Introduction to Cascade Style Sheet

Instructor: Dr.Dendej Rajrattanatrai

INTRODUCTION TO CSS

Why CSS?

- HTML was never intended for formatting or styling purposes but to define the content of a document
- Once you embed presentation inside of an HTML tag, it cannot be overridden
- The bottom line: separate the structure and content
- Gives the designer creative control of the content. What you can change:
  - Font and font sizes
  - Color
  - Layout
  - Position
  - Borders
- Control the layout of content on multiple devices using multiple style sheets (and thus improving accessibility)
- Provides visual feedback to users (Web Engineering)

Inserting and Using CSS

- As external file; declare in <head> section (preferred)

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

- Internally in <head> section (meh!)

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

- Inline within an element

```
<p style="...">Whoa there!</p>
```

Embedding a Style Sheet

- <!DOCTYPE html>
- <html lang="en">
-   <head>
-     <title>Sample</title>
-       <style type="text/css" media="all”>
-           p.Code {
-             margin: 0 .5in 0 .5in;
-             padding: 5px;
-           }
-       </style>
-   </head>
- <body>
-   <p>A Smalltalk example </p>
-   <p class="code">1000 factorial printString size</p>
- </body>
- </html>

Linking CSS to a Web page

- Link to External Style Sheet
- Embedding a Style Sheet
- Importing a Style Sheet
- Inlining a Style Sheet

Linking to Multiple Style Sheets

```
<!DOCTYPE>
<html lang="en">
  <head>
    <title>Sample</title>
      <link rel="Stylesheet" href="simple.css"
type="text/css"           media="screen" />
      <link rel="Stylesheet" href="small.css"
type="text/css"    media="handheld" />
      <link rel="Stylesheet" href="print.css"
type="text/css"    media="print" />
</head>
…
…
```

Media Types

Type	Description
braille	Braille tactile feedback devices
embossed	Paged braille printers
handheld	Handheld devices
print	Documents viewed in print preview mode & sent to printer
projection	Projected presentations
screen	Computer screens
speech	Speech synthesizers
tty	Fixed-pitch character grid
tv	Television-type devices

Different CSS based on size

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>SizeDetect</title>
    <link rel="Stylesheet" href="phone.css" type="text/css"
       media="screen and (max-device-width: 320px)"/>
    <link rel="Stylesheet" href="ipad.css" type="text/css"
       media="screen and (min-device-width: 768px)
       and (max-device-width: 768px)"/>
  </head>
<body>
  <p>Red = phone, Blue = iPad, Black = desktop</p>
</body>
</html>
```

All Selectors (CSS1 to CSS3)

*	E:first-child	E:enabled
E	E:last-child	E:disabled
E[foo]	E:first-of-type	E:checked
E[foo="bar"]	E:last-of-type	E::first-line
E[foo~="bar"]	E:only-child	E::first-letter
E[foo^="bar"]	E:only-of-type	E::before
E[foo$="bar"]	E:empty	E::after
E[foo*="bar"]	E:link	E.warning
E[foo|="en"]	E:visited	E#myid
E:root	E:active	E:not(s)
E:nth-child(n)	E:hover	E F
E:nth-last-child(n)	E:focus	E > F
E:nth-of-type(n)	E:target	E + F
E:nth-last-of-type(n)	E:lang(fr)	E ~ F

Syntax

- A CSS file contains a sequence of rules
- Example rule:

```
h1 { color: blue; }
```

  - `h1` is known as the selector
  - `color: blue;` is known as a declaration
  - `color` is known as the property
  - `blue` is known as the value
- The selector comes first, then the declarations are enclosed by brackets
- A declaration is a property-value pair separated by a colon
- Declarations always ends with a semicolon
- A rule can have many declarations
- You can put one declaration on each line
- To make a comment in a CSS, use /* COMMENT HERE */

Multiple Declarations

```
h1 {
font-style: normal;
font-weight: bolder;
text-align: center;
font-size: medium;
}
```

Grouping Selectors

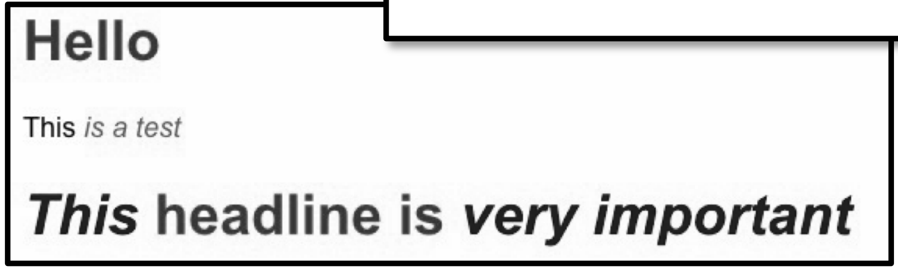
```
h1
{
font-style: normal;
text-align: center;
}

h2
{
font-style: normal;
text-align: center;
}
```

```
h1, h2
{
font-style: normal;
text-align: center;
}
```

Descendant Selectors

- h1 { color: red }
- em { color: green }
- h1 em { color: blue }

To <u>match</u> “em” inside of h1



Hello

This *is a test*

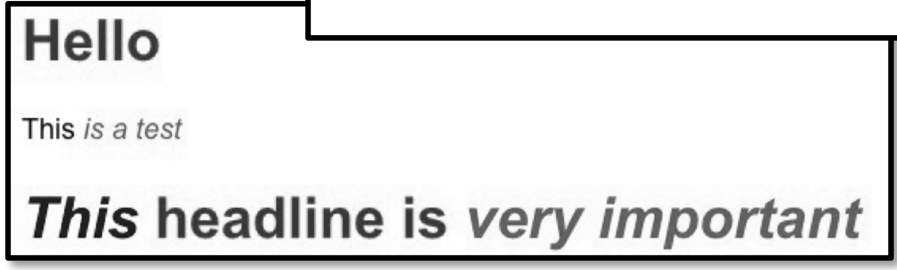
This **headline is** ***very important***

```
<h1>Hello</h1>
<p>This <em>is a test</em></p>
<h1><em>This</em>
 <span>headline is <em>very important</em>
</span></h1>
```

Child Selectors

```
h1 { color: red }
em { color: green }
h1 > em { color: blue }
```

To match “em” directly inside h1



Hello

This *is a test*

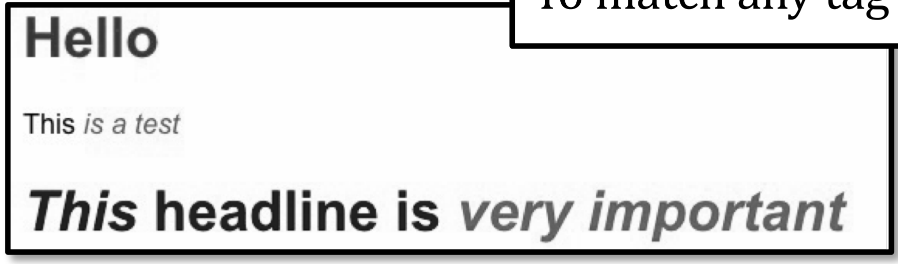
This* headline is *very important

```
<h1>Hello</h1>
<p>This <em>is a test</em></p>
<h1><em>This</em>
 <span>headline is <em>very important</em>
</span></h1>
```

Universal Selector (*)

```
h1 { color: red }
em { color: green }
h1 > * { color: blue }
```

To match any tag



Hello

This *is a test*

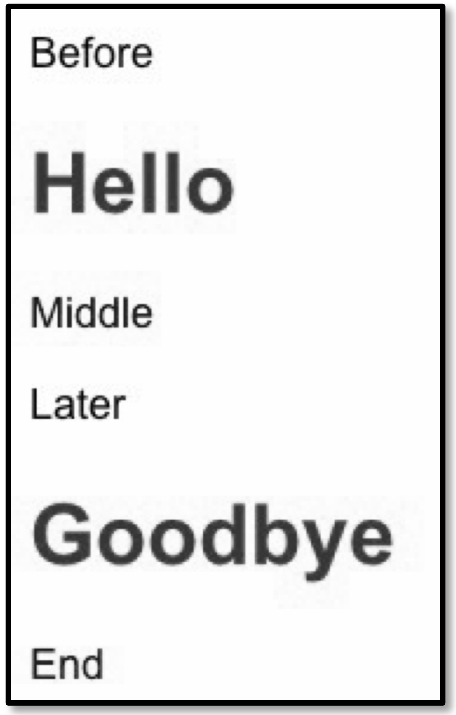
This* headline is *very important

```
<h1>Hello</h1>
<p>This <em>is a test</em></p>
<h1><em>This</em>
<span>headline is <em>very
important</em>
</span></h1>
```

Adjacent Sibling Selector

```
h1 { color: red }
em { color: green }
h1 + p { color: blue }
```

```
<p>Before</p>
<h1>Hello</h1>
<p>Middle</p>
<p>Later</p>
<h1>Goodbye</h1>
<p>End</p>
```


Before

Hello

Middle

Later

Goodbye

End

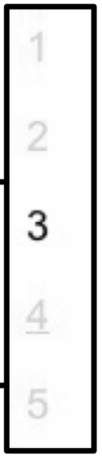
match p immediately after h1

Attribute Selector

```
<div name="sam">1</div>
<p name="sam" >2</p>
<p>3</p>
<a href="index.html" name="pete">4</a>
<p name="sam roger pete" >5</p>
```

[name] { color: lightblue}

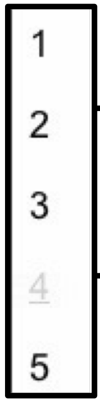
All tags with name attribute



```
1
2
3
4
5
```

a[name] { color: lightblue}

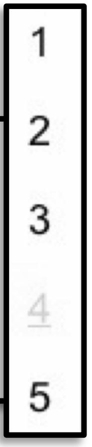
All a tags with name attribute



```
1
2
3
4
5
```

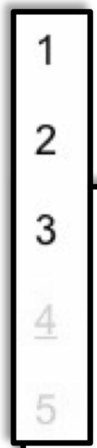
Attribute Selector

```
<div name="sam">1</div>
<p name="sam" >2</p>
<p>3</p>
<a href="index.html" name="pete">4</a>
<p name="sam roger pete" >5</p>
```


1
2
3
4
5

[name=pete] {color: lightblue}

All tags with name attribute with value equal to "pete"


1
2
3
4
5

[name~=pete] {color: lightblue; }

All tags with name attribute with values containing "pete" in whitespace list

Some Attribute Syntax

E[foo]	an E element with a "foo" attribute
E[foo="bar"]	an E element whose "foo" attribute value is exactly equal to "bar"
E[foo~="bar"]	an E element whose "foo" attribute value is a list of whitespace-separated values, one of which is exactly equal to "bar"
E[foo^="bar"]	an E element whose "foo" attribute value begins exactly with the string "bar"
E[foo$="bar"]	an E element whose "foo" attribute value ends exactly with the string "bar"
E[foo*="bar"]	an E element whose "foo" attribute value contains the substring "bar"
E[foo|="en"]	an E element whose "foo" attribute has a hyphen-separated list of values beginning (from the left) with "en"

Class selector

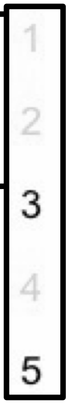
- Used to specify a style for a group of elements
- Class selector is most often used on several elements
- The class selector uses the HTML class attribute, and is defined with a "."
- class attribute can contain multiple class names, separated by spaces

More Class Selector Samples

```
<div class="sam">1</div>
<p class="sam" >2</p>
<p class="pete">3</p>
<a class="pete sam">4</a>
<p name="roger sam pete" >5</p>
```

.sam { color: lightblue }

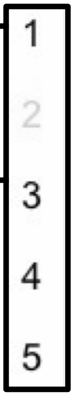
All tags with class sam



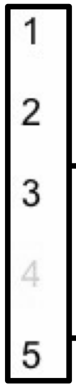
```
1
2
3
4
5
```

p.sam { color: lightblue }

All p tags with class sam



```
1
2
3
4
5
```



```
1
2
3
4
5
```

.sam.pete { color: lightblue }

All tags with class sam & pete

Pseudo-Classes

`:hover` - Hover over a link

`:focus` - Link is focused

`:active` - Click on link

`:link` - Unvisited links

`:visited` - Visited links

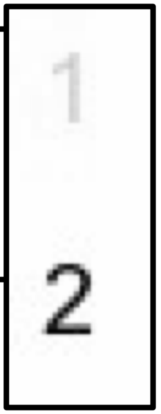
ID selector

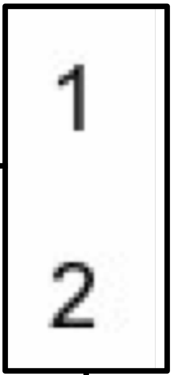
- Used to specify a style for a single, unique element
- Uses the id attribute of the HTML element.
- **<u>the ID cannot start with a number</u>**
- Defined with a “#”

ID Selector Samples

```
<div id="sam">1</div>
<p id="pete" >2</p>
```

#sam { color: lightblue }
Match the one element
with id = sam





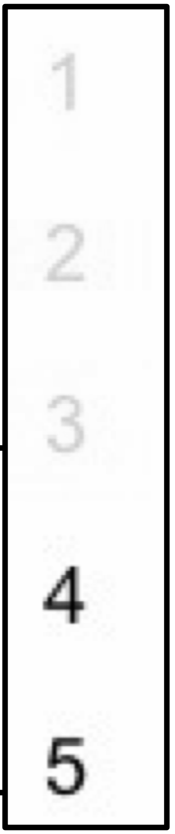
p#sam { color: lightblue }
Match the one p element
with id = sam

Multiple id same value - illegal

- <div id="sam">1</div>
- <p id="sam" >2</p>
- <p id="sam">3</p>
- <a id="pete sam">4</a>
- <p id="roger sam pete" >5</p>

```
#sam { color: lightblue }
```

Safari, Firefox, Chrome
Opera for mac



Cascading Rules

- What if there is more than one style specified for an HTML element?
- The styles will roll-up or "cascade" into a bigger rule
- The last declaration wins!
- Priorities:
  - Browser default
  - External style sheet
  - Internal style sheet
  - Inline style

Advanced Compositions

- Concatenation: separated by "."
  - Example: **`p.highlight.small { color: red; }`**
    - Applies to all paragraph elements whose class contains highlight and small
- "Or", separated by ","
  - Example: **`h1, h2, h3 {...}`**
    - Applies to <h1>, <h2> or <h3> elements
- Descendants, separated by *space*
  - Example: **`form p {...}`**
    - Applies to all paragraph tags inside form elements
- Child of, separated by ">"
  - Example: **`form > p {...}`**
    - Applies to all paragraph elements that are direct children of form elements