


CSS

CSS for Styling

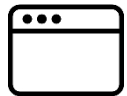
In this lecture we introduce the basics of CSS as a way to give different visual styles to HTML elements, changing their preset default appearance. *The nature of life is to grow.*

The bad way to produce styles

Tags such as **b**, **i**, **u**, and **font** are discouraged in strict HTML (XHTML)



```
<p>  
  <font face="Arial">Welcome to Greasy Joe's.</font>  
  You will <b>never</b>, <i>ever</i>, <u>EVER</u>  
  beat <font size="+4" color="red">OUR</font> prices!  
</p>
```



Welcome to Greasy Joe's. You will **never**, *ever*, EVER beat **OUR** prices!

Why is this bad?

Bad Practices, why?

Inline styles: the style attribute



```
<p style="font-family: sans-serif; color: red;">This is a paragraph</p>
```

Note: It has higher precedence than embedded or linked styles

Embedded styles



```
<head>
  <style type="text/css">
    p { font-family: sans-serif; color: red; }
    h2 { background-color: yellow; }
  </style>
</head>
```

Content vs. Presentation

- HTML is for content, the information on the page
- CSS is for presentation, how to display the page
- Keeping content separate from presentation is a very important web design principle
- If the HTML contains no styles, its entire appearance can be changed by swapping **.css** files

Cascading style sheets

It's called Cascading Style Sheets because the properties of an element cascade together in this order:

1. Browser's default styles
 2. External style sheet files (in a `<link>` tag)
 3. Internal style sheets (in a `<style>` tag in the page header)
 4. Inline style (the style attribute of an HTML element)
- Basically, cascading works from top to bottom inside the page (Depends on your order – later styles will always override top ones).

Linking external stylesheet (preferred way)

- Put all CSS rules in separate .css file
- Link in head section of html page using <link> tag.



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

Basic CSS rule syntax

- A **CSS** file consists of one or more rules
- A rule's selector specifies HTML element(s) and applies style properties
 - The `*` selector, selects all elements
 - To add a comment we use: `/* */`

```
selector {  
    property: value;  
    property: value;  
    ...  
}  
  
p {  
    font-family: sans-serif;  
    color: red;  
}
```



CSS properties for colors



```
p {  
  color: white;  
  background-color: blue;  
}
```



This paragraph uses the style above.

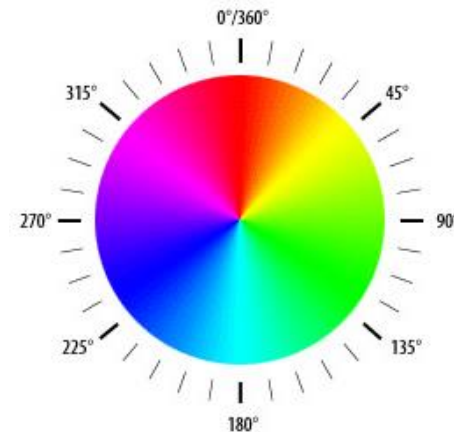
Specifying colors

- **Color names:** aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white (white), yellow
- **RGB & RGBA codes:** **red**, **green**, and **blue** values from 0 to 255
 - alpha (0-1) – defines opacity
- **HEX codes:** RGB values in base-16 from 00 (none) to FF (full)
- **HSL & HSLA codes:** HSL stands for **hue**, **saturation**, and **lightness**

Hue is a degree on the color wheel (from 0 to 360) - 0 (or 360) is red, 120 is green, 240 is blue.



```
h1 { color: red; }  
h2 { color: rgb(128, 0, 196); }  
h3 { color: rgba(128, 0, 196, 0.5); }  
h4 { color: #FF8800; }  
h5 { color: hsla(120, 60%, 70%, 0.3); }
```





CSS properties for fonts

property	description	Values
font-family	which font will be used	serif or “Courier New”
font-size	how large the letters will be drawn	A unit value, percentage, or named value
font-style	used to enable/disable italic style	normal(default), italic, oblique
font-weight	used to enable/disable bold style	normal(default), bold, bolder,...
font	Sets all font properties	style weigh size family
Complete list of font properties		

CSS properties for fonts

- If the first font is not found on the user's computer, the next is tried
- Placing a generic font name at the end of your font-family value ensures that every computer will use a valid font

Generic font names: serif, sans-serif, cursive, fantasy, monospace



```
h1{ /* which font will be used */
    font-family: Georgia;
}
h2 { /* enclose multi-word font names in quotes */
    font-family: "Courier New";
}
h3 { /* can specify multiple fonts from highest to lowest priority
    */
    font-family: Garamond, "Times New Roman", serif;
}
```

font-size, font-weight, font-style





```
p {  
    /* how large the letters will be drawn */  
    font-size: 14px;  
    /* used to enable/disable bold style */  
    font-weight: bold;  
    /* used to enable/disable italic style */  
    font-style: italic;  
}
```



Size Units

- **Units:** pixels (**px**), point (**pt**), m-size (**em**)
 - **pt** specifies number of points, where a point is 1/72 of an inch on screen
 - **px** specifies number of pixels on the screen
 - **em** relative to the font-size of the element (2em means 2 times the size of the current font)
- **Vague font sizes:** xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
- **Fluid Layout:** viewport height(vh), viewport width (vw)
 - https://www.w3schools.com/cssref/tryit.asp?filename=trycss_unit_vw


CSS properties for text

Property	Description	Values
text-align	alignment of text within its element	left, center, right, justify
text-decoration	decorations such as underlining	underline, overline, line-through, blink, none
text-indent	Indent first line	a size(px, pt, %, em)
line-height	vertical size of each line	a size(px, pt, %, em)
letter-spacing	Horizontal gap between letters	a size(px, pt, %, em)
word-spacing	Horizontal gap between words	a size(px, pt, %, em)
text-overflow 	How to handle too-long text	clip, ellipsis, ellipsis-word
text-shadow 	A “drop shadow” next to text	Two distances(px, pt, %, em) plus an optional shaow color

CSS properties for text

```
h2 {  
    /* Can also be overline, line-through, blink, or none.  
       Effects can be combined */  
    text-decoration: underline overline;  
    /* Shadow is specified as an X-offset, a Y-offset, and an optional color */  
    text-shadow: -2px 5px gray;  
}  
p {  
    /* Alignment of text within its element,  
       can be left, right, center, or justify */  
    text-align: center;  
    line-height: 30px;  
    word-spacing: 30px;  
    /* Indent the first line */  
    text-indent: 50px;  
}
```


CSS properties for background

Property	Description	Values
background-color	color to fill background	A color
background-image	image to place in background	url(image URL)
background-position	placement of bg image within element	Two tokens for x/y as top, bottom, left, right, center, or a size(pt, px, %, em)
background-repeat	whether/how bg image should be repeated	repeat(default), repeat-x, repeat-y or no-repeat
background-attachment	whether bg image scrolls with page	scroll(default), fixed
background-size 	scaling of bg image	a size(pt, px, %, em), cover, or contain
background	shorthand to set all background properties	

Background

```
body {  
    /* image to place in background */  
    background-image: url("../images/draft.jpg");  
    /* How bg image should be repeated */  
    /* can be repeat (default), repeat-x, repeat-y, or no-repeat */  
    background-repeat: repeat-x;  
    /* placement of bg image within element */  
    /* value consists of two tokens, can be top, left, right, bottom,  
       center, a percentage, or a length value in px, pt, etc */  
    background-position: 370px 20px;  
}  
p {  
    /* shorthand to set all background properties */  
    background: #ffffff url("image.png") no-repeat right top;  
}
```



Opacity

The **opacity** property sets the opacity level for an element. Value ranges from 1.0 (opaque) to 0.0 (transparent)



```
div {  
  opacity: 0.5;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor.

CSS properties for borders

`border: border-width border-style border-color;`



```
p { border: 5px solid red; }
```

border-color, border-width, border-style, border-bottom, border-left, border-right, border-top,
border-bottom-color, border-bottom-style, border-bottom-width, border-left-color, border-left-style,
border-left-width, border-right-color, border-right-style, border-right-width, border-top-color,
border-top-style, border-top-width



```
h2 {  
    border-left: thick dotted #CC0088;  
    border-bottom-color: rgb(0, 128, 128);  
    border-bottom-style: double;  
}
```

The `list-style-type` property

- `none` : No marker
- `disc` (default), `circle`, `square`
- `decimal` : 1, 2, 3, etc.
- `decimal-leading-zero` : 01, 02, 03, etc.
- `lower-roman` : i, ii, iii, iv, v, etc.
- `upper-roman` : I, II, III, IV, V, etc.
- `lower-alpha` : a, b, c, d, e, etc.
- `upper-alpha` : A, B, C, D, E, etc.
- `lower-greek` : alpha, beta, gamma, etc.



```
ol { list-style-type: lower-roman; }
```

Styling tables

All standard CSS styles can be applied to a table, row, or cell



```
table, td, th {  
    border: 2px solid black;  
}  
table {  
    border-collapse: collapse;  
}  
td {  
    background-color: yellow;  
    text-align: center;  
    width: 30%;  
}
```

Column 1	Column 2
1,1	1,2
2,1	2,2

Without border-collapse

Column 1	Column 2
1,1	1,2
2,1	2,2

With border-collapse

By default, the overall table has a separate border from each cell inside, the **border-collapse** property merges these borders into one.

Main Point

We discussed the CSS Properties for color, font, text, background, lists and table which are the basic properties used on almost every page.

There are a lot of details here, but few concepts. As long as you know the general concepts the details will follow with practice. Highest first—capture the fort to control the entire territory.

<http://www.creativebloq.com/web-design/10-design-concepts-web-developers-need-know-11135255>

Style inheritance

- Some styles, like font family, text-alignment etc., are automatically inherited by child elements from their parent element.
- Others are not automatically inherited (margin, padding).



```
body { font-family: sans-serif; background-color: blue; }  
p { color: red; }  
a { text-decoration: underline; }  
h2 { font-weight: bold; text-align: center; }
```




This is a heading.

A styled paragraph. [Previous slides](#) are available on the web site.

Body styles

- To apply a style to the entire body of your page, write a selector for the body element
- Saves you from manually applying a style to each element



```
body {  
    font-size: 16px;  
}
```

Browser CSS Reset code

- It's a good practice to reset some body values before we start coding our own CSS rules:

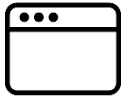
```
* {  
  margin: 0;  
  padding: 0;  
  font-size: 100%;  
  line-height: 1;  
}
```

Multiple selectors, Conflict & Overriding



```
/* select multiple elements separated by commas */
p, h1, h2 {
    color: green;
    background-color: grey;
}
/* when two styles set conflicting values for the same
property, the latter style takes precedence */

h2 {
    background-color: blue;
}
```



This paragraph uses the above style.

This h2 uses the above styles.

The HTML `class` and `id` attribute

- **`id` attribute** allows you to give a unique ID to any element on a page
 - Each ID must be unique; can only be used once in the page
- **`class` attribute** is used to group some elements and give a style to only that group
 - unlike an id, a class can be reused as much as you like on the page

class vs id examples



```
<p id="mission">Our mission is to provide the most</p>
<p class="special">See our spectacular spatula specials</p>
<p class="special shout">Today only, satisfaction guaranteed</p>
```

```
#mission {
  font-style: italic;
  color: #000000;
}
```



```
.special { /* any element with class="special" */
  background-color: yellow; font-weight: bold;
}
p.shout { /* only p elements with class="shout" */
  color: red;
  font-family: cursive;
}
```

Class naming

- focus on the semantics and meaning of the content vs appearance
- Bad example: redtext, bigfont
 - if change style later, it doesn't make sense to be called redtext.
- Good example:
 - warningMsg
 - errorMsg

CSS pseudo-classes pseudo-elements

- A **pseudo-class** is used to define a special state of an element
 - Style an element when a user mouse's over it
 - Style visited and unvisited links differently
 - Style an element when it gets focus
- A CSS **pseudo-element** is used to style specified parts of an element
 - Style the first letter, or line, of an element
 - Insert content before, or after, the content of an element



```
selector:pseudo-class { property:value; }  
selector::pseudo-element { property:value; }  
/* Notice the double colon notation - ::pseudo-element versus  
:pseudo-class */
```

CSS pseudo-classes pseudo-elements

class	description
:active	an activated or selected element
:focus	an element that has the keyboard focus
:hover	an element that has the mouse over it
:link	a link that has not been visited
:visited	a link that has already been visited
:first-child	an element that is the first one to appear inside another
:nth-child(N)	applies to EVERY Nth child of its parent (even, odd, n)
::first-letter	the first letter of text inside an element
::first-line	the first line of text inside an element

Example: Specify a background color for every <p> element that is the second child of its parent:

```
p:nth-child(2) {  
  background: red;  
}
```


Examples pseudo-classes



```
/* unvisited link */
a:link {
    color: #FF0000;
}
/* visited link */
a:visited {
    color: #00FF00;
}
/* mouse over link */
a:hover {
    color: #FF00FF;
}
/* click on a link */
a:active {
    color: #0000FF;
}
```

More info and examples: [Pseudo-classes](#) and [Pseudo-elements](#)

Main Point


- The Cascading in CSS indicates that there are multiple levels of style sheets. More specific styles overwrite more general styles. We can be more specific by using Class selectors (can apply to multiple elements) and even more so with id selectors (individual elements) and context selectors.
Life is found in layers.

Main Point

The `<div>` tag provides a generic block level element that can be used for any division or section of your page. The `` tag provides a generic inline element for specifying any range of text inside a box. By using these tags, combined with CSS selectors we can write powerful and reusable CSS rules to style and layout pages. *This is an example of efficiency through abstraction. We can have rules that apply to many different elements by abstracting over tags and classes. Do less and accomplish more by acting from abstract levels of awareness.*

W3C CSS Validator

- Check your CSS to make sure it meets the official CSS specifications
- Pickier than the web browser, which may render malformed CSS correctly



```
<p>  
  <a href="https://jigsaw.w3.org/css-validator/check/referer">  
    </a>  
</p>
```

Assignments

1. Style your home page from yesterday, follow instructions starting at this [slide](#).
2. Design web page based on [this specification](#).
 - You can get the text content [from here](#).
 - You can get images [from here](#).