3. Retina display in iOS devices.

**Retina Display** (marketed by Apple with a lowercase 'D' as **Retina display**) is a brand name used by [Apple](https://en.wikipedia.org/wiki/Apple_Inc.) for screens that have a higher[pixel density](https://en.wikipedia.org/wiki/Pixel_density) than their previous models.

The goal of Retina Displays is to make the display of text and images extremely crisp, so pixels are not visible to the naked eye. This allows displays to rival the smooth curves and sharpness of printed text and immediacy of photographic prints.

The first, and most important is pixel density. When Steve Jobs launched the iPhone 4, and with it the first Retina display, he described it as having a screen with so many pixels packed closely together that they were imperceptible to the human eye at a distance of twelve inches. He went to great lengths to explain that, because the iPhone 4’s screen packed in 300 pixels per inch, most people wouldn’t see them at all when the phone was a foot from their eyes.

Since then, Apple has launched several devices with Retina displays. Some have pixel densities of more than 300 pixels per inch, some with less. How can they all be called Retina? Because there are two crucial elements to whether or not pixels are perceptible: density and distance. The further your eyes are from the screen, the lower the pixel density needed to make the pixels ‘disappear.’

Generally speaking, the bigger the screen, the further your eyes are likely to be from it and so the lower the pixel density required to ‘qualify’ as a Retina display.

So, for example, iPhones from 4 to 5s had a pixel density of 326 pixels per inch, while the current 27in iMac has only 218 pixels per inch. Given that, for most of us, the distance we sit from a 27in screen is more than one and a half times the distance at which we hold an an iPhone from our eyes, the lower pixel density is irrelevant.

How does a Retina display compare to non-Retina?

In simple terms, Apple ‘converts’ a device’s display to Retina by doubling the number of pixels vertically and horizontal, meaning it has four times as many pixels as its non-Retina counterpart.

If it did that and nothing else, however, there would be a problem. User interface elements like menus and icons would look tiny. To compensate for this, Apple created what it calls HiDPI mode, where each interface element is doubled in size vertically and horizontally and so appears at the same size as it would on a non-Retina display.

The effect of a Retina display is to make everything look more crisp. Text, especially, benefits from Retina – it looks smoother, with the curves on characters looking like curves instead of jagged steps.

## What’s Retina HD and Retina 4K/5K?

Steve Jobs painted Apple into something of a corner when he described the pixels in the iPhone 4 as imperceptible to the human eye. Where do you go from there? How do you describe a screen that’s even better? Apple’s answer has been to borrow terminology from the video and broadcast industries.

When the iPhone 6 came out in 2014, it had a vertical resolution of 750 pixels, a little more than the 720 pixels which forms one of the standards for HD video. The iPhone 6 Plus had a vertical resolution of 1080 pixels, exactly the number of pixels needed for the higher of the two HD video standards. Never one to miss an opportunity for a catchy label, Apple labelled the iPhone 6 and 6 Plus’ displays ‘Retina HD.’ It did the same in 2015 with the 6s and 6s Plus when they launched in 2015 with the same screen resolutions.

# CSS Techniques for Retina Displays

## Pixels and Screen Density

When we speak about pixels, they are nothing but the smallest physical unit in a display. It is a minute area of illumination which when combined with many other pixels form an image. Each pixel has its own brightness level and colors standard as instructed by the operating system. It is the imperceptible distance between two pixels that makes the image look sharper and bright.

Screen density on the other hand refers to the total number of such pixels that physically appear on a display screen. It is generally measured in pixels per inch format popularly known as PPI.

The term **Retina** mentioned in the title of this post is a friendly word used by Apple to lay emphasis on the double density pixels screen of its devices. On such high resolution display screens like of Retina Ipad and MacBook Pro, it is nearly impossible for a human eye to see the pixels from a normal viewing distance.

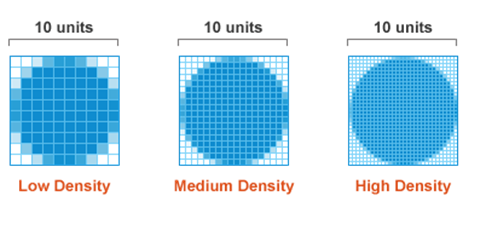
## CSS Pixel

CSS pixel is an abstract unit used by the browsers to draw images and other content on a web page. CSS pixels are DIPs which means they are device independent pixels. They readjust themselves according to the pixel density of the screen they are rendered in.

If we have the following piece of code:

<div style=”width:150px; height:200px”></div>

The above HTML component would look 150px by 200px in size in a standard display while 300px by 400px in a Retina display to retain the same physical size.



## Resizing Images

### **1)      Using alternate high resolution pixels**

Suppose we have an image of 200px by 400px (CSS pixels) and we want to display it properly in a Retina display. We can upload an alternate image of size 400px by 800px in our server and render them whenever the webpage is opened in a Retina device.

But to make them appear physically of the same dimensions we have to use CSS to resize them. The following is the piece of code that explains this.

/\* for low resolution display \*/

.image {

background-image: url(/path/to/my/lowreslogo.png);

background-size: 200px 300px;

height: 300px;

width: 200px;

}

/\* for high resolution display \*/

@media only screen and (min--moz-device-pixel-ratio: 2),

only screen and (-o-min-device-pixel-ratio: 2/1),

only screen and (-webkit-min-device-pixel-ratio: 2),

only screen and (min-device-pixel-ratio: 2) {

.image {

background: url(/path/to/my/highreslogo.png) no-repeat;

background-size: 200px 400px;

/\* rest of your styles... \*/

}

}

### **2)      Using @face-fonts instead of images icon**

If you are displaying a large number of icons in your webpage, then resizing each on them with a double resolution icon will be a hectic job. We will try to use @font-faces instead of images. These image fonts will automatically resize themselves on the high resolution devices just like normal fonts do. Using @face-fonts is an alternate solution to Bitmap icons.

Today, there are many good quality icon fonts that can meet your requirements for website design. Examples of such fonts are Fontello and Inkscape.

The following code shows how to use @font-faces as a replacement to image icons:

#### **In HTML:**

/\* define span tag for letters and give them a class in HTML \*/

<span class=”myicon”>d</span>

#### **In CSS:**

/\* First import your font \*/

@font-family: myFont;

src: url('Modern\_Icons.ttf'),

url('Modern\_Icons.eot'); /\* IE9 \*/

. myicon {

font-family: ‘Modern Icons’;

}

### **3)      Using SVG images instead of Bitmap images**

Bitmap images are raster images that multiply their pixels in Retina displays. But they come with a limitation of getting multiplied infinite number of times. This is where SVG images come to role. They also solve our problem of uploading alternate images of double resolution plus they solve the bandwidth problem too!

#### **In HTML:**

<img src="example.svg" width="150" height="200"/>

#### **In CSS:**

.image {

background-image: url(example.svg);

background-size: 150px 200px;

height: 150px;

width: 200px;

}

### **4)      Using JavaScript to replace all the images with double sized image**

As mentioned, replacing low resolution images with double resolution images consumes extra bandwidth and the website will load slowly. Still, if you want to go with the first method as discussed above, then replacing each one of them by writing individual code will be a difficult task.

We can use JavaScript to replace all images in a webpage. The following code explains this.

$(document).ready(function(){

if (window.devicePixelRatio > 1) {

var lowresImages = $('img');

images.each(function(i) {

var lowres = $(this).attr('src');

var highres = lowres.replace(".", "@2x.");

$(this).attr('src', highres);

}); }

});

The above code assumes that you have named the images with low resolution as myimage.png and high resolution images as myimage@2x.png. It finds the first dot in the image name and replaces it withmyimage@2x.

One of the major disadvantages of using the above JavaScript method is that the Retina display device would have to download both versions of images every time the page loads. This will surely affect the loading time of your website.

In my opinion, you must go with vector images method (SVG) as their size is low. For icon images you can use @font-faces method. These two methods will definitely make your website shine and lively in Retina devices.

## Working with high resolution favicons

Today favicons are used in multiple purposes both in standard devices as well as Retina displays. Favicons are used as bookmarks on home screen in Apple devices. So they have to be of better quality. You can export both 16px and 32px versions to make a favicon Retina ready. Use Apple’s Icon composer,**Graphic Tools in XCode**, for a good Retina favicon.