Cascade and Inheritance

Harry J. Wang, Ph.D.

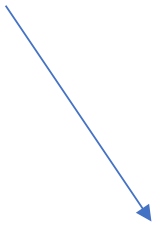
University of Delaware

Fall 2017

Review: Internal Style Sheet

- Use <style> to include CSS code in HTML files.
- To style an HTML element using CSS, you must first select that element in the CSS file (selector is discussed later in this lecture)

```
<head>
  <title>Animals Around the World</title>
<style>
    h1 {
    font-family: Arial;
    }
  </style>
</head>
<body>
<h1>BREAKING NEWS</h1>
```



Review: External Style Sheet

- Link CSS files with HTML using <link> tag (URL vs. relative link paths)

Animals.html:

```
<head>
  <title>Animals Around the World</title>
<link href="https://www.codecademy.com/stylesheets/style.css" type="text/css" rel="stylesheet">
<link href="/animals.css" type="text/css" rel="stylesheet">
</head>
```

Animals.css:

animals.css

```
1   h1 {
2   font-family: Arial;
3   }
```

Inline Style

```
<!DOCTYPE html>
<html>
  <head>
    <title>Playing with Inline Styles</title>
  </head>
  <body>
    <p style="color:blue;font-size:46px;">
      I'm a big, blue, <strong>strong</strong> paragraph
    </p>
  </body>
</html>
```

Cascading Order

- The default priority order, where no.1 as the most important:
  1. Inline styles
  2. Internal style sheet
  3. External style sheets
  4. Browser default (e.g., hyper link is blue)
- If two rules have the same weight, the latter wins.

Internal Priorities

- Internal priorities for CSS (no. 1 has the highest priority):
  1. #id
  2. .class
  3. element
- Specificity wins, header.masthead img will override img
- W3C table on calculating internal weight for selectors:
  - **a** represents the number of #id attributes in the selector (100 points each)
  - **b** represents the number of class attributes (10 point each)
  - **c** represents the number of tag names (1 point each)

li	a=0 b=0 c=1 -> specificity = 1
ul li	a=0 b=0 c=2 -> specificity = 2
ul ol li	a=0 b=0 c=3 -> specificity = 3
li.red	a=0 b=1 c=1 -> specificity = 11
ul ol li.red	a=0 b=1 c=3 -> specificity = 13
#slogan	a=1 b=0 c=0 -> specificity = 100

!important

- The !important rule overrides all!
- If !important is specified more than once for the same element, they will fall back to the normal cascading rules.
- Also, declaring a shorthand property to be !important is equivalent to declaring all of its sub-properties to be !important.

```
.class-style {
  background-color: gray;
  border: none !important;
}
```

- **DO NOT use !important, unless you have to!**
- See an example: https://jsfiddle.net/harryjwang/ghmuxsdf/

Inheritance

- Some property values applied to an element will be inherited by that element‘s children, and some won’t, e.g.:
  - font-family and color are to be inherited, so that one can set a site-wide base font by applying a font-family to the <html> element and then override the fonts on individual elements where needed.
  - margin, padding, border, and background-image are NOT to be inherited, so that set these properties on a container element will not mess up the style of the child elements.
- Use common sense to decide whether a property is inherited or not or check CSS Reference to be sure.

Controlling Inheritance

- CSS provides three special values to handle inheritance:
  - inherit : This value sets the property value applied to a selected element to be the same as that of its parent element.
  - initial : This value sets the property value applied to a selected element to be the same as the value set for that element in the browser's default style sheet. If no value is set by the browser's default style sheet and the property is naturally inherited, then the property value is set to inherit instead.
  - unset : This value resets the property to its natural value, which means that if the property is naturally inherited it acts like inherit, otherwise it acts like initial.
- See an example: https://jsfiddle.net/harryjwang/bss34qyu/

Chrome DevTools

- Open Chrome DevTools (three ways):
  - Select **More Tools** > **Developer Tools** from Chrome's Main Menu.
  - Right-click a page element and select **Inspect**.
  - Press Command+Option+I (Mac) or Control+Shift+I (Windows, Linux).
- Inspect and Edit Pages and Styles: https://developers.google.com/web/tools/chrome-devtools/inspect-styles/
- Viewing And Changing CSS: https://developers.google.com/web/tools/chrome-devtools/css/

References

- http://monc.se/kitchen/38/cascading-order-and-inheritance-in-css
- https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Cascade_and_inheritance

Questions?