

## External Styles: Example (2)

```
<!DOCTYPE html>
<html>
<head>
  <title>Importing style sheets</title>
  <link type="text/css" rel="stylesheet"
        href="styles.css" />
</head>
<body>
  <h1>Shopping list for <em>Monday</em>:</h1>
  <li>Milk</li>
  ...
```

33

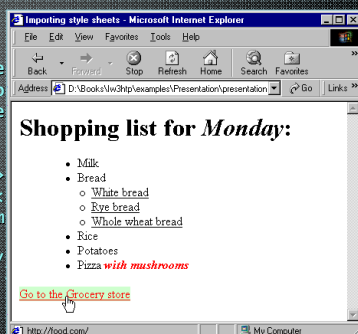
## External Styles: Example (3)

```
...
<li>Bread
  <ul>
    <li>White bread</li>
    <li>Rye bread</li>
    <li>Whole wheat bread</li>
  </ul>
</li>
<li>Rice</li>
<li>Potatoes</li>
<li>Pizza <em>with mushrooms</em></li>
</ul>
<a href="http://food.com" title="grocery
  store">Go to the Grocery store</a>
</body>
</html>
```

34

## External Styles: Example (4)

```
...
<li>Bread
  <ul>
    <li>White
    <li>Rye b
    <li>Whole
  </ul>
</li>
<li>Rice</li>
<li>Potatoes<
<li>Pizza <em>
</ul>
<a href="http://
  store">Go to
</body>
</html>
```



35

## Attribute Selectors

Picking Elements with Certain Attributes



## Attribute Selectors

- ◆ [ ] selects elements based on attributes

- ◆ Element with a given attribute

```
a[title] {color:black}
```

Selects <a> elements with title

- ◆ Elements with a concrete attribute value

```
input[type=text] { font-family:Consolas}
```

- ◆ Selects <input> elements with type=text

- ◆ Elements whose attribute values contain a word

```
a[title*=logo] {border: none}
```

- ◆ Selects <a> elements whose title attribute value contains logo

## Attribute Selectors

Live Demo



## Pseudo Selectors

Relative to Element Content or State

```
html, body, div { padding: 0; margin: 0; font-size: 100%; }  
/* Start of "Micro clearfix" w/  
* ( ( 2000-12 )  
* :before;  
* :after { content: ""; display: table; }  
* :after { clear: both; }
```

**:before and :after**

```
body  
<div class="container">  
  <div class="section">first</div>  
  <div class="section">last</div>  
</div>
```

## Common Pseudo Selectors

- ◆ Pseudo-classes define state

- ◆ :hover, :visited, :active, :lang

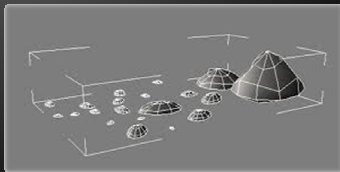
- ◆ Pseudo-elements define element "parts" or are used to generate content

- ◆ :first-line, :before, :after

```
a:hover { color: red; }  
p:first-line { text-transform: uppercase; }  
.title:before { content: ">"; }  
.title:after { content: "<"; }
```

## Common Pseudo Selectors

Live Demo



## Structural Pseudo-classes

- ◆ **:root**
  - ◆ The root of the document
- ◆ **E:nth-child(n)**
  - ◆ An E element, the n-th child of its parent
- ◆ **E:nth-last-child(n)**
  - ◆ An E element, the n-th child of its parent, counting from the last on
- ◆ **E:nth-of-type(n)**
  - ◆ An E element, the n-th sibling of its type

43

## Structural Pseudo-classes (2)

- ◆ **E:nth-last-of-type(n)**
  - ◆ An E element, the n-th sibling of its type, counting from the last one
- ◆ **E:last-child**
  - ◆ An E element, last child of its parent
- ◆ **E:first-of-type**
  - ◆ An E element, first sibling of its type
- ◆ **E:last-of-type**
  - ◆ An E element, last sibling of its type

43

## Structural Pseudo-classes (3)

- ◆ **E:only-child**
  - ◆ An E element, only child of its parent
- ◆ **E:only-of-type**
  - ◆ An E element, only sibling of its type
- ◆ **E:empty**
  - ◆ An E element that has no children (including text nodes)
- ◆ More detailed descriptions:  
<http://www.w3.org/TR/css3-selectors/#structural-pseudos>

44



## Structural Selectors

Live Demo

## The UI Element States Pseudo-Classes

- ◆ **E:enabled**
  - ◆ A user interface element E which is enabled
- ◆ **E:disabled**
  - ◆ A user interface element E which is disabled
- ◆ **E:checked**
  - ◆ A user interface element E which is checked (for instance a radio-button or checkbox)
  - ◆ Currently supported only in Opera and IE10 !

46



## UI Selectors

Live Demo

## Other CSS 3 Selectors

- ◆ **E:target**
  - ◆ An E element being the target of the referring URI
- ◆ **E:not(s)**
  - ◆ An E element that does not match simple selector
- ◆ **E ~ F**
  - ◆ An F element preceded by an E element

48

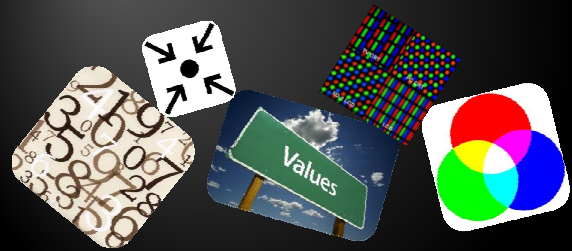


## Other CSS 3 Selectors

Live Demo

## CSS Values

Types, Ranges, Units



## CSS Values

- ◆ All values in CSS are strings
  - ◆ They can represent values that are not strings
  - ◆ I.e. 14px means size 14 pixels
- ◆ Colors are set in a red-green-blue format (RGB)
  - ◆ Both in hex and decimal

```
li.nav-item {  
  color: #44f1e1;  
}
```

```
li.nav-item {  
  color: rgb(68, 241, 255)  
}
```

51

## Size Values

- ◆ When setting a size (width, height, font-size...) the values are given as numbers
  - ◆ Multiple formats / metrics may be used
  - ◆ Pixels, ems, e.g. 12px, 1.4em
  - ◆ Points, inches, centimeters, millimeters
    - ◆ E.g. 10pt, 1in, 1cm, 1mm
  - ◆ Percentages, e.g. 50%
    - ◆ Of the size of the container/font size
  - ◆ Zero can be used with no unit: border: 0;

52

## Size Values

Live Demo



## Color Values

Colors in CSS can be represented in few ways

- Using red-green-blue

- Or red-green-blue-alpha

The opacity values are from 0.0 to 1.0

```
color: #f1a2ff  
color: rgb(241, 162, 255)  
color: rgba(241, 162, 255, 0.1)
```

- Using hue-saturation-light

- Or hue-saturation-light-alpha

```
color: hsl(291, 85%, 89%);  
color: hsl(291, 85%, 89%, 0.1);
```

## RGB Colors

- RGB colors are defined with values for red, green and blue intensity

- Syntax:

- #44fa36 – values are in hex

- rgb(<red>, <green>, <blue>) – decimal values

- The range for red, green and blue is between integers 0 and 255

```
color: #07f2b3  
color: rgb(7, 242, 179)
```



## RGBA Colors

- Standard RGB colors with an opacity value for the color (alpha channel)

- Syntax: rgba(<red>, <green>, <blue>, <alpha>)

- The range for red, green and blue is between integers 0 and 255

- The range for the alpha channel is between 0.0 and 1.0

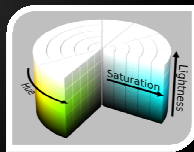
- Example: rgba(255, 0, 0, 0.5)






## HSL Colors

- ♦ Hue is a degree on the color wheel
  - ♦ 0 (or 360) is red, 120 is green, 240 is blue
- ♦ Saturation is a percentage value
  - ♦ 100% is the full color
- ♦ Lightness is also a percentage
  - ♦ 0% is dark (black)
  - ♦ 100% is light (white)
  - ♦ 50% is the average



57

## HSLA Colors

- ♦ HSLA allows a fourth value, which sets the Opacity (via the Alpha channel) of the element
- ♦ As RGBA is to RGB, HSLA is to HSL
- ♦ Supported in IE9+, Firefox 3+, Chrome, Safari, and in Opera 10+
- ♦ Example:
  - ♦ `hsla(0, 100%, 50%, 0.5)`
  - ♦ Result: 



58

## Color Values

Live Demo



## Default Browser Styles

Why Things Look Different on Different Browsers?



## Default Browser Styles

- ♦ Browsers have predefined CSS styles
  - ♦ Used when there is no CSS information or any other style information in the document
- ♦ Caution: default styles differ in browsers
  - ♦ E.g. margins, paddings and font sizes differ most often
- ♦ Usually developers reset them

```
* { margin: 0; padding: 0; }  
  
body, h1, p, ul, li { margin: 0; padding: 0; }
```

61

## CSS Cascade (Precedence)

- ♦ There are browser, user and author stylesheets with "normal" and "important" declarations
  - ♦ Browser styles (least priority)
  - ♦ Normal user styles
  - ♦ Normal author styles (external, in head, inline)
  - ♦ Important author styles
  - ♦ Important user styles (max priority)

```
a { color: red !important ; }
```

<http://www.slideshare.net/maxdesign/css-cascade-1658158>

62

## CSS Specificity

- ♦ CSS specificity is used to determine the precedence (priority) of the CSS style declarations with the same origin
  - ♦ Simple calculation: #id = 100, .class = 10, :pseudo = 10, [attr] = 10, tag = 1, \* = 0
  - ♦ Same number of points? Order matters!
- ♦ See also:
  - ♦ <http://www.smashingmagazine.com/2007/07/27/css-specificity-things-you-should-know/>
  - ♦ [http://css.maxdesign.com.au/selectutorial/advanced\\_conflict.htm](http://css.maxdesign.com.au/selectutorial/advanced_conflict.htm)

63

## CSS Rules Precedence

Live Demo





## CSS References

- ♦ The CSS documentation at WebPlatform.org:
  - ♦ <http://docs.webplatform.org/wiki/css>
- ♦ CSS documentation at Mozilla
  - ♦ <https://developer.mozilla.org/en-US/docs/CSS>
- ♦ CSS3 tutorial
  - ♦ <http://www.w3schools.com/css3/>
- ♦ A list of all CSS 2.1 properties is available at <http://www.w3.org/TR/CSS2/propidx.html>