

Fundamentals of Web Development

Third Edition by Randy Connolly and Ricardo Hoar



Chapter 6

Web Media

In this chapter you will learn . . .

- The two different ways to digitally represent graphic information
- The different color models
- Color depth, image size, and resolution
- The different graphic file formats
- The different audio and video file formats
- How HTML5 provides support for audio and video

Representing Digital Images

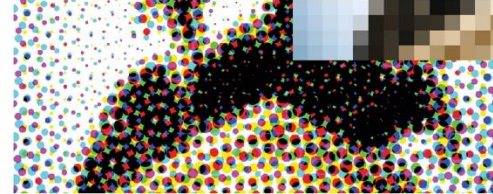
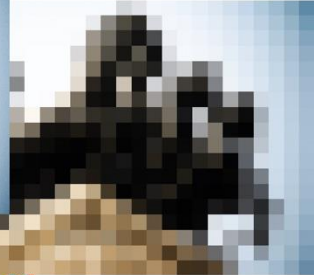
Images on your desktop monitor or your mobile screen are small squares of colored light called **pixels** that are arranged in a two-dimensional grid.

In contrast, printed images are not created from pixels, but from small overlapping dots usually called **halftones**

Original photographic image



Output as pixels
(size exaggerated)



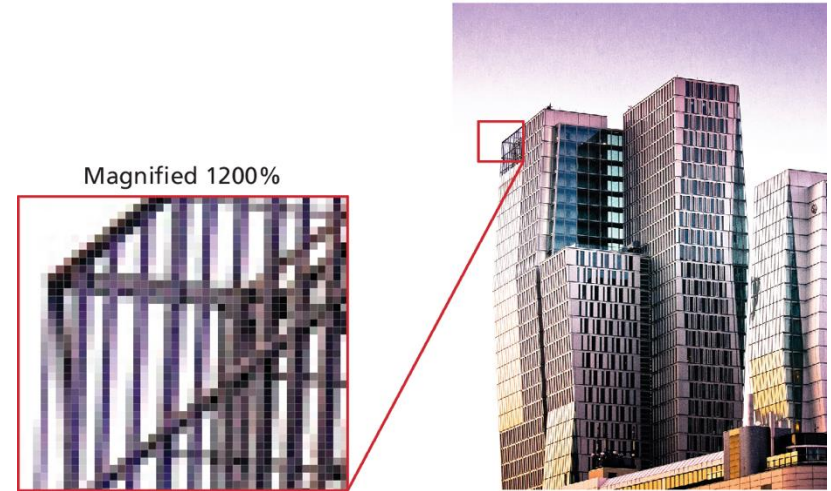
Output as halftones
(size exaggerated)

Raster images

In a **raster image** (also called a **bitmap image**) the smaller components are pixels.

Each colored square uses a number that represents its color value.

Raster images can be manipulated on a pixel-by-pixel basis by programs such as Adobe Photoshop, Apple Aperture, Microsoft Paint, or the opensource GIMP

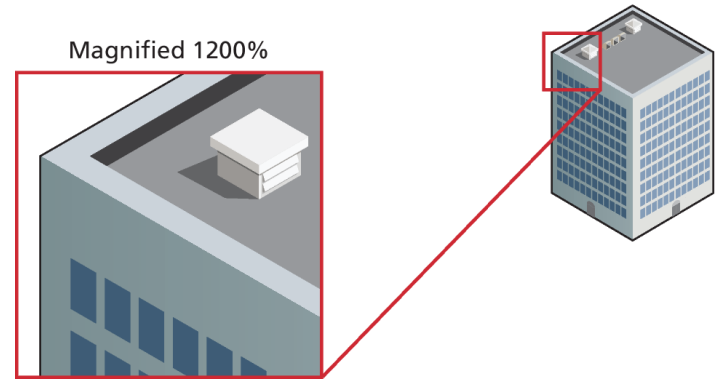


Vector images

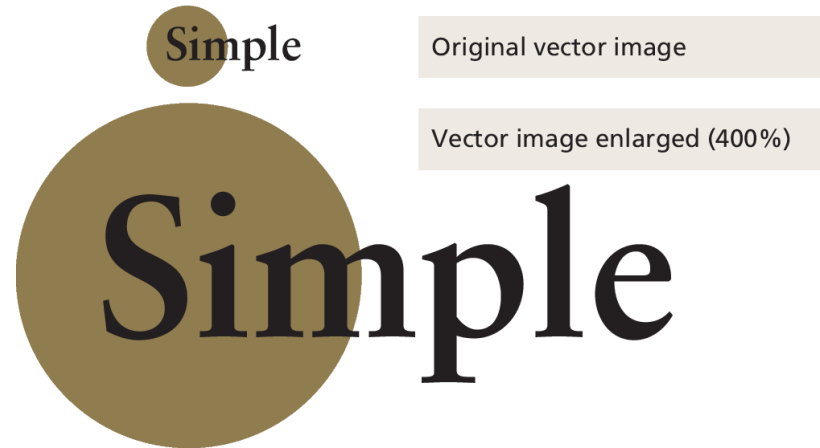
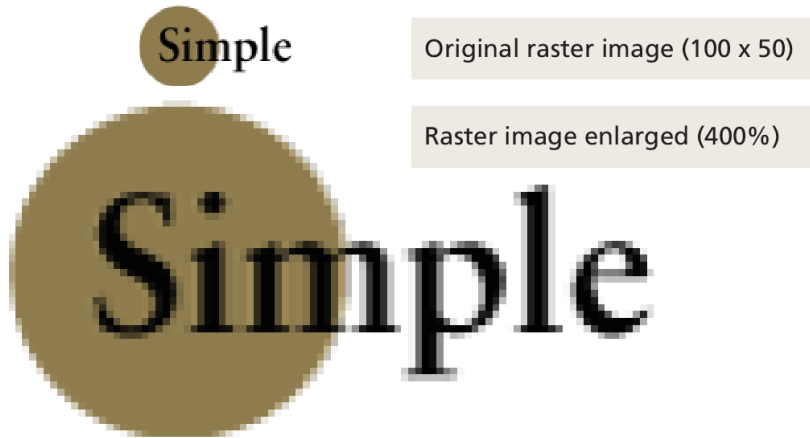
A **vector image** is composed of objects such as lines, circles, Bezier curves, and polygons.

Vector images are resolution independent and can be shrunk or enlarged without a loss of quality

Software includes Adobe Illustrator, Microsoft Visio, Adobe Animate (formerly Adobe Flash), Affinity Designer (Mac only), and the open-source Inkscape



Resizing raster images versus vector images



Color Models

There are many ways to describe color in web development.

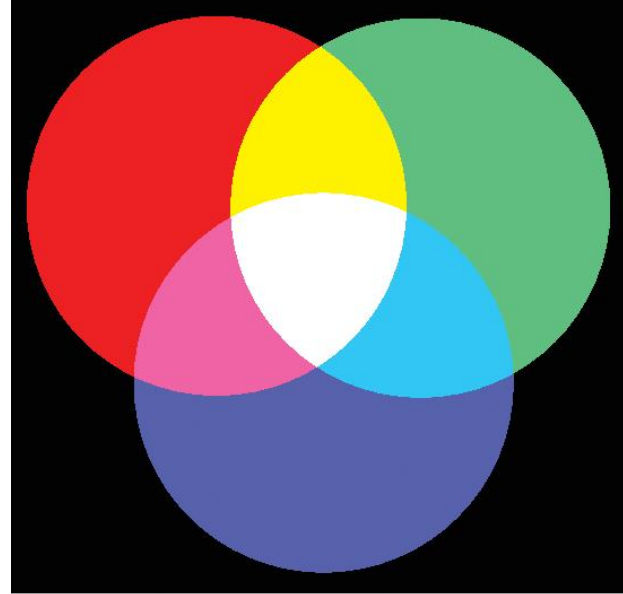
- Names Colors
- RGB (Red Green Blue)
- CMYK (Cyan-Magenta-Yellow-Key)
- HSL (Hue-Saturation-Lightness)

RGB Color Model

The **RGB color model** (Red Green Blue) relies on the fact that human visible color spectrum can be displayed using a combination of red, green, and blue lights

Each pixel is composed of tiny red, green, and blue subpixels.

Because the RGB colors combine to create white, they are also called **additive colors**.



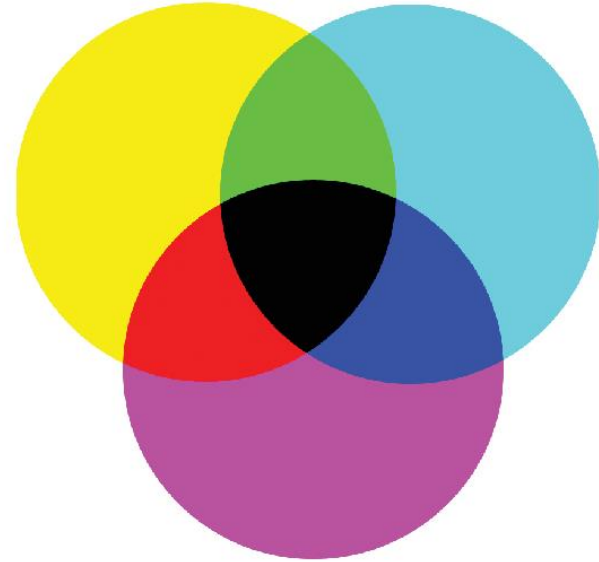
A

CMYK Color Model

In the **CMYK color model** for Cyan-Magenta-Yellow-Key (or black) color is created through overlapping cyan, magenta, yellow, and black dots that create the illusion of the combined color

For this reason, these colors are called **subtractive colors**.

The practical consequence is that an RGB image might not look the same when it is printed on a CMYK device;

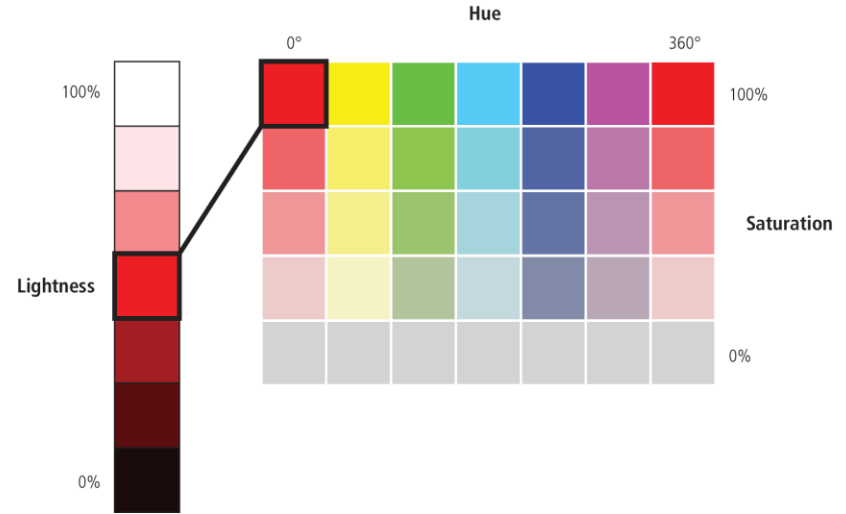


B

HSL color model

The **HSL color model** is more closely aligned to the way we generally talk about color. It breaks a color down into three components:

1. Hue (what we generally refer to as color)
2. Saturation (the intensity of a color)
3. Lightness (or Brightness)

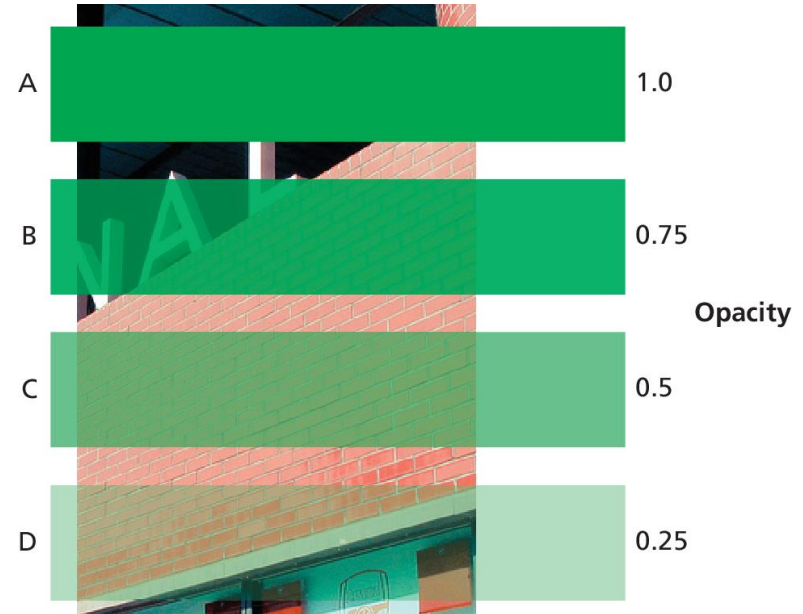


Opacity in CSS

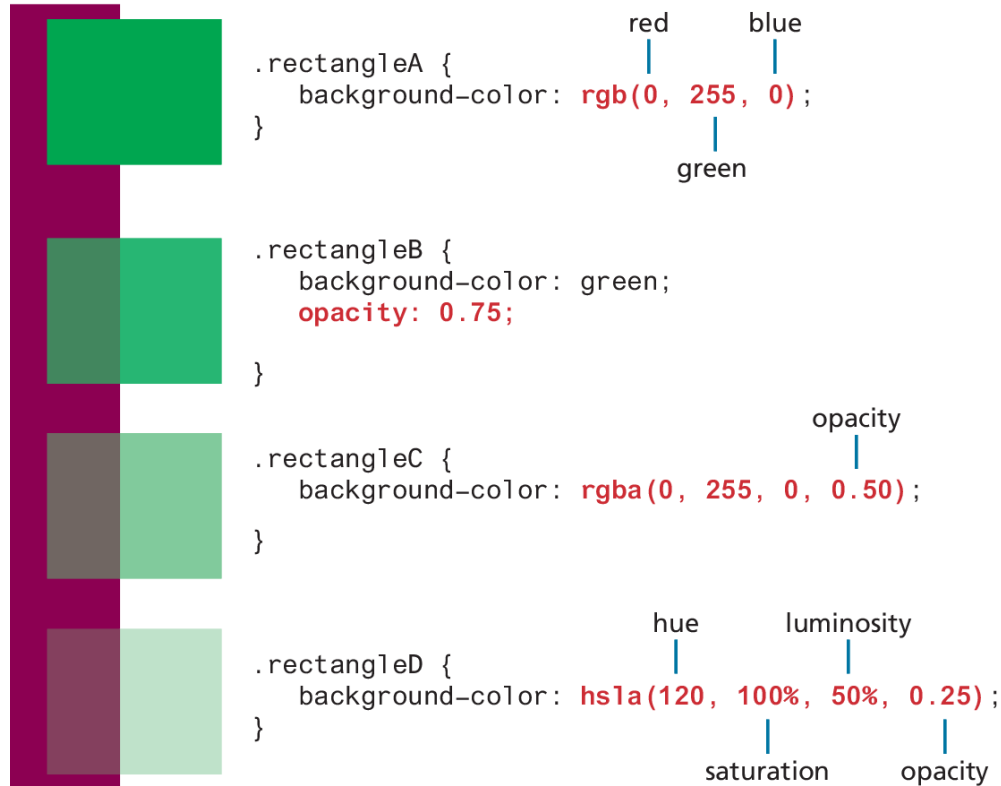
Opacity is, the degree of transparency in the color. (also referred to as **alpha transparency**)

In CSS, there is an opacity property that takes a value between 0 and 1.0.

- 0 means that the element is fully transparent.
- 100 means that the element is fully opaque—that is, it has no transparency.



Opacity using CSS



The diagram illustrates four CSS rectangles, each with a different opacity value and color notation. A vertical purple bar is on the left. Each rectangle is shown as a semi-transparent colored square over a solid colored square. The CSS code for each rectangle is shown to the right, with labels pointing to the values in the code.

- rectangleA**: `background-color: rgb(0, 255, 0);`
 - red points to 0
 - blue points to 0
 - green points to 255
- rectangleB**: `background-color: green; opacity: 0.75;`
- rectangleC**: `background-color: rgba(0, 255, 0, 0.50);`
 - opacity points to 0.50
- rectangleD**: `background-color: hsla(120, 100%, 50%, 0.25);`
 - hue points to 120
 - saturation points to 100%
 - luminosity points to 50%
 - opacity points to 0.25

Gradients

A **gradient** is a transition or blend between two or more colors.

Gradients use the **background-image** property.

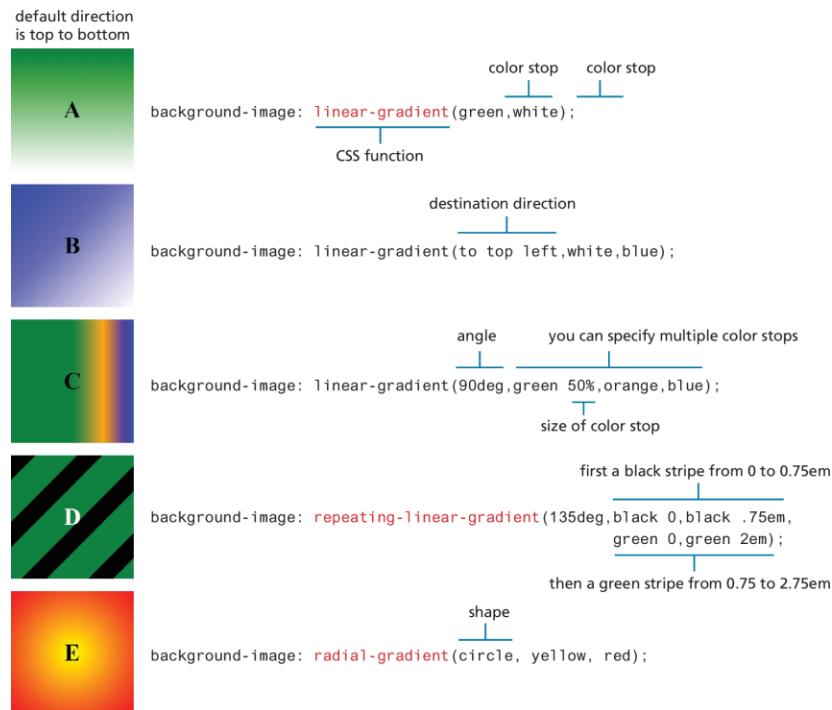


Image Concepts

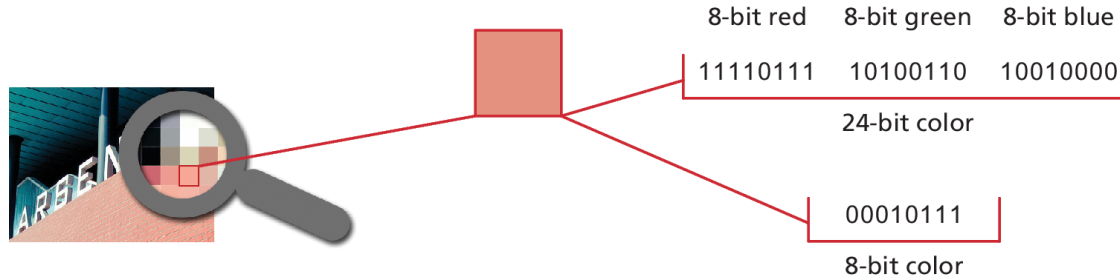
There are a number of other concepts that you should be familiar with in order to fully understand digital media.

- Color Depth
 - Dithering
- Image Size
 - Interpolating
- Display Resolution

Color Depth

Color depth refers to the maximum number of possible colors that an image can contain.

For raster images, this value is determined by the number of bits used to represent the color or tone information for each pixel

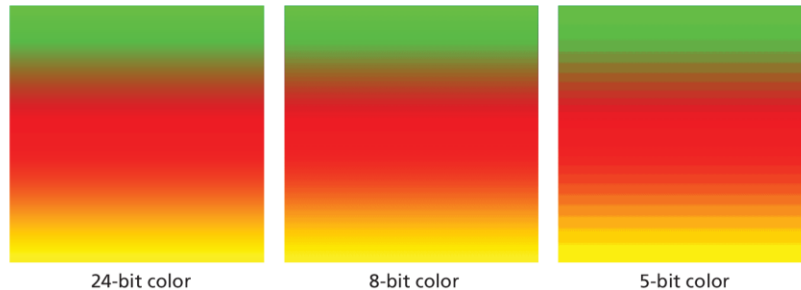


Color Depth (ii)

The more bits used to represent color, the more possible colors an image can contain.

- 8 bits = 256 colors
- 24 bits (true color) = 16.8 million

Most LCD monitors are only 18-bit display devices (262,144 colors) and use **dithering** to create the illusion of more color.



Notice the banding
due to the dithering
(dithering is more
obvious on screen
than on paper)

Image Size

Image size refers to how many pixels a raster image contains

The size of an image onscreen is determined by

- Pixel dimensions of the image,
- Monitor size, and the
- Computer's display resolution

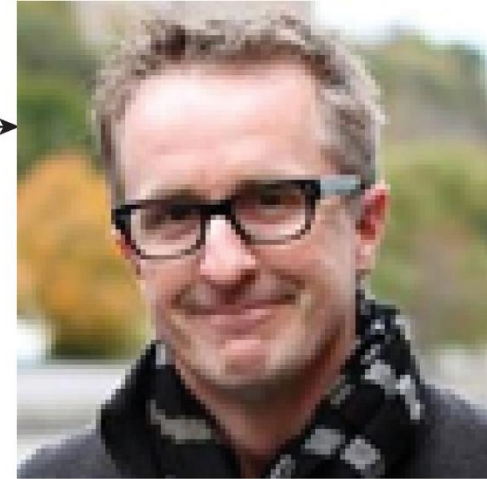
Whenever you resize, the program (the browser, Photoshop, etc) doing the resizing must **interpolate**

Interpolation and Resizing

*Resizing an image
always reduces its
quality*



Enlarging a small image a substantial amount will noticeably reduce its quality.



Decreasing the size of an image does reduce the quality as well, but it is not nearly as noticeable.

Resizing artwork

You can resize in the browser but be cautious.

You will get better results if you enlarge images in a dedicated image editing program (Figure 6.18)

Large images, scaled down in the browser use unnecessary bandwidth.



Original (200 x 50)



Enlarged in browser via
``



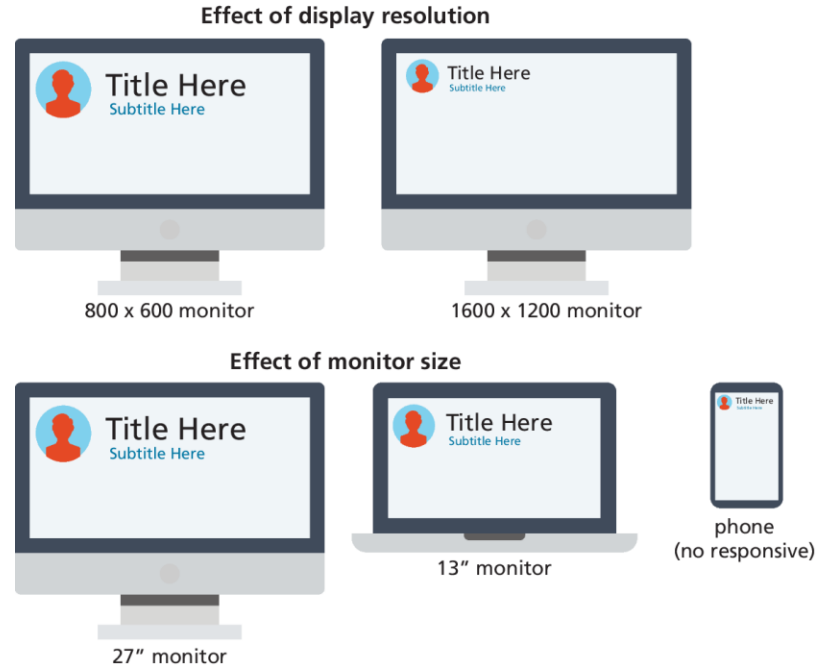
Enlarged original (600 x 150)

Display Resolution

The **display resolution** refers to how many pixels a device can display.

Some common display resolutions include 1920 × 1600 px, 1280 × 1024 px, 1024 × 768 px, and 320 × 480 px

A web page will appear smaller on a high-resolution system (and larger on a low-resolution system)



File Formats

- JPEG
- GIF
- PNG
- SVG
- WebP

JPEG

JPEG (Joint Photographic Experts Group) or JPG is a 24-bit, true-color file format that is ideal for photographic images.

It uses a compression scheme to reduce the file size (and hence download time) of the image

JPEG is quite poor for vector art or diagrams or any image with a large area of a single color

At the highest levels of compression, you will begin to see blotches and noise (also referred to as **artifacts**)

JPEG example compression



Original = 931 K



JPG Quality 100 = 335 K



JPG Quality 60 = 136 K



JPG Quality 30 = 77 K



JPG Quality 10 = 52 K

JPEG and art work



original



Saved as jpg

Notice the noise and artifacts!

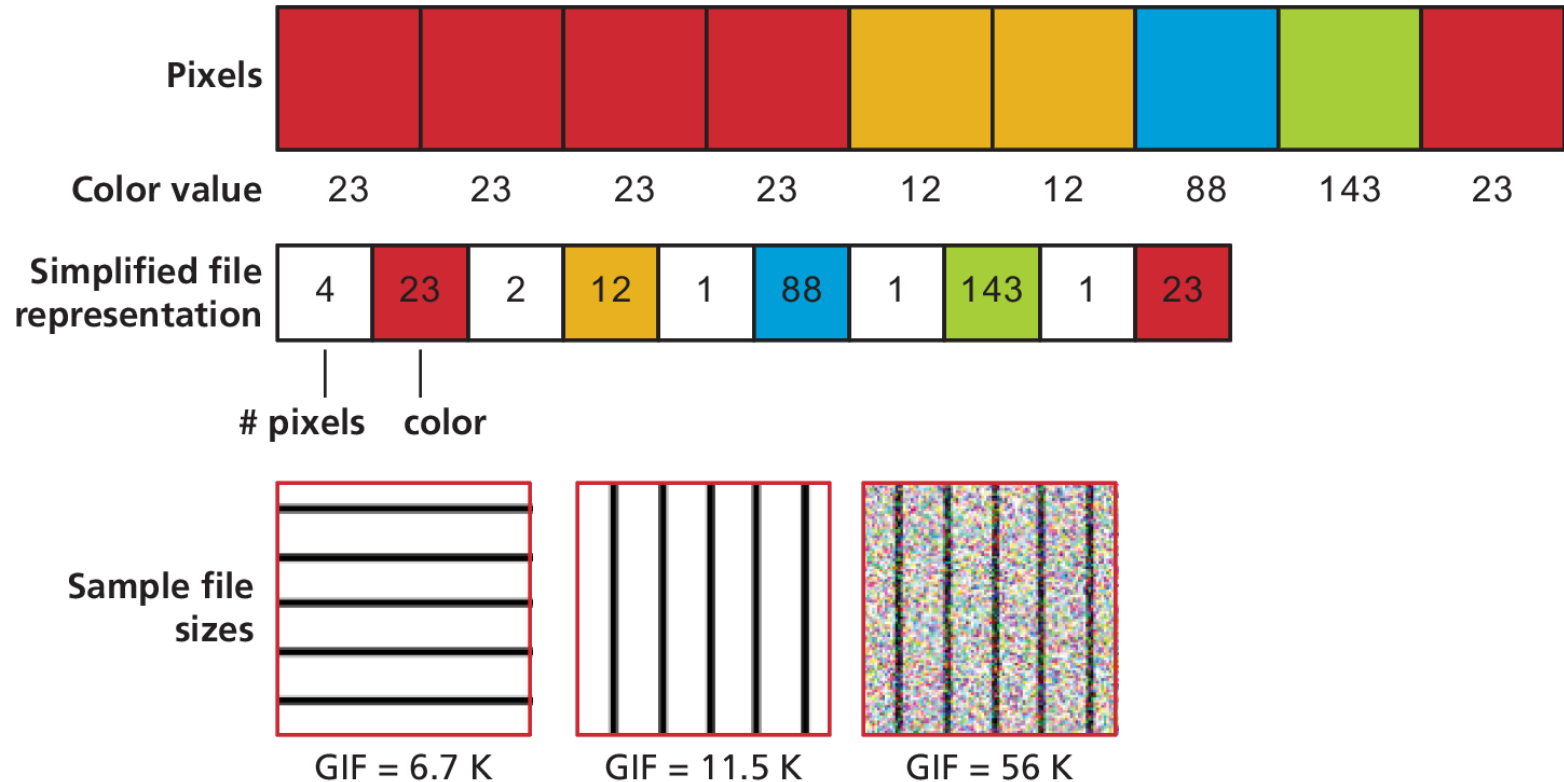
GIF

The **GIF** (Graphic Interchange Format) file was the first image format supported by the earliest web browsers.

GIF is an 8-bit or less format, meaning that it can contain no more than 256 colors!

GIF uses a lossless compression system, which means that no pixel information is lost. The compression system, is called **runlength compression** (also called **LZW compression**).

Gif Run-length compression

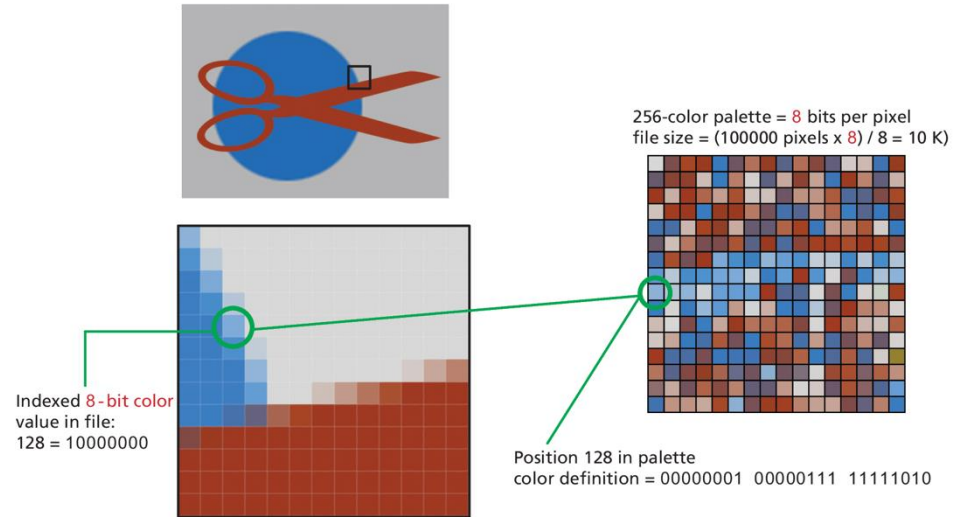


Color Palette

The GIF file format uses indexed color

8 bits (or fewer) dedicated to each color pixel in the image.

Those 8 bits for each pixel reference (or index) a color that is described in a **color palette** (also called a color table or color map)

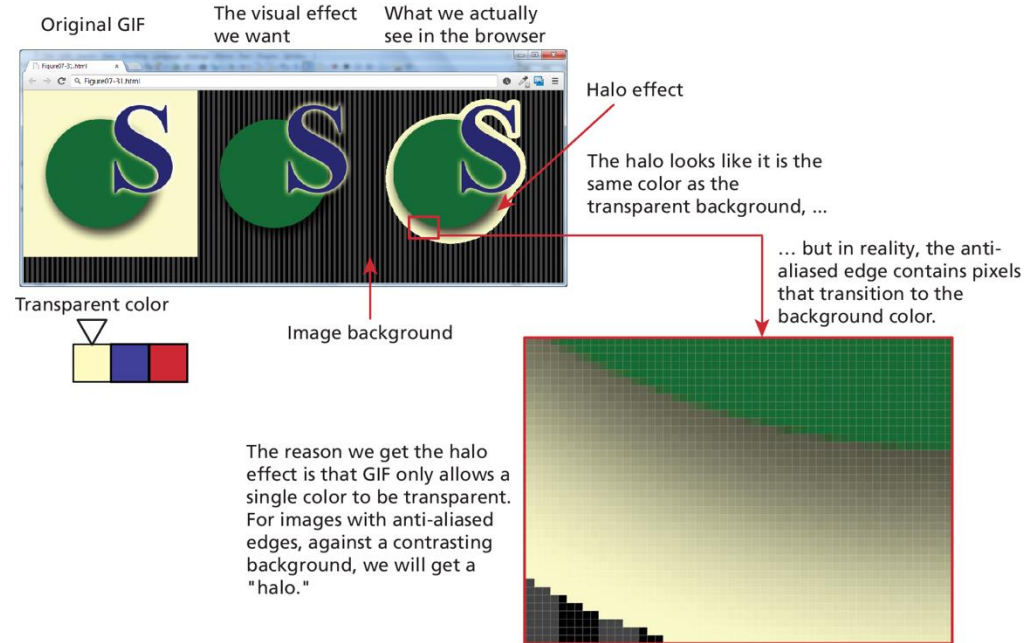


Transparency in Gif

GIF has 1-bit transparency.

A pixel is either fully transparent or fully opaque

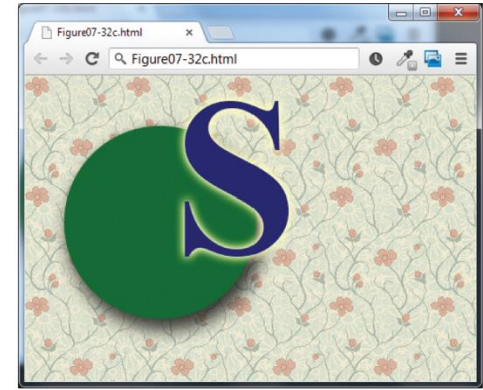
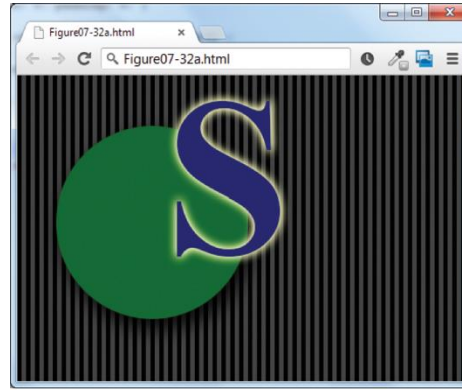
Anti-aliasing “smoothing” along boundary edges often result in a “halo” of color. Formats like png can address this issue.



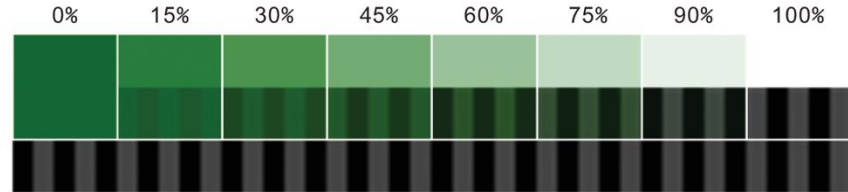
PNG Format

The **PNG** (Portable Network Graphics)

- Lossless compression
- 8-bit (or 1-bit, 2-bit, and 4-bit) indexed color as well as full 24-bit true color
- From 1 to 8 bits of **transparency**.



PNG format with 256 levels of transparency



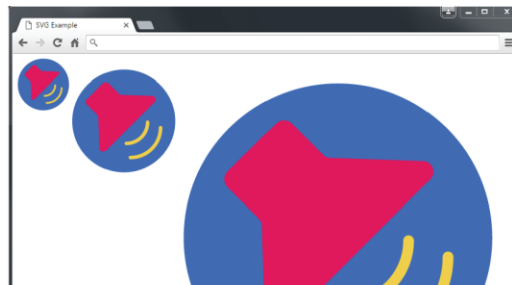
Transition showing six levels of transparency

SVG Format

SVG (Scalable Vector Graphics) is a *vector* format

Like all vector formats, SVG graphics do not lose quality when enlarged or reduced.

The files are actually XML files



```



```

Because SVG is a vector format, there is no loss of quality when it is resized

```
<?xml version="1.0" encoding="utf-8"?>
<svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
  viewBox="0 0 95 94" style="enable-background:new 0 0 95 94;" xml:space="preserve">
<style type="text/css">
  .st0{fill:#366BC9;}.st1{fill:#E0105B;}.st2{fill:#EFC44A;}
</style>
<path class="st0" d="M92.7,46.9c0,25.1-20.4,45.5-45.5,45.5C22.1,92.4,1.7,72.1,7.4,46.9c0-25.1,20.4-45.5,45.5-45.5
  C72.3,1.4,92.7,21.8,92.7,46.9L92.7,46.9z M92.7,46.9"/>
<path class="st1" d="M42.8,22.5l-9.2-9.2c-1.3-1.3-3.4-1.3-4.7,0L14.7,27.4c-1.3,1.3-1.3,3.4,0,4.7l9.2,9.2c0.4,0.4,0.7,0.9,1.5
  11,28.6c0.6,2.5,3.7,3.3,5.5,1.5l43-43c1.8-1.8,1.4-9.1,5.5-51-28.6-1C43.7,23.2,43.2,22.9,42.8,22.5L42.8,22.5"/>
<path class="st2" d="M51.7,80.3c-0.3-0.3-0.5-0.7-0.5-1.1c0-0.9,0.7-1.6,1.6-1.6c66.7,77.7,78,66.4,78,52.6c0-0.9,0.7-1.6,1.6-1.6
  c0.9,0.1,6.0,7.1,6.1,6c0,15.6-12.7,28.2-28.2,28.2C52.4,80.8,52,80.6,51.7,80.3L51.7,80.3z M51.7,80.3"/>
<path class="st2" d="M48.1,67.8c-0.3-0.3-0.5-0.7-0.5-1.1c0-0.9,0.7-1.6,1.6-1.6c9.5,0,17.3-7.8,17.3-17.3c0-0.9,0.7-1.6,1.6-1.6
  c0.9,0.1,6.0,7.1,6.1,6c0,11.3-9.2,20.4-20.4,20.4C48.8,68.2,48.4,68,48.1,67.8L48.1,67.8z M48.1,67.8"/>
</svg>
```

SVG is compressed XML

Other Image Formats

The **TIF** (Tagged Image File) format is a cross-platform lossless image format especially useful to print professionals.

WebP is a new image file format promoted by Google. It supports *both* lossy and lossless compression, and Google claims WebP compression results are superior in comparison to JPG or PNG formats. Lossless WebP also supports transparency. At the time of writing, however, Safari on iOS does not support this format.

ICO is another web file format (.ico) whose sole use is for **favicon** (short for favorite icon) images.

Audio and Video

Until HTML5, adding audio or video to a web page typically required making use of additional plug-ins, often Adobe Flash (Chapter 8).

With HTML5 you can add these media features in HTML without the involvement of any plug-in.

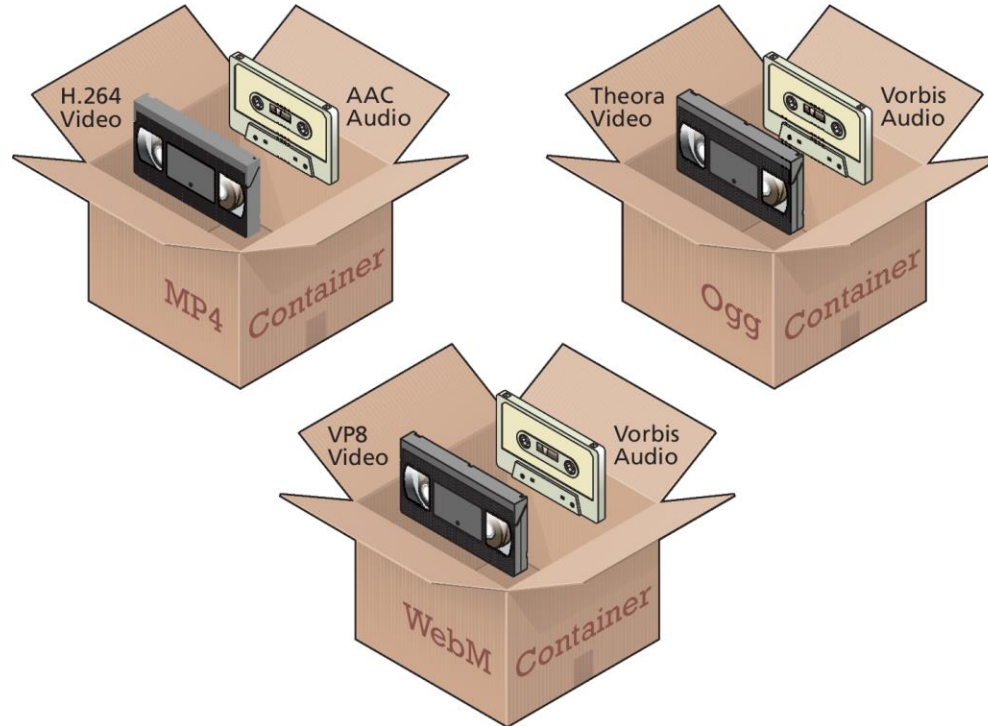
- Unfortunately, the browsers do not support the same list of media formats, so browser incompatibilities are still a problem with audio and video.

Media Concepts

There are a *lot* of different audio and video formats. While we won't cover them all, two concepts are essential to understanding media formats

- **Media encoding** – Media is compressed/and encoded using a codec
 - **Codec** (for **compression/decompression**) software used to encode the media. There are literally thousands of codecs.
- **Container formats** A container is similar in concept to ZIP files: both are compressed file formats that contain other content. There is a large number of container formats.

Media encoding and containers

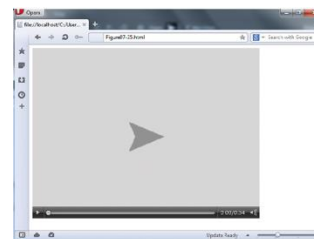


Browser Video Support

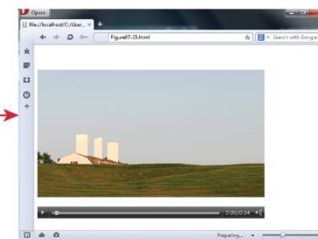
- **MP4 container with H.264 Video and AAC Audio.** This combination is generally referred to as **MPEG-4** and has the **.mp4** or **.m4v** file extension. H.264 is a proprietary codec and the browser manufacturer must pay a licensing fee to decode it
- **WebM container with VP8 video and Vorbis audio.** Files using this combination usually have the **.webm** file extension. This combination was created by Google to be open-source and royalty free.
- **Ogg container with Theora video and Vorbis audio.** Like the previous combination, this one is open-source and royalty free. Files using this combination usually have the **.ogv** file extension.

Using the <video> element

Due to varying browser support, you will need to serve more than one type. Thankfully, HTML5 makes this a reasonably painless procedure.



Showing poster image before playback



After playback begins (Opera)

```
<video id="video" poster="preview.png" controls width="480" height="360">
  <source src="sample.mp4" type="video/mp4; codecs=avc1.42E01E, mp4a.40.2"/>
  <source src="sample.webm" type="video/webm; codecs=vp8, vorbis"/>
  <source src="sample.ogv" type="video/ogg; codecs=theora, vorbis"/>

  <!-- Use Flash if above video formats not supported -->
  <object width="480" height="360" type="application/x-shockwaveflash" data="sample.swf">
    <param name="movie" value="sample.swf">
    <param name="flashvars" value="controlbar=over&image=preview.png&file=sample.mp4">
    
  </object>
</video>
```



Chrome



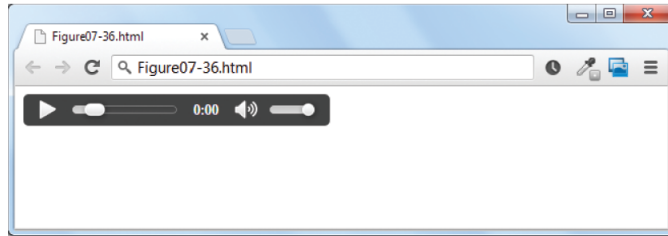
Firefox

Browser Audio Support

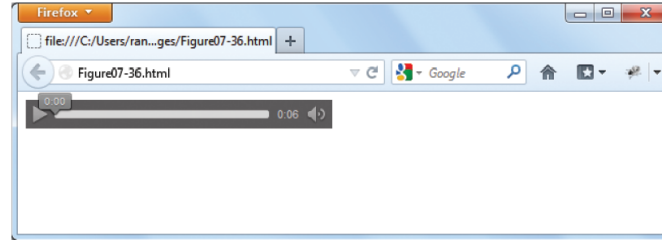
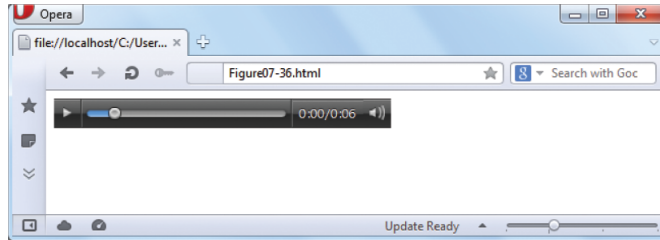
- **MP3.** Both a container format and a codec. It is patented and requires browser manufacturers to pay licensing fees. Usually has the **.mp3** file extension.
- **WAV.** Also a container and a codec. Usually has the **.wav** file extension.
- **OGG.** Container with Vorbis audio. Open-source. Usually has the **.ogg** file extension.
- **Web.** Container with Vorbis audio. Open-source. Usually has the **.webm** file extension.
- **MP4.** Container with AAC audio. Also requires licensing. Usually has the **.m4a** file extension.

Using the <audio> element

As with video, if you intend to provide audio in your pages, you will need to serve more than one type.



```
<audio id="example" controls preload="auto">  
  <source src="example.ogg" type="audio/ogg">  
  <source src="example.wav" type="audio/webm">  
  <source src="example.webm" type="audio/webm">  
  <p>Browser doesn't support the audio control</p>  
</audio>
```



Working with Color

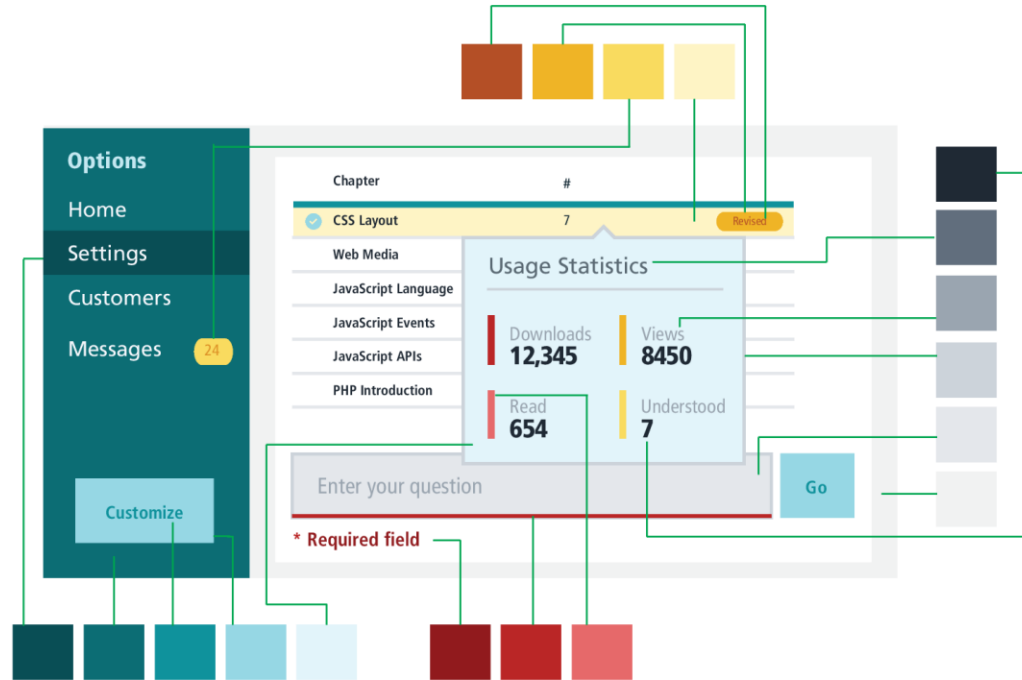
If you are learning web development within a program that focuses on design, you will no doubt find (or have found) yourself spending a great deal of time learning about color relationships and color psychology.

If your program that focuses mainly on programming, will likely need to learn those concepts yourself.

If you are not completely confident in your ability to pick harmonious color combinations, there is a variety of online tools such as paletton.com, colordesigntool.com, and colormind.io

Most web user interfaces typically need six or seven variations (shades) of three or four colors

Practical color in web interfaces



This page just has four basic colors but uses multiple variations of all four.

Using color shades with CSS

```
/* Define primary colors via CSS variables, using hsl or  
hex. By convention, numbers 100, 200, etc indicate  
shades */
```

```
:root {  
  --color-primary-100: hsl(184,88%, 94%);  
  --color-primary-200: #87EAF2;  
  --color-primary-300: #38BEC9;  
  --color-primary-400: #14919B;  
  --color-primary-500: #0A6C74;  
}
```

```
/* Use variables where needed */
```

```
header {  
  background-color: var(--color-primary-500);  
  color: var(--color-primary-100);  
}
```

```
/* Alternately, define colors in utility classes */
```

```
.bg-primary-100 {  
  background-color: #E0FCFF;  
}  
.bg-primary-500 {  
  background-color: #0A6C74;  
}  
.text-primary-100 {  
  color: #E0FCFF;  
}
```

```
/* Switch to HTML to show how to use utility color  
classes */
```

```
<article class="bg-primary-500 text-primary-100">
```

LISTING 6.1 Using color shades with CSS

Key Terms

additive colors	container formats	halftones	LZW compression	run-length compression
alpha transparency	device pixels	HSL color model	media encoding	saturation
anti-aliasing	digital representation	hue	MPEG-4	subtractive colors
artifacts	display resolution	image size	opacity	SVG
bitmap image	dithering	interpolate	pixels	TIF
CMYK color model	favicon	JPEG	PNG	vector image
codec	gamut	lightness	raster image	WebP
color depth	GIF	lossless compression	reference pixel	web-safe color palette
color palette	gradient	lossy compression	RGB color model	

Copyright



This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.