixels. Then save **AS** the same name but in the thumbnails directory. For the two Birmingham photos, either resize to 100 X 150 or crop to images of that size and save them in the thumbnails directory as well.
- II. Update the image attributes to load these thumbnail images at the appropriate sizes. Reload the page and note how the four images and two images now line up in a row. This is because the `img` element is an inline element and the browser window now has enough space to line them up.
- III. Add an html directory in the mypod folder which will contain the documents with the larger photos and descriptions.

# Exercise with thumbnails (cont.)

- IV. Open up a new text document and enter the following HTML. Then save it as `seattle_downtown.html` in the `html` directory:

```
<html>
  <head>
    <title>myPod: Seattle Ferry</title>
    <style type="text/css"> body { background-color:
#eaf3da; } </style>
  </head>
  <body>
    <h1>Seattle Ferry</h1>
    <p>
      
    </p>
  </body>
</html>
```

## Exercise with thumbnails (cont.)

- V. Use the previous page as a template for your other pages which will be linked to by the other images from the main index.html page.
- VI. Turn all of the thumbnail images on the main page into links pointing to the corresponding web pages containing the full sized images. You can turn images into links in the same way you do for text.

# Adding transparent logo

- If you have Photoshop, open mypod.psd in mypod/logo. The checkers represent the transparent part.
- Change the matte color of the image so that it matches the web page background color (with RGB hex code eaf3da). A “matte” is used to “soften the text’s edges against the background color.” But using the wrong color matte means you will see halos.
- The colors are formed by “adding” the intensities of red, green and blue (from 0 to 255) first two digits of the hex code represent the red (ea), the next two the green (3d) and the last two the blue (da). In hexadecimal, we have 16 digits and we use A through F for the digits 10 through 15. To get the number in our familiar decimal format, multiply the left digit by 16 and add it to the right number.
  - ❑ Example:  $3d = 3 * 16 + 13 = 48 + 13 = 61$  (A = 10, B = 11, C = 12, D = 13, E = 14, F = 15)
- Since the logo has a transparent background, save it as a GIF or PNG-8, being sure to check the Transparency option. Then add it to the top of your main page:

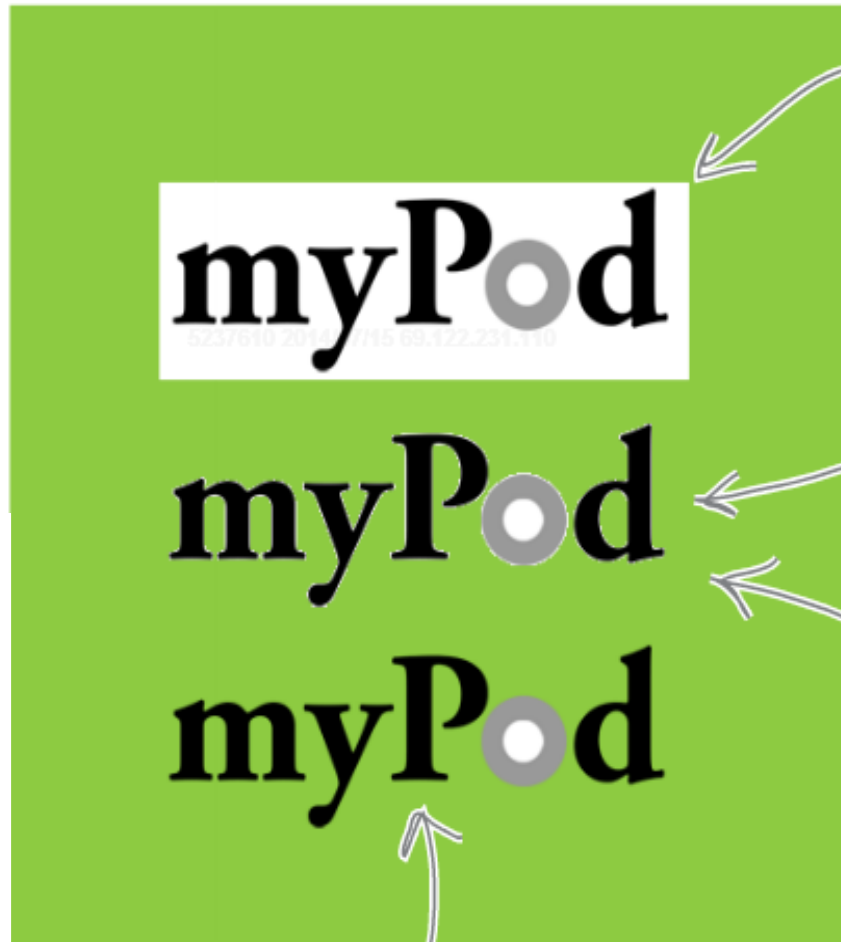
<p>

``

</p>

# Logos (with different transparency)

Here's the logo saved in three different ways and displayed on a web page with a green background.



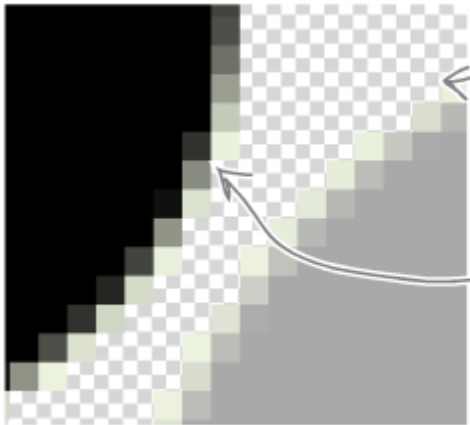
Without transparency, things look pretty bad. Clearly, a white background isn't going to work on a green web page. (It might, however, work just fine on a white web page).

Here's what we get if we check Transparency and save. Better... but what's that white "halo" around the letters in the logo?

The halos happen because the photo editing application creates a "matte" to soften the text's edges against the background color. When it did that for this logo, however, it assumed it was softening the edges against a white background.

Ah, now we're talking; this looks great. For this version, we told Photoshop Elements to create the matte around the text using a green background. How? We'll show you next.

Now, when you look close up at the logo, you'll see the matte matches the green color in the background of the myPod web page.



# Softening text edges

**Q:** Why does the text need its edges softened?

**A:** Check out the two versions of the myPod logo below:



myPod  
myPod

You'll see the top version has very hard, jagged edges and is less readable. This is the way text is displayed by default on a computer screen. The second version has had its edges softened using a technique called *anti-aliasing*. Words that are anti-aliased on a computer screen are more readable and more pleasant to the eye.

# Notes

- ❑ This is primarily a summary of Chapter 5 of *Head First HTML and CSS*, 2<sup>nd</sup> Edition by Elisabeth Robson and Eric Freeman, 2012. It contains images, exercises, and code from the book.