CS472 WAP

# CSS Levels

Cascading Style Sheets (CSS)

Diversity from Unity

**Maharishi International University**
**Fairfield, Iowa** © 2020

## Main Point Preview

- 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 (groups of elements) and even more so with id selectors (individual elements).
- *Life is found in layers.*

## 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;
}
```

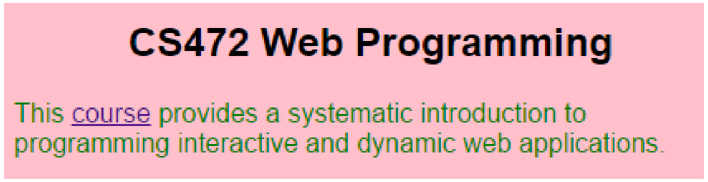
# Inheriting styles

Styles get inherited from containing elements

**Not all properties are inherited** (notice link's color below)

- E.g., margin

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



## CS472 Web Programming

This course provides a systematic introduction to programming interactive and dynamic web applications.

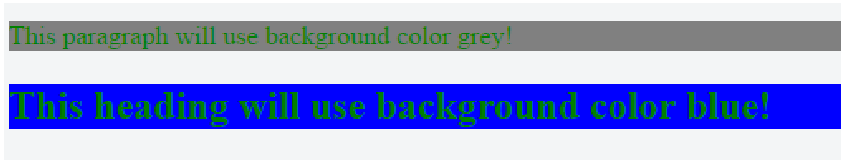
## Browser CSS Reset code

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

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

## Styles that Conflict

```
/* 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;
}
<p> This paragraph will use background color grey! </p>
<h2> This heading will use background color blue! </h2>
```



This paragraph will use background color grey!

This heading will use background color blue!

## Styles Pane (Chrome, Firefox)

- Invoke the DevTools inspector on element
- displays styles for selected DOM node
- gray background are read-only
- exclamation marks mean that property name and/or value is not understood by the browser,
  - is ignored
- A style declaration may contain several properties with the same name. Only the last one takes effect, canceling the preceding ones.
  - will be struck out, like overridden properties.

## *Cascading* style sheets

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

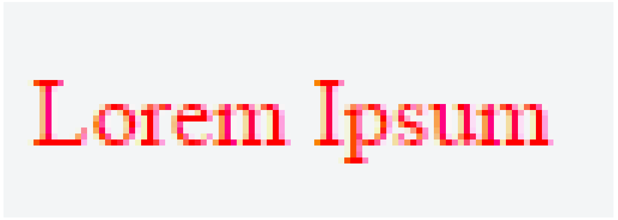
1. Browser's default styles (reference)
2. External style sheet files (in a <link> tag)
3. Internal style sheets (in a <style> tag in the page header)
4. Inline style (the style attribute of an HTML element)

- Basically, cascading works from top to bottom inside the page (Depends on your order – later styles will always override top ones).
- Inheritance is how elements in the HTML markup inherit properties from their parent elements
- cascade is how different CSS rule sets are applied to HTML elements, and how conflicting rules do or don’t override each other.

## Override Rules

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

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



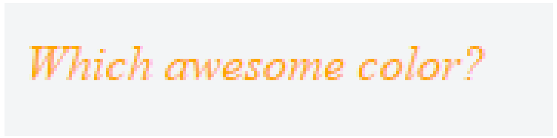
Lorem Ipsum

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

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


*Which awesome color?*

## Specificity and conflicts

- Specificity- decide which one should win when two or more rules conflict.
- Rules: each rule’s overall selector is given a score based upon approximately the following rules. The rule with the highest score wins if there’s a conflict.
  - Any HTML element mentioned in the rule scores 1 point
  - Any class mentioned in the rule scores 10 points
  - Any ID mentioned in the rule scores 100 points
- Examples:
  - p.banner - 11
  - div.box > p - 12
  - body #logo .box p.banner - 122

## The HTML `class` and `id` attribute

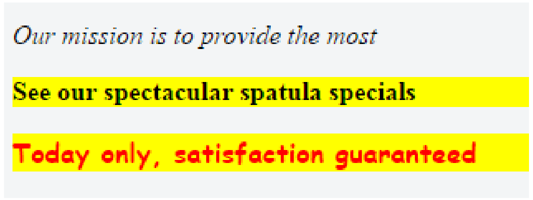
- **`id` attribute** allows you to give a unique ID to any element on a page
  - Each ID must be unique;
  - Can only be used once in the page

- **`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 { /* the element with id="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;
}
```



*Our mission is to provide the most*

**See our spectacular spatula specials**

**Today only, satisfaction guaranteed**

## class naming

- focus on the semantics and meaning of the content vs appearance
  - Bad examples: redtext, bigfont
    - if change style later, it doesn’t make sense to be called redtext.
  - Good examples: warning-msg, error-msg

## pseudo-classes and 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

h

- A CSS **`pseudo-element`** is used to style specified parts of an element
  - *Do not exist in the DOM, are created with CSS*
  - Style the first letter, or line, of an element
    - ::first-line, ::first-letter
  - Insert content (pseudo element) before, or after, the content of an element
    - ::before,  ::after

```
selector:pseudo-class { property:value; } /* single colon */

selector::pseudo-element { property:value; }  /* double colon */
```

## 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 |
| :nth-child(`expr`)<br><br>:first-child, :last-child | targets specific children of a given element |
| :not(`selector`) | all elements that do not match the given CSS selector |
| ::first-line | the first line of text inside an element |
| ::first-letter | the first letter of text inside an element |

Complete list

## 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

## CSS context selectors

`selector1 selector2 {  properties }`

Applies the given properties to **selector2 only** if it is inside a selector1 on the page

`selector1 > selector2 {  properties }`

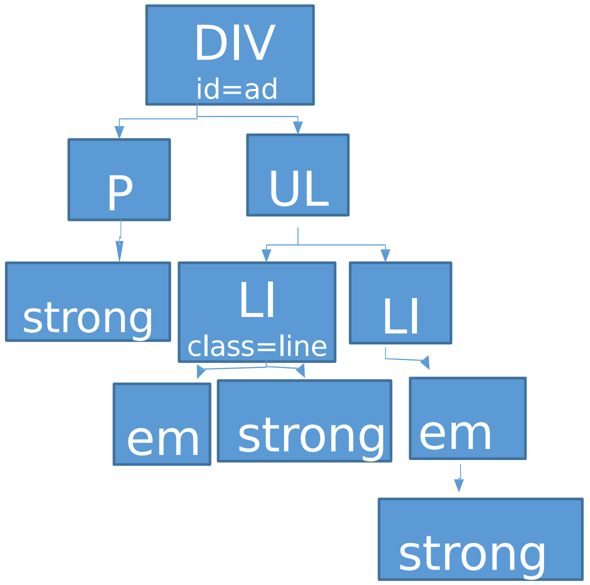
- Applies the given properties to **selector2 only** if it is direct child of selector1 (in the DOM)
  - mneumonic: think of > as indicating a direct link (to a child)

## Example

```
<div id="ad">
  <p>Shop at <strong>Hardwick's Hardware</strong></p>
  <ul>
    <li class="line"><em>The </em><strong>best</strong> prices!</li>
    <li><em><strong>Act while supplies last!</strong></em></li>
  </ul>
</div>
```

```
ul > li { background-color: blue; }
li strong { color: red; }
li > strong { color: green; }
#ad li.line strong { text-decoration: underline; }
```

See example: lesson3_examples\contextselector.html, contextselectordirect.html



## Add *content* to your website using CSS

- CSS has a property called `content`.
- only be used with the pseudo elements `::after` and `::before`.
- The ::after selector inserts something after the content of each selected element(s).
  - Use the content property to specify the content to insert.
  - Use the ::before selector to insert something before the content.

- https://www.w3schools.com/cssref/tryit.asp?filename=trycss_sel_after

# 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 (groups of elements) and even more so with id selectors (individual elements).
- *Life is found in layers.*

MAHARISHI INTERNATIONAL UNIVERSITY

HIGHER EDUCATION • FOR HIGHER CONSCIOUSNESS