

Lecture 2: CSS

CS472 Web Programming

Maharishi University of Management

Department of Computer Science

Assistant Professor Obinna Kalu, September 2017

Except where otherwise noted, the contents of this document are Copyright 2012 Marty Stepp, Jessica Miller, Victoria Kirst and Roy McElmurry IV. All rights reserved. Any redistribution, reproduction, transmission, or storage of part or all of the contents in any form is prohibited without the author's expressed written permission. Slides have been modified for Maharishi University of Management Computer Science course CS472 in accordance with instructors agreement with authors.

Maharishi University of Management - Fairfield, Iowa © 2017




All rights reserved. No part of this slide presentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or recording, or by any information storage and retrieval system, without permission in writing from Maharishi University of Management.

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. **Science of Consciousness:** The nature of life is to grow.

The bad way to produce styles

Tags such as **strong**, **em**, **u**, and **font** are discouraged in strict HTML



```
<p>
  <font face="Arial">Welcome to Greasy Joe's.</font>
  You will <strong>never</strong>, <em>ever</em>, <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?

Cascading Style Sheets (CSS): `<link>`

- CSS describes the appearance and layout of information on a web page (as opposed to HTML, which describes the content of the page)
- Can be embedded in HTML or placed into separate `.css` file (preferred)



```
<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: `/* */`

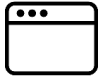
```
selector {  
    property: value;  
    property: value;  
    ...  
}  
  
p {  
    font-family: sans-serif;  
    color: red;  
}
```



CSS properties for colors



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



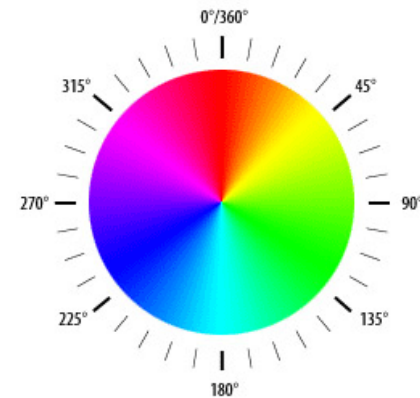
This paragraph uses the style above.

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
- **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

- 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: 14vw;  
  /* used to enable/disable bold style */  
  font-weight: bold;  
  /* used to enable/disable italic style */  
  font-style: italic;  
}
```

Size Units

- pixels (**px**), point (**pt**), m-size (**em**), viewport height(**vh**), viewport width (**vw**)
- xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
- percentage font sizes, e.g.: 90%, 120%

px vs pt vs em vs vh vs vw

- **pt** specifies number of points, where a point is 1/72 of an inch on screen
- **px** specifies a 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)
- **Viewport** is the browser window size. If the viewport is 50cm wide, 1vw = 0.5cm

@font-face



The **@font-face** rule allows you to specify font files. You no longer have to worry about whether the viewer of the site actually has the font that you specified using font-family



```
/* can't be used as element style!*/
@font-face {
  font-family: myFont;
  src: url('chap03/Purisa.ttf');
}
p {
  font-family: myFont;
}
```

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;
}
```

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.

W3C CSS Validator

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



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

Main Point

Cascading Style Sheets (CSS) provide a way for web developers to specify the appearance of content on the page by using selectors to specify the element(s), and then the values that properties of those elements (like font or color) should have. It is a best practice to keep CSS in a separate file, doing so makes your pages more flexible. **Science of Consciousness:** Purification leads to progress.

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;  
    /* Space between the lines in two paragraphs */  
    line-height: 30px;  
    /* Space between words in <p> elements should be 30 pixels */  
    word-spacing: 30px;  
    /* Indent the first line of all <p> elements with 50 pixels */  
    text-indent: 50px;  
}
```

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; }
```

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;  
}
```

Main Point

We discussed the CSS Properties for color, font, text, background, and lists, which are the basic properties used on almost every page. **Science of Consciousness:** Seek the highest first.

Bad Practices, why?

Embedding style sheets



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

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

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 (reference)
 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).

Inheriting styles

- When multiple styles apply to an element, they are inherited
- **Not all properties are inherited** (notice link's color below) (*width, background, float.. etc*)



```
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.

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-line	the first line of text inside an element
: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

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](#)

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;
}
```

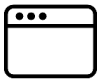
Override Rules



```
<p class="RedColor BlueColor"> Lorem Ipsum </p>
```



```
#YellowColor{ color: yellow; }  
.BlueColor{ color: blue; }  
.RedColor{ color: red; }
```



Lorem Ipsum

What will happen if we add `id="YellowColor"` ?

What will happen if we update the CSS code to `.BlueColor{ color: blue !important; }` ?

Style Specificity

When multiple styles apply to an element and have the same origin precedence.

The most specific one applies. If they have the same specificity, then the later one will be used.



```
aside { color: gray; }  
p { color: green; }  
em { color: yellow; }  
.awesome { color: blue; }  
em.awesome { color: red; }  
#recent { color: black; }  
em#recent.awesome { color: orange; }
```



Which awesome color?



```
<aside>  
  <p> <em id="recent" class="awesome">Which awesome color?</em> </p>  
</aside>
```

Main Point

The Cascading in CSS indicates that there are multiple levels of style sheets. More specific styles overwrite more general styles. We can specify things more specifically by using Class selectors (groups of elements) and even more so with id selectors (individual elements). **Science of Consciousness:** Life is found in layers.

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

Changing Appearances

1. How a page is displayed is affected by both the HTML and the CSS
2. Although every HTML tag has a default way of displaying, it can easily be changed with CSS and should never be the basis for using it. Instead use HTML tags based on meaning.

-
1. **Transcendental consciousness** is the field that underlies all differences.
 2. **Impulses within the Transcendental field:** the infinitely dynamism of the unified field constantly expresses itself and becomes all aspects of creation.
 3. **Wholeness moving within itself:** In Unity Consciousness, one experiences that this unbounded diversity is nothing but the self.

