

# HTML/CSS

HTML review and extra

CSS review

CSS Color

# ID

- Every HTML element can carry the id attribute
- ID is used to uniquely identify that element from other elements on the page
- Can be used to allow the script to work with that particular element
- Global attribute - it can be used on any element
- ID value should start with a letter or an underscore (not a number or any other character)
- No two elements on the same page can have the same value for their id attributes
- It allows you to style it differently than any other instance of the same element on the page

<p> The practice of design is a responsibility and a privilege. It is an independent pursuit that precipitates collective action.</p>

<p id="pullquote"> Design advances the interplay between the exalted and the everyday, and stimulates an interchange between diverse disciplines. </p>

<p> Serving as both unifying force and catalyst for change, the designer plays a pivotal role in the advocacy of ideas.</p>

The practice of design is a responsibility and a privilege. It is an independent pursuit that precipitates collective action.

DESIGN ADVANCES THE INTERPLAY BETWEEN THE EXALTED AND THE EVERYDAY, AND STIMULATES AN INTERCHANGE BETWEEN DIVERSE DISCIPLINES.

Serving as both unifying force and catalyst for change, the designer plays a pivotal role in the advocacy of ideas.

# Class

- Every HTML element can carry a class attribute
- Class is used to identify several elements as being different from the other elements on the page.
- Class value should describe the class it belongs to
- The class attribute on any element can share the same value

# Class

<p class="important"> Associate Professor of Graphic Design, Sharon Oiga, organized Resind/Fast-Forward, a weekend of design inspiration hosted by the Society of Typographic Arts. </p>

<p> The informal event addresses issues and practices of design and typography that have or have not changed and discusses the why, why not, and how. </p>

<p class="important"> Friday, January 24– Sunday, January 26 </p>

ASSOCIATE PROFESSOR OF GRAPHIC DESIGN, SHARON OIGA, ORGANIZED RESIND/FAST-FORWARD, A WEEKEND OF DESIGN INSPIRATION HOSTED BY THE SOCIETY OF TYPOGRAPHIC ARTS.

The informal event addresses issues and practices of design and typography that have or have not changed and discusses the why, why not, and how.

FRIDAY, JANUARY 24–SUNDAY, JANUARY 26

# Block Elements

Always appear to start on a new line in the browser window:

- `<h1>`
- `<p>`
- `<ul>`
- `<li>`

```
<h1>Hiroshi Sugimoto</h1>
```

```
<p>The dates for the ORIGIN OF ART exhibition are as follows:</p>
```

```
<ul>
```

```
<li>Science: 21 Nov - 20 Feb 2010/11</li>
```

```
<li>Architecture: 6 Mar - 15 May 2011</li>
```

```
<li>History: 29 May - 21 Aug 2011</li>
```

```
<li>Religion: 28 Aug - 6 Nov 2011</li>
```

```
</ul>
```

RESULT

## Hiroshi Sugimoto

The dates for the ORIGIN OF ART exhibition are as follows:

- Science: 21 Nov - 20 Feb 2010/11
- Architecture: 6 Mar - 15 May 2011
- History: 29 May - 21 Aug 2011
- Religion: 28 Aug - 6 Nov 2011

# Inline Elements

Always appear to continue on the same line as the previous elements:

- `<a>`
- `<b>`
- `<img>`
- `<em>`

Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this `<em>Origins of Art</em>` cycle is organized around four themes: `<b>science, architecture, history</b>` and `<b>religion</b>`.

## RESULT

Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this *Origins of Art* cycle is organized around four themes: **science, architecture, history** and **religion**.

## <div>

- <div> element allows to group a set of elements together in one block-level box
- the contents of the <div> element will start on a new line
- Using an id or class attribute on the <div> element allows to create CSS style rules to indicate how much space the <div> element should occupy on the screen and change the appearance of all the elements contained within it



<div>

```
<div id="header">
```

```

```

```
<ul>
```

```
<li><a href="schools.html">Schools</a></li>
```

```
<li><a href="partners.html">Partners</a></li>
```

```
<li><a href="about.html">About</a></li>
```

```
</ul>
```

```
</div><!-- end of header ->
```

---

## ArchDesigArts at Chicago

---

Schools

Partners

About

# <span>

- an inline equivalent of the <div> element. It is used to:
  - Contain a section of text where there is no other suitable element to differentiate it from its surrounding text
  - Contain a number of inline elements
  - To control the appearance of the content of these elements using CSS

usually a class or id attribute is used with <span> elements

## <span>

<p> Graphic Design can be described as the process of visually communicating ideas and information through the use of signs, symbols, and images. </p>

Graphic Design can be described as the process of visually communicating ideas and information through the use of signs, symbols, and images.

# <iframe>

- Iframe= inline frame
- a little window in which you can see another page

## Embed Google Maps

The content of the iframe can be any html page (either located on the same server or anywhere else on the web)

# <iframe>

## <iframe> attributes

### src

The src attribute specifies the URL of the page to show in the frame

### height

The height attribute specifies the height of the iframe in pixels

### width

The width attribute specifies the width of the iframe in pixels

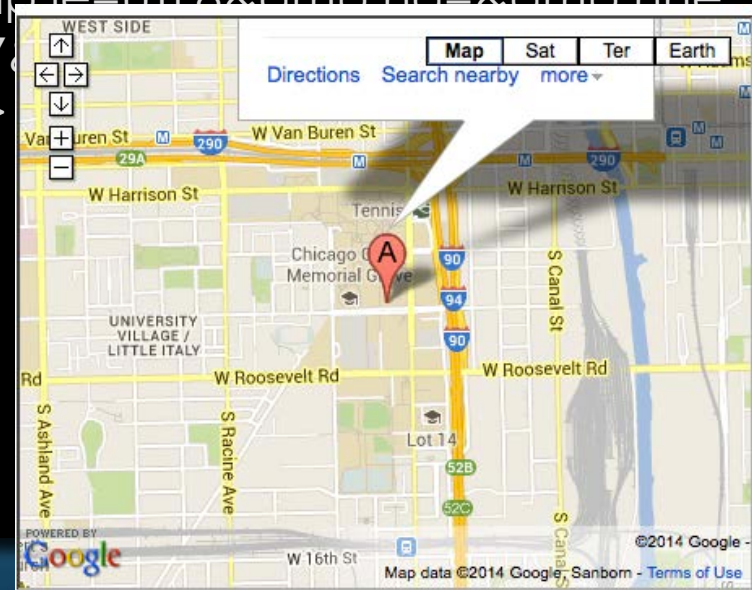
# <iframe>

```
<iframe width="450" height="350"
```

```
src="https://maps.google.com/maps?f=q&source=s_q&hl=en&am  
mp;geocode=&q=842+W+taylor+street,+chicago+il&aq=&sl  
=41.833733,-
```

```
87.731964&ssp=0.714204,1.454315&ie=UTF8&hq=&hne  
ar=842+W+Taylor+St,+Chicago,+Illinois+60607  
41.869904,-87.648308&output=embed">
```

```
</iframe>
```



# <iframe>

## scrolling

Not supported in HTML5

indicates whether the iframe should have scrollbars or not.

Scrollbars allow the user to move around the frame to see more content.

values:

**yes** (to show scrollbars)

**no** (to hide scrollbars)

**auto** (to show them only if needed)

# <iframe>

## frameborder

Not supported in HTML5

indicates whether the frame should have a border or not  
values:

0 indicates that no border should be shown

1 indicates that a border should be shown

## seamless

New in HTML5

Older browsers do not support

can be applied to an iframe where scrollbars are not desired

The seamless attribute does not need a value, but you will often see authors give it a value of seamless



# <meta>

- inside the <head> element
- contains information about the web page (described in attributes)
- not visible to users
- tells search engines about web page, who created it, and whether or not it is time sensitive (it can be set to expire)
- the <meta> tag does not have a closing tag

# <meta>

## name

The value of the name attribute can be anything you want it to be. Some defined values for this attribute that are commonly used are:

### description

contains a description of the page. This description is commonly used by search engines to understand what the page is about and should be a maximum of 155 characters. Sometimes it is also displayed in search engine results.

### keywords

contains a list of comma-separated words that a user might search on to find the page. In practice, this no longer has any noticeable effect on how search engines index your site.

# <meta>

## robots

indicates whether search engines should add this page to their search results or not.

values:

**noindex** can be used if this page should not be added

**nofollow** can be used if search engines should add this page in their results but not any pages that it links to

## author

defines the author of the web page

## pragma

prevents the browser from caching the page. (That is, storing it locally to save time downloading it on subsequent visits.)

# <meta>

## expires

Because browsers often cache the content of a page, the expires option can be used to indicate when the page should expire (and no longer be cached).

# <meta>

```
<!DOCTYPE html>
<html>
<head>
<title>Information About Your Pages</title>
<meta name="description"
content="Intro to CSS" />
<meta name="keywords" content="introduction, styling, design" />
<meta name="robots" content="nofollow" />
<meta http-equiv="author" content="SOD" />
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="expires"
content="Fri, 31 Dec 2014 23:59:59 GMT" />
</head>
<body>
</body>
</html>
```

# Escape characters

[http://www.w3schools.com/tags/ref\\_entities.asp](http://www.w3schools.com/tags/ref_entities.asp)

<

**Less-than sign**  
&lt;  
&#60;

¢

**Cent sign**  
&cent;  
&#162;

‘

**Left single quote**  
&lsquo;  
&#8216;

>

**Greater-than sign**  
&gt;  
&amp;

£

**Pound sign**  
&pound;  
&#163;

’

**Right single quote**  
&rsquo;  
&#8217;

&

**Ampersand**  
&amp;  
&#38;

¥

**Yen sign**  
&yen;  
&#165;

“

**Left double quotes**  
&lquo;  
&#8220;

”

**Quotation mark**  
&quot;  
&#34;

€

**Euro sign**  
&euro;  
&#8364;

”

**Right double quotes**  
&rdquo;  
&#8221;

©

**Copyright symbol**  
&copy;  
&#169;

×

**Multiplication sign**  
&times;  
&#215;

®

**Registered trademark**  
&reg;  
&#174;

÷

**Division sign**  
&divide;  
&#247;

™

**Trademark**  
&trade;  
&#8482;

# Cascading Style Sheets (CSS)

CSS is a simple mechanism for adding style (e.g. fonts, colors, spacing) to web documents

Example styles

Text

Typeface

Size

Color

Italics, bold, upper-case, etc.

Boxes

Width and height

Borders

Images, Position in the browser window, etc.

# Cascading Style Sheets (CSS)

- designed for HTML in 1996
- has its own syntax
- CSS1 (1996), CSS2 (1998), CSS Mobile Profile
- CSS3 (working drafts)
- browser support weaker for newer versions
- CSS1 IE5 onwards, Firefox, etc.
- CSS level 2 IE8, Firefox, Opera
- [http://www.w3schools.com/css/css\\_reference.asp](http://www.w3schools.com/css/css_reference.asp)



# CSS+HTML

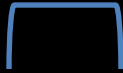
## CSS rules

- each rule starts with an element name
- followed by a list of style properties bracketed by { and }
- element name defines where it is applied
- each style property starts with the property's name, then a colon and lastly the value for this property. When there is more than one style property in the list, you need to use a semicolon between each of them

# CSS + HTML

CSS works by associating rules with HTML elements. These rules govern how the content of specified elements should be displayed. A CSS rule contains two parts: a **selector** and a **declaration**.

Selector



p {

font-family: Arial; }



Declaration

# CSS+HTML

**Selectors** indicate which element the rule applies to. The same rule can apply to more than one element if you separate the element names with commas.

**Declarations** indicate how the elements referred to in the selector should be styled. Declarations are split into two parts (a property and a value), and are separated by a colon.

Selector



p {

font-family: Arial; }




Declaration

# CSS+HTML

CSS declarations sit inside curly brackets and each is made up of two parts: a **property** and a **value**, separated by a colon.

You can specify several properties in one declaration, each separated by a semi-colon.

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow; }  
      
    Property Value
```

# CSS+HTML

**Properties** indicate the aspects of the element you want to change. Ex.: font, width, height and border.

**Values** specify the settings you want to use for the chosen properties. Ex.: color, etc.

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow; }  
    [ ] [ ]  
Property Value
```

# CSS+HTML – example lab3

```
<!DOCTYPE html>
<html>
<head>
<title>Introducing CSS</title>
<style type="text/css">
body {
font-family: Arial, Verdana, sans-serif;}
h1, h2 {
color: #ee3e80;}
p {
color: #665544;}
</style>
</head>
<body>
<h1>From Structure to Style</h1>
<p>CSS Cascading Style Sheets CSS allows to create rules to specify how the content of an
element should be presented.
</p>
<h2>Selector</h2>
<p>Selectors indicate which element the rule applies to. The same rule can apply to more than
```

# External CSS

## <link>

- can be used in an HTML document to tell the browser where to find the CSS file
- inside the <head> element

Attributes:

### href

specifies the path to the CSS file (which is often placed in a folder called css or styles).

### type

specifies the type of document being linked to. The value should be text/css.

### rel

specifies the relationship between the HTML page and the file it is linked to. The value should be stylesheet when linking to a CSS file.

# External CSS

```
<link href="css/styles.css" type="text/css" rel="stylesheet" />
```



# External CSS

All web pages can share the same style sheet

use the <link> element on each HTML page to link to the same CSS document

the same code does not need to be repeated in every page

less code and smaller HTML pages

once the user has downloaded the CSS stylesheet, the rest of the site will load faster

Making changes is easy in a single CSS file – all web pages will be updated automatically

HTML code is easier to read

# Internal CSS

```
<style type="text/css">
```

```
.....
```

```
</style>
```

- includes CSS rules within an HTML page by placing them inside a `<style>` element
- inside the `<head>` element of the page

The `<style>` element should use the `type` attribute to indicate that the styles are specified in CSS. The value should be `text/ css`.

# Internal CSS

When building a site with more than one page, you should use an external CSS style sheet. This:

- Allows all pages to use the same style rules (rather than repeating them in each page).
- Keeps the content separate from how the page looks.
- Means you can change the styles used across all pages by altering just one file (rather than each individual page).

# Inline CSS

`<p style = "font-size: 20pt">` This text has the font-size style applied to it, making it 20pt. `</p>`

`<p style = "font-size: 20pt; color: #0000ff">` This text has the font-size and color styles applied to it, Making it 20pt and blue. `</p>`

# Mobile CSS

reduce http connections

(expensive – cost time to initiate and fetch)

- Internal CSS styles
- Groups of adjacent images  
(one image rather than several)

# Selectors

SELECTOR	MEANING	EXAMPLE
UNIVERSAL SELECTOR	Applies to all elements in the document	<code>* {}</code> Targets all elements on the page
TYPE SELECTOR	Matches element names	<code>h1, h2, h3 {}</code> Targets the <code>&lt;h1&gt;</code> , <code>&lt;h2&gt;</code> and <code>&lt;h3&gt;</code> elements
CLASS SELECTOR	Matches an element whose <code>class</code> attribute has a value that matches the one specified after the period (or full stop) symbol	<code>.note {}</code> Targets any element whose <code>class</code> attribute has a value of <code>note</code> <code>p.note {}</code> Targets only <code>&lt;p&gt;</code> elements whose <code>class</code> attribute has a value of <code>note</code>
ID SELECTOR	Matches an element whose <code>id</code> attribute has a value that matches the one specified after the pound or hash symbol	<code>#introduction {}</code> Targets the element whose <code>id</code> attribute has a value of <code>introduction</code>

# Selectors

## CHILD SELECTOR

Matches an element that is a direct child of another

```
l1>a {}
```

Targets any <a> elements that are children of an <l1> element (but not other <a> elements in the page)

---

## DESCENDANT SELECTOR

Matches an element that is a descendent of another specified element (not just a direct child of that element)

```
p a {}
```

Targets any <a> elements that sit inside a <p> element, even if there are other elements nested between them

---

## ADJACENT SIBLING SELECTOR

Matches an element that is the next sibling of another

```
h1+p {}
```

Targets the first <p> element after any <h1> element (but not other <p> elements)

---

## GENERAL SIBLING SELECTOR

Matches an element that is a sibling of another, although it does not have to be the directly preceding element

```
h1~p {}
```

If you had two <p> elements that are siblings of an <h1> element, this rule would apply to both

# How CSS Rules Cascade

If there are two or more rules that apply to the same element, it is important to understand which will take precedence.

-writing simpler style sheets to create generic rules that apply to most elements and then override the properties on individual elements that need to appear differently

## Last Rule

If the two selectors are identical, the latter of the two will take precedence. Here you can see the second `i` selector takes precedence over the first.



# How CSS Rules Cascade

## Specificity

If one selector is more specific than the others, the more specific rule will take precedence over more general ones.

## !important

You can add **!important** after any property value to indicate that it should be considered more important than other rules that apply to the same element.

# How CSS Rules Cascade

## Example: CSS2

h1 is more specific than \*

p b is more specific than p

p#intro is more specific than p

# Inheritance

If you specify the font-family or color properties on the <body> element, they will apply to most child elements

saves time and results in simpler style sheets

background-color or border properties are **not** inherited by child elements

To force a lot of properties to inherit values from their parent elements use **inherit** for the value of the properties.

# Inheritance

```
body {  
    font-family: Arial, Verdana, sans-serif;  
    color: #665544;  
    padding: 10px;}  
.page {  
    border: 1px solid #665544;  
    background-color: #efefef;  
    padding: inherit; }
```

# Color

## RGB values

These express colors in terms of how much red, green and blue are used to make it up

```
rgb(100,100,90)
```

## hex Codes

These are six-digit codes that represent the amount of red, green and blue in a color, preceded by a pound or hash # sign

```
#ee3e80
```

## Color names

There are 147 predefined color names that are recognized by browsers

```
DarkCyan
```

## Rgba (opacity CSS3)

Adds alpha for transparency support

# Foreground Color

```
/* color name */  
h1 {  
    color: DarkCyan;}  
/* hex code */  
h2 {  
    color: #ee3e80;}  
/* rgb value */  
p {  
    color: rgb(100,100,90);}
```

# Background Color

background-color property sets the color of the background for a “box” of each HTML element

```
body {  
    background-color: rgb(200,200,200);}  
  
h1 {  
    background-color: DarkCyan;}  
  
h2 {  
    background-color: #ee3e80;}  
  
p {  
    background-color: white;}
```

# Background Color

background-color property sets the color of the background for a “box” of each HTML element

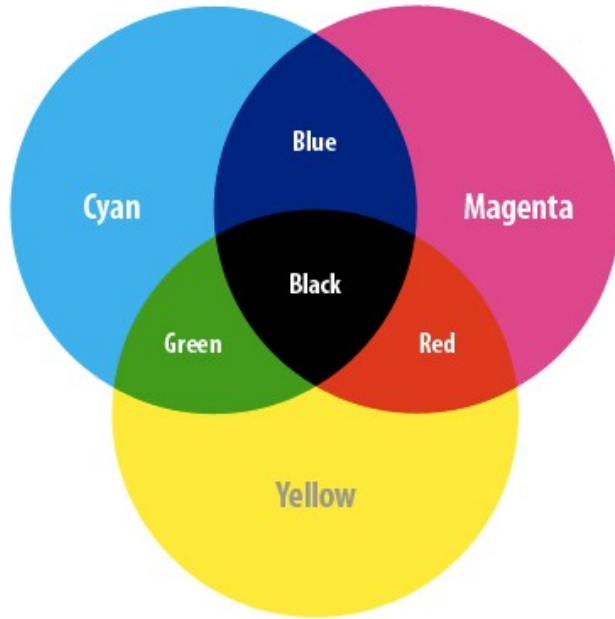
Unspecified color is transparent (not white)

```
body {  
    background-color: rgb(200,200,200);}
h1 {  
    background-color: DarkCyan;}
h2 {  
    background-color: #ee3e80;}
p {  
    background-color: white;}
```



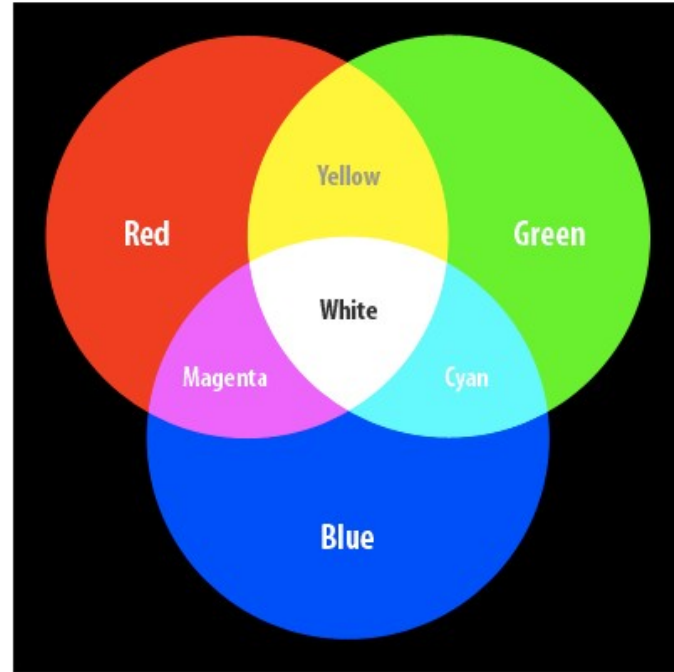
# Color Systems

## CMYK – The Subtractive System



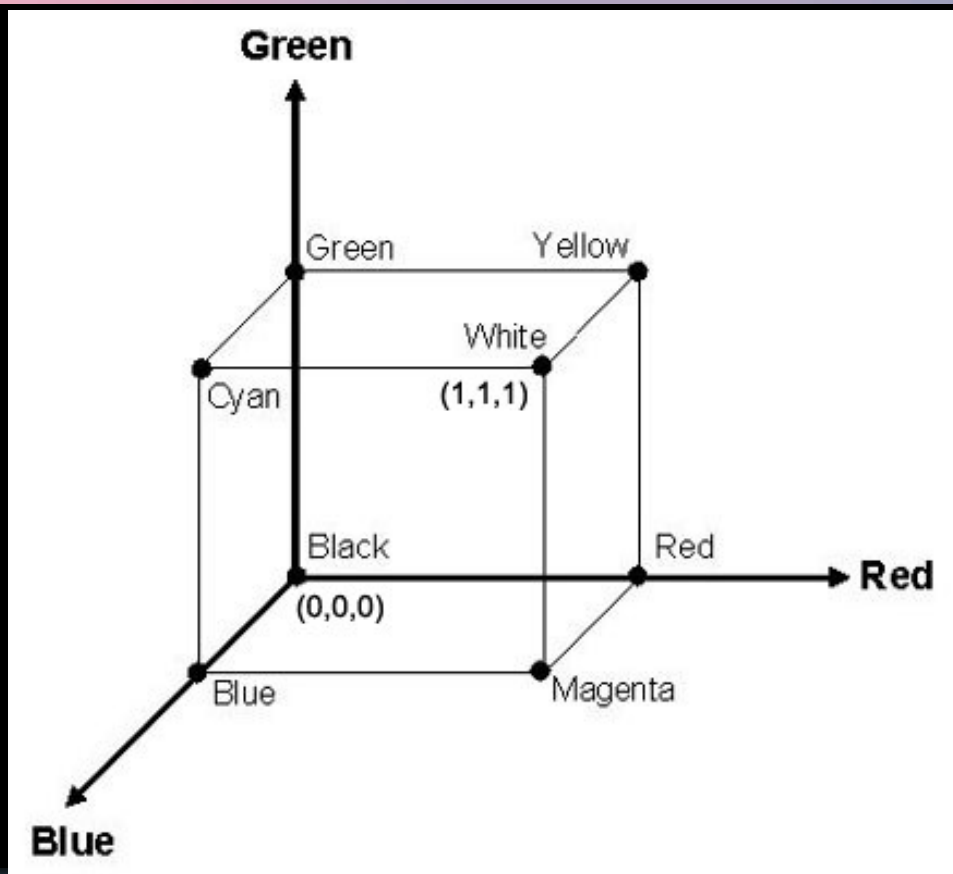
*CMYK Color Model*

## RGB – The Additive System

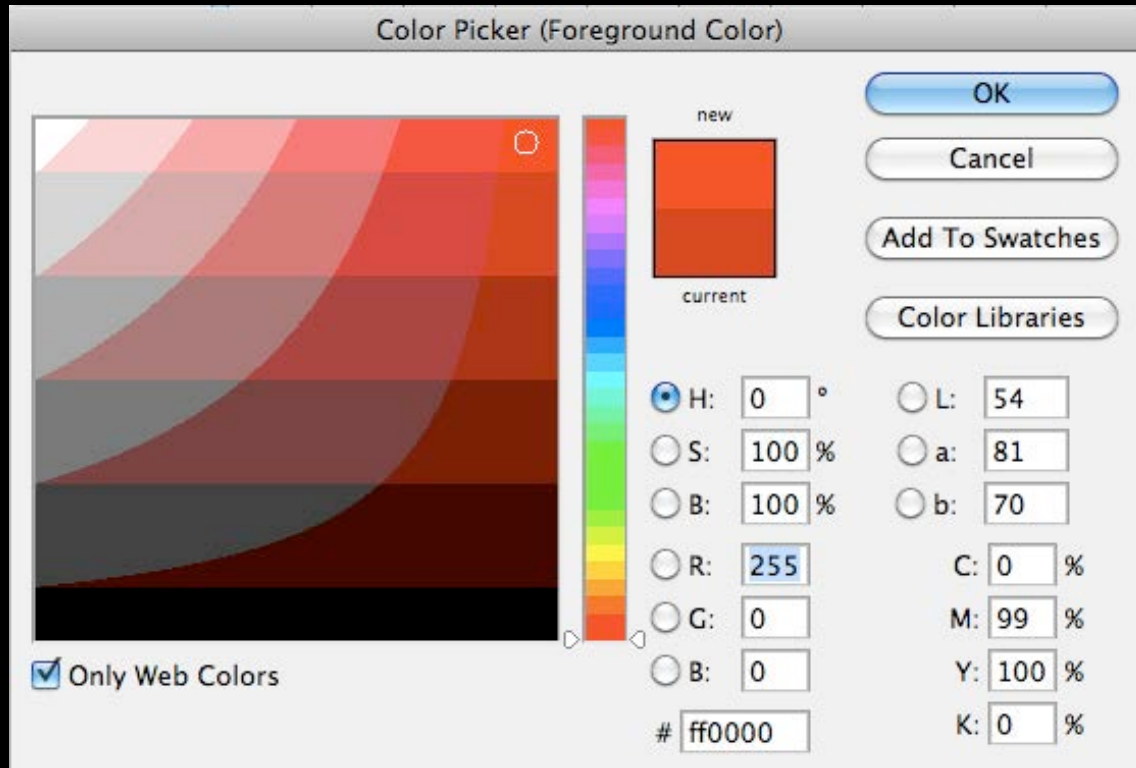


*RGB Color Model*

# Color Cube



# Color Picker (in Photoshop)



# RGB Color

Values for red, green, and blue are expressed as numbers between 0 and 255.

```
rgb(255,0,0)
```

This red color is made up of the following values:

255 red

0 green

0 blue

# Hexadecimal Color

In mathematics and computer science, **hexadecimal** (also base 16, or **hex**) is a positional numeral system with a radix, or base, of 16.

It uses sixteen distinct symbols, most often the symbols 0–9 to represent values zero to nine, and A, B, C, D, E, F (or alternatively a–f) to represent values ten to fifteen.

Hexadecimal numbers are used on web pages to specify colors.

<b>Decimal:</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Hexadecimal:</b>	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

No intensity

Full intensity

# Hexadecimal Color

The color is defined by its mix of Red, Green and Blue intensities, each of which can be in the range:

- 0 to 255 (in decimal)
- 00 to FF (in hexadecimal)

HEX # rrggbb

BLACK #000000

WHITE #FFFFFF



# Hexadecimal Color

HEX # rrggbb

red #FF0000

green #00FF00

blue #0000FF

- The first two digits are the red color blend, from no intensity ( 00 )to full intensity red ( FF )
- The second two digits are the green color blend, from no intensity ( 00 )to full intensity red ( FF )
- The third two digits are the blue color blend, from no intensity ( 00 )to full intensity red ( FF ) □

# Hexadecimal Color

HEX # rrggbb

magenta # ?

cyan # ?

pink # ?

[What the HEX?](#)

<b>Decimal:</b>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>Hexadecimal:</b>	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

No intensity

Full intensity



# Opacity (CSS3)

opacity allows you to specify the opacity of an element and any of its child elements (supported by the most recent browsers )

it is a number between 0.0 and 1.0

0.5 is 50% opacity

```
p.one {  
  background-color: rgb(0,0,0);  
  opacity: 0.5;}  
  
p.two {  
  background-color: rgb(0,0,0);  
  background-color: rgba(0,0,0,0.5);}
```

# HSL (CSS3)

## Hue Saturation Brightness

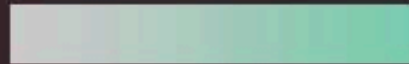
### HUE

Hue is the colloquial idea of color. In HSL colors, hue is often represented as a color circle where the angle represents the color, although it may also be shown as a slider with values from 0 to 360.



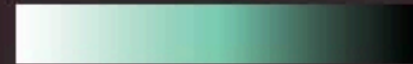
### SATURATION

Saturation is the amount of gray in a color. Saturation is represented as a percentage. 100% is full saturation and 0% is a shade of gray.



### LIGHTNESS

Lightness is the amount of white (lightness) or black (darkness) in a color. Lightness is represented as a percentage. 0% lightness is black, 100% lightness is white, and 50% lightness is normal. Lightness is sometimes referred to as *luminosity*.



# HSLA (CSS3)

Hue Saturation Brightness + Alpha = hsla

Alpha is expressed as a number between 0 and 1.0.

0.5 - 50% transparency

0.75 - 75% transparency

```
body {
```

```
    background-color: #C8C8C8;
```

```
    background-color: hsl(0,0%,78%);}
```

```
p {
```

```
    background-color: #ffffff;
```

```
    background-color: hsla(0,100%,100%,0.5);}
```

# HSLA (CSS3)

Hue Saturation Brightness + Alpha = hsla

Alpha is expressed as a number between 0 and 1.0.

0.5 - 50% transparency

0.75 - 75% transparency

```
body {
```

```
    background-color: #C8C8C8;
```

```
    background-color: hsl(0,0%,78%);}
```

```
p {
```

```
    background-color: #ffffff;
```

```
    background-color: hsla(0,100%,100%,0.5);}
```