COMP 1950

Web Development and Design 2

Day 05

Day 04 – Review Exercise

- Open up the day-04-review-start folder
- Style the page to match styles shown on the projector screen
- The “Toyota” logo should grow in height when hovered. This size change should be animated with a transition
- The banner image should fade-in on page load
  - Use CSS animations for this
- The banner info text should slide-in from the left on page load
  - Use CSS animations for this
- The gallery images should zoom in when hovered. This should be animated with a transition

Agenda – Day 05

- Day 04 Review
- Responsive Images
- Responsive Menu Systems

Responsive Images

- Due to the multitude of screen sizes and screen densities it is difficult to serve optimized images for all devices
  - Factors to consider when serving images to devices
    - Are they on a hi-dpi device (retina)
      - If yes, should they get the hi-dpi or retina version of the image
      - What if they are on a slow or expensive connection?
      - Should users on regular devices be forced to download “retina” versions of images
    - Should we use vector graphics wherever possible
    - Is it appropriate to use background images or are foreground images required
  - The better you know your audience the better decisions you can make in regards to most of the above questions

Retina

Nokia N770 – first retina device

Image source: Retinafy your web sites & apps by Thomas Fuchs
Text source: Retinafy your web sites & apps by Thomas Fuchs

What Does Retina Mean?

- “Retina” is a marketing term created by Apple to denote devices that are of sufficiently high resolution that any additional increase in resolution would not be perceptible to the human eye in normal use[1]
- Non apple devices can be retina as well, though they will often use the term Hi-DPI (dots (pixels) per inch)
- Although Retina screens have a higher total number of pixels, these pixels do not map directly to CSS pixels, instead most Apple retina devices will map 2 retina pixels to a single CSS device pixel horizontally and 2 retina pixel to a single CSS pixel vertically.
- This means that a 320 pixel wide image on a non-retina display will still appear to be 320 pixels wide on a retina display, with the difference being that the retina image will appear sharper with more detail, since it has 4 times the number of pixels (2 x width, 2 x height) to render a single CSS device pixel

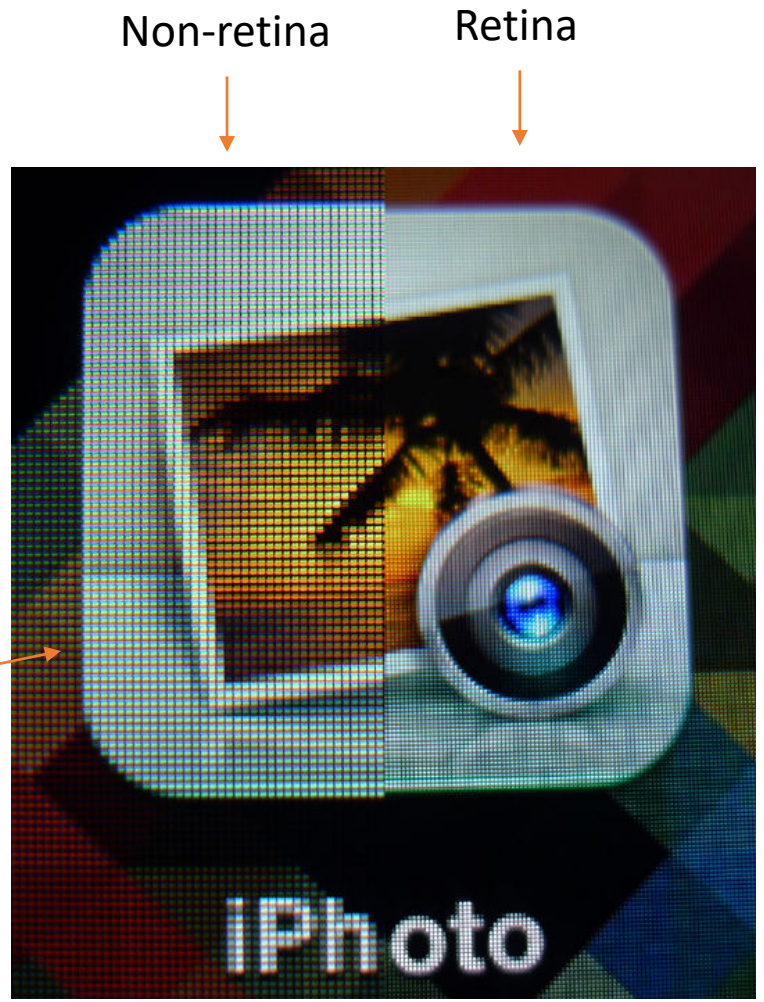
1. Definition source: http://www.mobileburn.com/definition.jsp?term=Retina

Retina = More sharpness and detail

Retina Pixels vs CSS Pixels

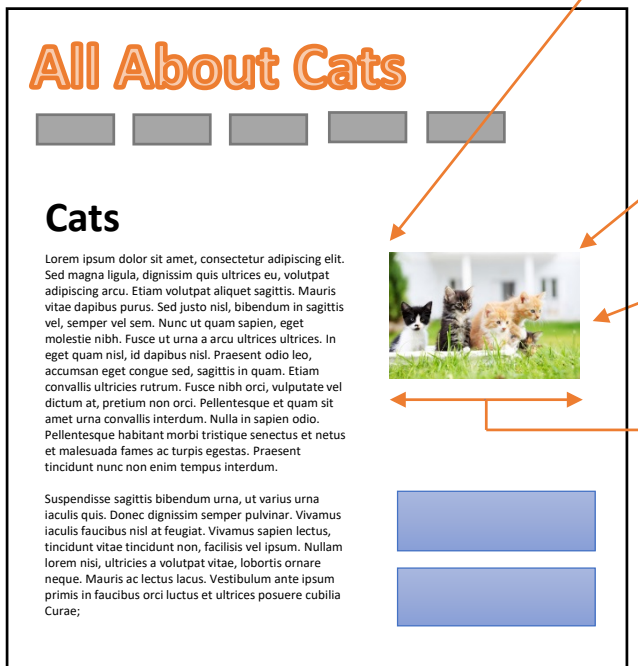
- Apple Retina pixels are made up of 4 CSS Pixels
- Widths and heights of elements are reported using CSS device pixels
- Retina devices will appear to have greater sharpness and clarity
- Even though items have twice the amount pixels the actual size on the screen based on CSS pixels is the same

Notice how both images are the same size, and the difference is that the images have different levels of sharpness and detail



Retina Pixels vs CSS Pixels

Retina optimized image – 500px wide

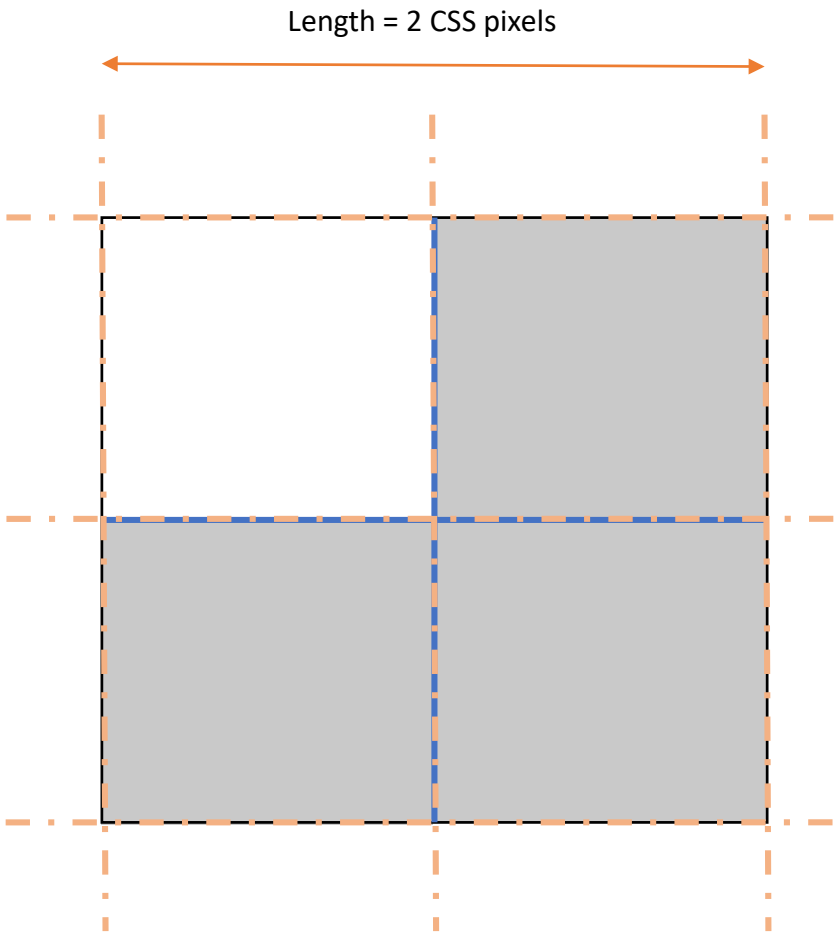


Width = 250 CSS device pixels

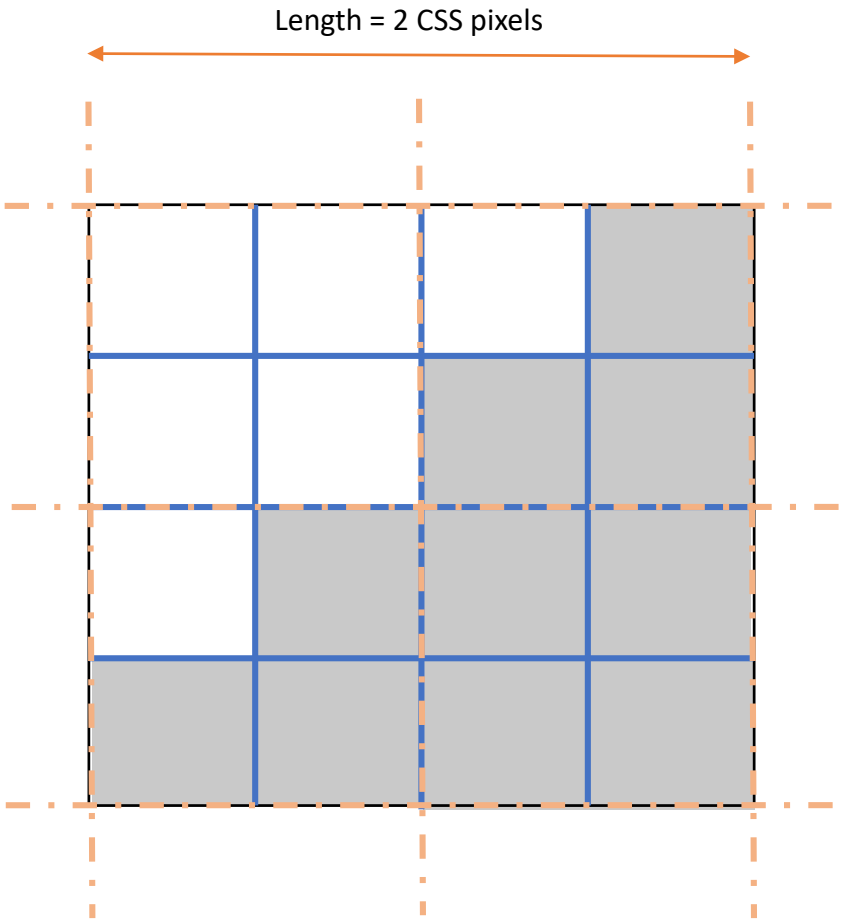
- The original 500px retina image is made to fit into a place on the web site that is 250 CSS device pixels wide
- Retina devices will use the entire 500px wide version of the image to render a sharp image to fit into the 250 CSS device pixel wide space
- Non retina devices will scale the 500px wide image to fit into the 250 CSS device pixel wide space

Retina Pixels vs CSS Device Pixels

Non-retina
1 pixel = 1 CSS Device Pixel



Retina
4 pixels = 1 CSS Device Pixel



Create Foreground Retina Images

- Create images for the largest use case (including retina)
- Make images flexible by allowing the browser to scale the images down
- To support retina devices make your images twice the required size needed in the layout
  - For example:
    - If your layout requires a profile picture to be 100px wide by 200px high then you should make your profile images 200px wide by 400px tall

Browser scales image down for smaller devices

Mobile image ← Desktop image ← Retina - 2x the actual size required in the page layout

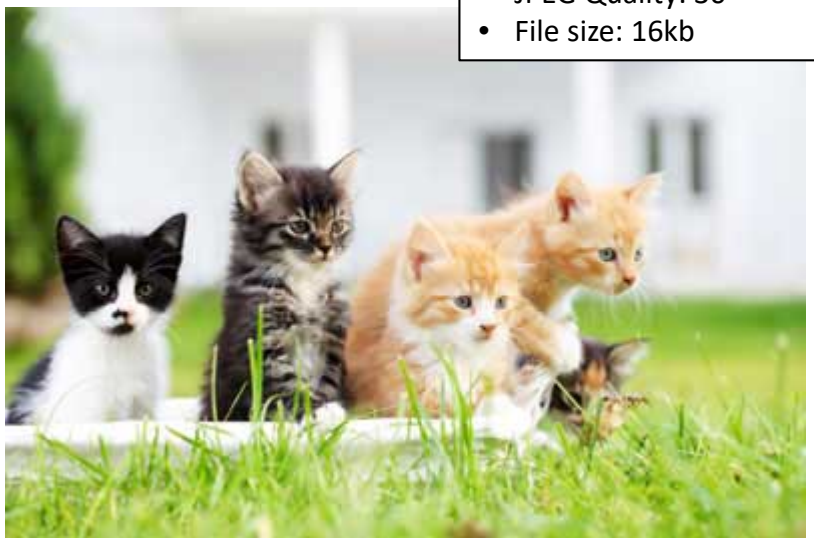
- **This techniques will work well for most images, but not all**
- **Images such as icons or text as images may not scale down well, in this case create different versions of the image for different devices, or consider other alternatives such as vector images**

Optimize JPEGs for Retina

- Creating retina images that will be served to all devices creates the problem of low resolution devices being served larger images with larger file sizes without any perceived benefits
- Fortunately with JPEGs we can lower the quality level, while maintaining a larger dimensional size for retina and keep the file size low enough that the difference in file size is smaller

- Size: 200px x 133px
- JPEG Quality: 80
- File size: 16kb

- Size: 400px x 265px
- JPEG Quality: 30
- File size: 16kb



Optimize JPEGs for Retina

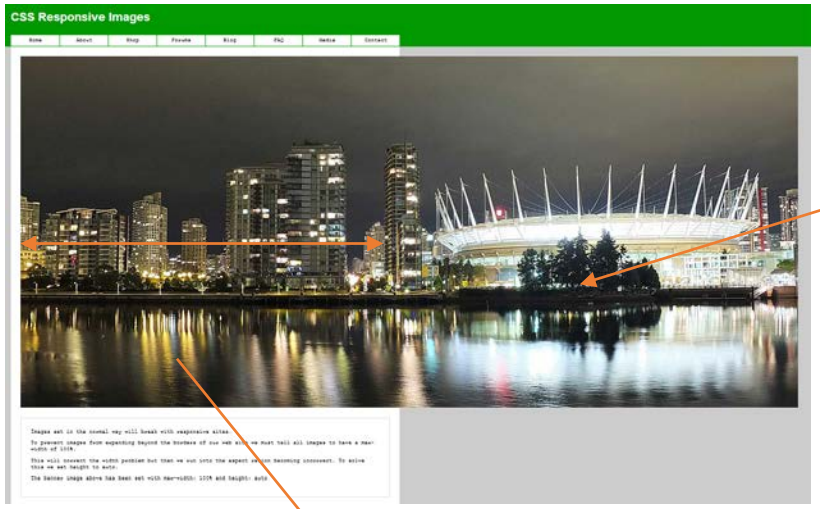
- Size: 400px x 265px
- JPEG Quality: 30
- File size: 16kb

- Size: 200px x 133px
- JPEG Quality: 80
- File size: 16kb

- Large 400px x 265px image scaled down to fit into a 200px by 133px space
- JPEG Quality: 30
- File size: 16kb

Coding Responsive Images

- Since the actual image size of our images will often be larger than the actual viewport that our images will fit into we need to style our images so that they stay contained within their parent elements
- By default images will display at their actual dimensions



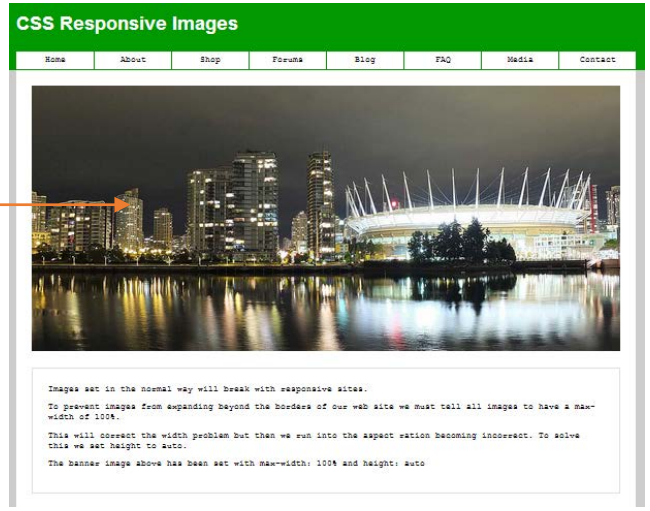
By default images will expand beyond their parent container if they are too large to fit inside their parents container

Width of parent container

Coding Responsive Images

- To prevent images from expanding beyond their image container and to make them flexible for display across a variety of devices from desktop to mobile you can use the following CSS code in your CSS file

```
img {
      max-width: 100%;
      height: auto;
}
```



- **The above code tells the image to only go as large as is possible to completely fit inside the parent element**
- **The image will only go as large as the images actual width or the width of the parent, which ever comes first**
- **The “height: auto;” tells the browser to maintain the height of the image relative to it’s width**

Srcset, Sizes and Picture

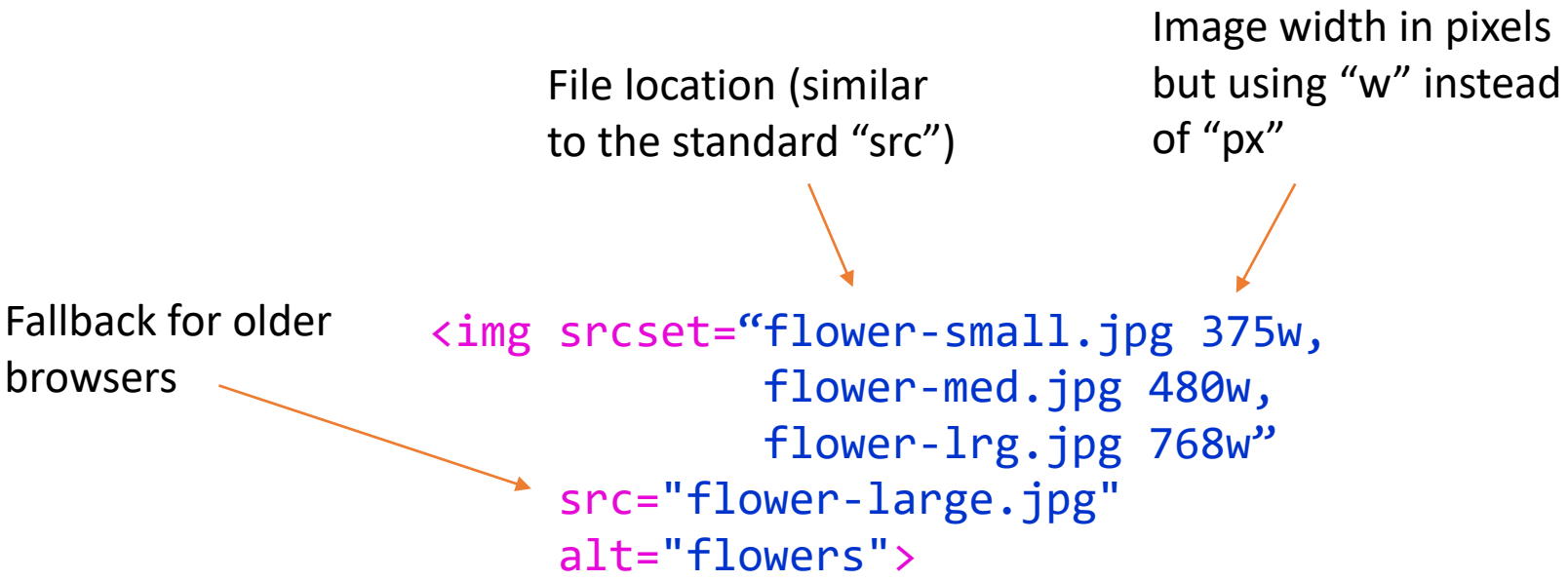
- In 2014 the Responsive Images Group developed some new image attribute and new image elements for serving different images to devices with different device widths

- **Visit The Responsive Images Community Group for information on these new attributes and elements**

http://responsiveimages.org

The New Image Attributes

- srcset:
  - The srcset attribute allows the web designer to declare several images for the browser to choose from
  - The web designer also passes in information about the images width, this helps the browser decide which image to choose
    - The image width should be the actual width of the image in pixels (but written as 300w – where 300 is the pixel width and “w” is unit used



The New Image Attributes

- sizes:
  - The sizes attribute gives the browser information about how large the image will be on the page.
  - We can tell the browser how large the image will be at various screen widths by using a sort of miniature media query along with the size of the image on the page
  - “vw” unit stands for “viewport width”
    - 1vw = 1% of the viewport width
  - You can specify other CSS lengths as well
  - You can also specify a pixel density for the images listed in the srcset attribute using #x, for example: 1x (for non-retina) and 2x for retina
    - srcset=flower@1x.jpg 1x, flower@2x.jpg 2x
  - You can not mix the “w” units with the “x” value, use one or the other

```
<img srcset="flower-small.jpg 375w,
             flower-med.jpg 480w,
             flower-lrg.jpg 768w"
     src="flower-large.jpg"
     sizes="(max-width: 1024px) 50vw,
            (max-width: 568px) 100vw,
            400px" alt="flowers">
```

media query to tell the browser what width to use a size at

The size that image needs to fill in the layout

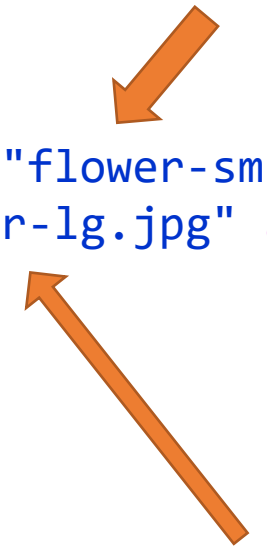
The Picture Element

- If you require complete control of when a certain image should be loaded at certain widths then you will need to use the "<picture>" element
- The picture element looks similar to the video or audio element
- Inside the picture element you use a "<source>" element to declare an image source. You can also set a "media" attribute inside the "<source>" element. This is where you pass a mini media querie that tells the browser when to load that image.
- The "<picture>" element is mostly used if you require art direction on your images
    - For example, you wanted to provided a tighter crop of an image on mobile devices
- See the next slide for a code example

The Picture Element

The sources that browser will pull the images from. The source elements also can include media queries which tell the browser when to load which image

```
<picture>
  <source srcset="flower-sm.jpg" media="(max-width: 400px)">
  <img src="flower-lg.jpg" alt="flower">
</picture>
```



This is used as a fallback for older browsers, or as the default image if no other image in the sources is loaded due to none of their media queries conditions resulting in true

Background Images

- You can load different background images for different sized devices through the use of media queries
- When a media querie resolves to true the browser will load the background image that is specified inside that media querie
- Never code content (foreground) images as background images. Content based images (such as an image related to the content of an article) should always be coded using the "<img>" tag or the "<picture>" element

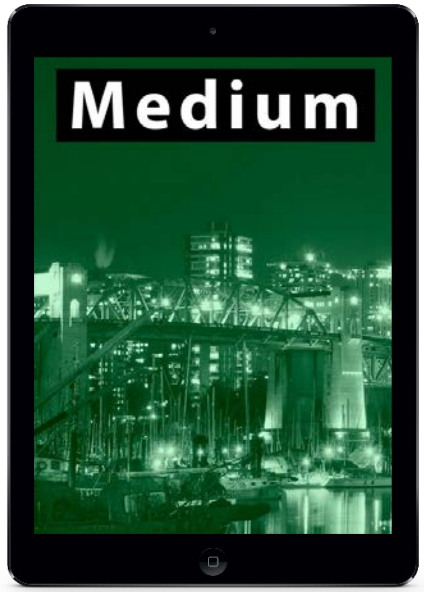
Background Images

```
@media only screen and (min-width: 1000px) {
  body {
    background-image: url(bg-large.jpg);
  }
}

@media only screen and (max-width: 999px) {
  body {
   background-image: url(bg-medium.jpg);
  }
}

@media only screen and (max-width: 500px){
  body {
    background-image: url(bg-small.jpg);
  }
}
```

768px wide



SVG

- SVG stands for Scalable Vector Graphics
- Vector Graphics differ from bitmapped graphics like JPG and PNG in that they use lines and shapes to layout the graphics on screen.
- Vector graphics are resolution independent which makes them ideal for responsive design since one SVG image can be used for small screens all the way up to large monitors without loss of quality
- Adobe Illustrator can be used to create SVG graphics
- SVG are suitable for icons, logos and other "graphic" elements of a web page
- SVG should NOT be used for photographic images
- SVGs can be included onto a web page in one of three ways
  - With a standard "<img> tag
  - Placing the SVG code directly into an HTML5 document
  - Linking to an external SVG source from an SVG element in an HTML5 document

Online Image Compressors

- JPEGs:
  - http://www.jpegmini.com
- PNGs:
  - http://tinypng.org

Retina Media Query

- To target only retina devices in your CSS use the following media querie

```
@media
only screen and (-webkit-min-device-pixel-ratio: 2),
only screen and ( min--moz-device-pixel-ratio: 2),
only screen and ( -o-min-device-pixel-ratio: 2/1),
only screen and ( min-device-pixel-ratio: 2),
only screen and ( min-resolution: 192dpi),
only screen and ( min-resolution: 2dppx) {

        /* Retina styles go here */

}
```