



WEB SYSTEMS & TECHNOLOGIES

2. Cascading Style Sheets (CSS)

```
171 #content .article img.left.border {  
172     padding: 0 9px 9px 0;  
173     border-right: 1px dotted #999;  
174     border-bottom: 1px dotted #999; }  
175 #content .article blockquote {  
176     margin-left: 10px;  
177     padding-left: 10px;  
178     border-left: 3px solid #252525; }  
179 #content .article ul {  
180     padding-left: 1em;  
181     list-style-type: circle; }
```

Welcome to My Homepage

Use the menu to select different Stylesheets

[Stylesheet 1](#)

[Stylesheet 2](#)

[Stylesheet 3](#)

[Stylesheet 4](#)

[No Stylesheet](#)

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links: [Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click [here](#) to see how the page looks like with no stylesheet:

No Stylesheet.

Side-Bar

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Welcome to My Homepage

Use the menu to select different Stylesheets

Stylesheet 1

Stylesheet 2

Stylesheet 3

Stylesheet 4

No Stylesheet

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links:

[Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click [here](#) to see how the page looks like with no stylesheet:

[No Stylesheet](#).

Side-Bar

Loreum ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Loreum ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

- ▶ What is CSS?
- ▶ Styling with Cascading Stylesheets (CSS)
- ▶ Selectors and style definitions
- ▶ Linking HTML and CSS
- ▶ Fonts, Backgrounds, Borders
- ▶ The Box Model
- ▶ Alignment, Z-Index, Margin, Padding
- ▶ Positioning and Floating Elements
- ▶ Visibility, Display, Overflow

TABLE OF CONTENTS

Content

- ▶ Separate content from presentation

(HTML document)

Presentation

(CSS Document)

Title

 Lorem ipsum dolor sit amet,
 consectetuer adipiscing elit. Suspendisse
 at pede ut purus malesuada dictum.
 Donec vitae neque non magna aliquam
 dictum.

 • Vestibulum et odio et ipsum

 • accumsan accumsan. Morbi at

 • arcu vel elit ultricies porta. Proin

CSS: A NEW PHILOSOPHY

 tortor purus, luctus non, aliquam nec,
 interdum vel, mi. Sed nec quam nec odio
 lacinia molestie. Praesent augue tortor,
 convallis eget, euismod nonummy,
 lacinia ut, risus.

Bold

Italics

Indent

THE RESULTING PAGE

Title

***Lorem ipsum dolor sit amet,
 consectetuer adipiscing elit.
 Suspendisse at pede ut purus
 malesuada dictum. Donec vitae neque
 non magna aliquam dictum.***

- *Vestibulum et odio et ipsum*
- *accumsan accumsan. Morbi at*
- *arcu vel elit ultricies porta. Proin*

***Tortor purus, luctus non, aliquam nec,
interdum vel, mi. Sed nec quam nec
odio lacinia molestie. Praesent augue
tortor, convallis eget, euismod
nonummy, lacinia ut, risus.***



CSS INTRO

Styling with Cascading Stylesheets

- ▶ Cascading Style Sheets (CSS)
 - ▶ Used to describe the presentation of documents
 - ▶ Define sizes, spacing, fonts, colors, layout, etc.
 - ▶ Improve content accessibility
 - ▶ Improve flexibility
- ▶ Designed to separate presentation from content
- ▶ Due to CSS, all HTML presentation tags and attributes are deprecated, e.g. font, center, etc.

CSS INTRODUCTION

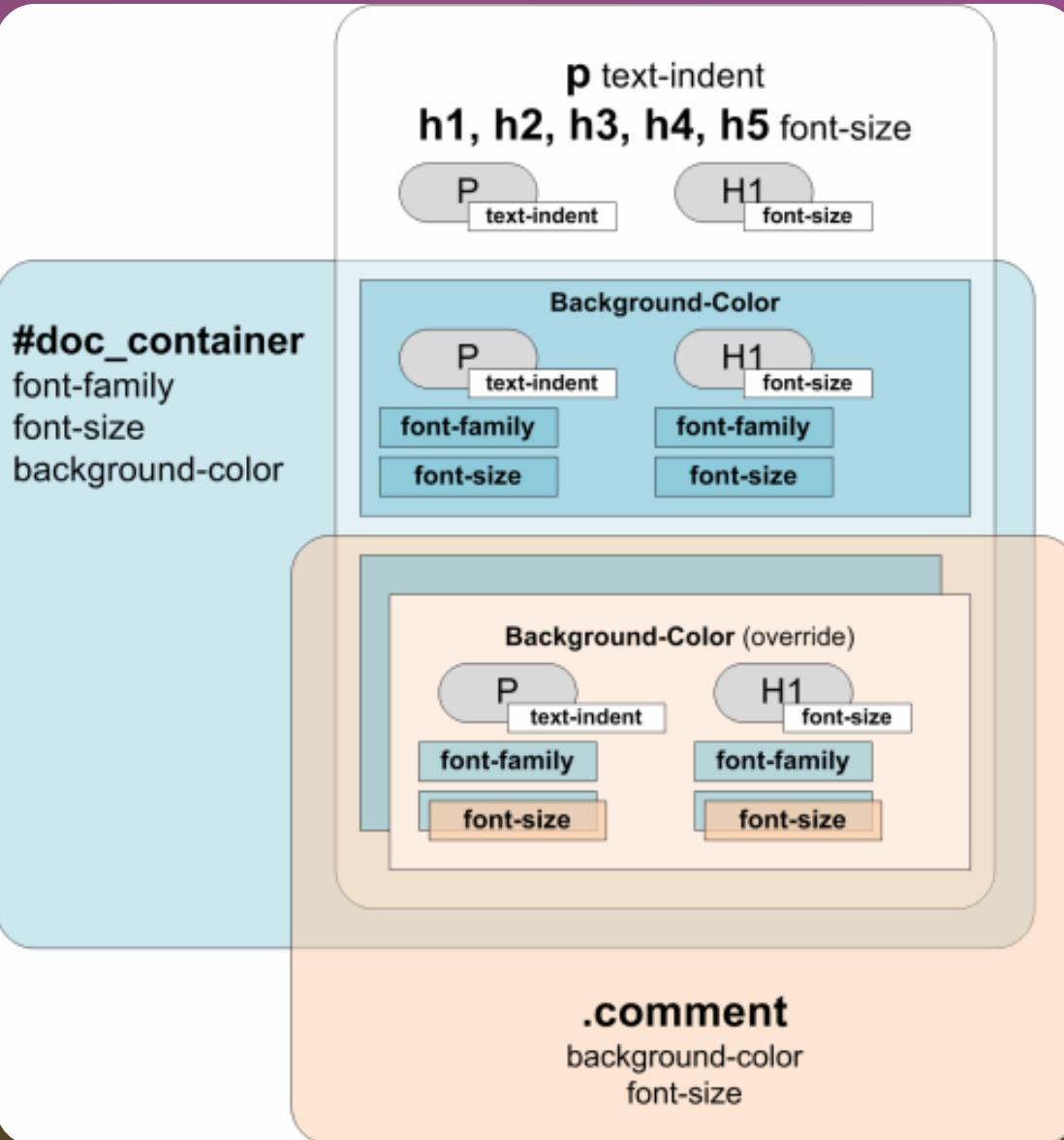
- ▶ CSS can be applied to any XML document
 - ▶ Not just to HTML / XHTML
- ▶ CSS can specify different styles for different media
 - ▶ On-screen
 - ▶ In print
 - ▶ Handheld, projection, etc.
 - ▶ ... even by voice or Braille-based reader

CSS INTRODUCTION (2)

- ▶ Priority scheme determining which style rules apply to element
 - ▶ Cascade priorities or specificity (weight) are calculated and assigned to the rules
 - ▶ Child elements in the HTML DOM tree inherit styles from their parent
 - ▶ Can override them
 - ▶ Control via `!important` rule

WHY “CASCADING”?

WHY “C”

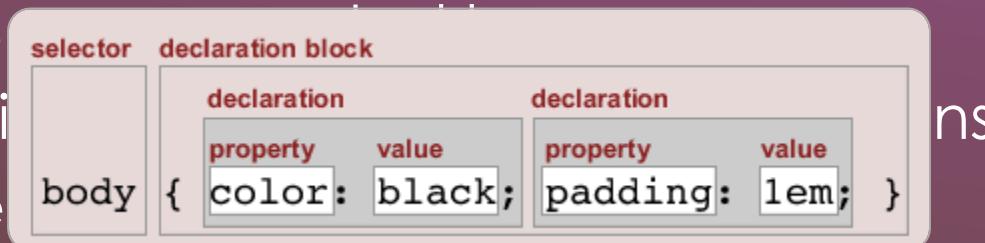


- ▶ Some CSS styles are inherited and some not
 - ▶ Text-related and list-related properties are inherited - **color, font-size, font-family, line-height, text-align, list-style**, etc
 - ▶ Box-related and positioning styles are not inherited - **width, height, border, margin, padding, position, float**, etc
 - ▶ **<a>** elements do not inherit color and text-decoration

WHY “CASCADING”? (3)

- Stylesheets consist of rules, selectors, declarations, properties and values

- Selectors
- Declarations
- Properties
colons



<http://css.maxdesign.com.au/>

STYLE SHEETS SYNTAX

```
h1,h2,h3 { color: green; font-weight: bold; }
```

- ▶ Selectors determine which element the rule applies to:
 - ▶ All elements of specific type (tag)
 - ▶ Those that match a specific attribute (id, class)
 - ▶ Elements may be matched depending on how they are nested in the document tree (HTML)
- ▶ Examples:

SELECTORS

```
.header a { color: green }
```

```
#menu>li { padding-top: 8px }
```

- ▶ Three primary kinds of selectors:

- ▶ By tag (type selector):

```
h1 { font-family: verdana,sans-serif; }
```

- ▶ By element id:

```
#element_id { color: #ff0000; }
```

- ▶ By element class name (only for HTML):

```
.myClass {border: 1px solid red}
```

- ▶ Selectors can be combined with commas
h1, .link, #top-link {font-weight: bold}

This will match `<h1>` tags, elements with class `link`, and element with id `top-link`

SELECTORS (3)

- ▶ Pseudo-classes define state
 - ▶ :hover, :visited, :active , :lang(value); :target
- ▶ Pseudo-elements define element "parts" or are used to generate content

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

- Match relative to element placement:

```
p a {text-decoration: underline}
```

This will match all <a> tags that are inside of <p>

- * – universal selector (avoid or use with care!):

```
p * {color: black}
```

This will match all descendants of <p> element

- + selector – used to match “next sibling”:

```
img + .link {float:right}
```

This will match all siblings with class name link that appear immediately after tag

SELECTORS (5)

- > selector – matches direct child nodes:

```
p > .error {font-size: 8px}
```

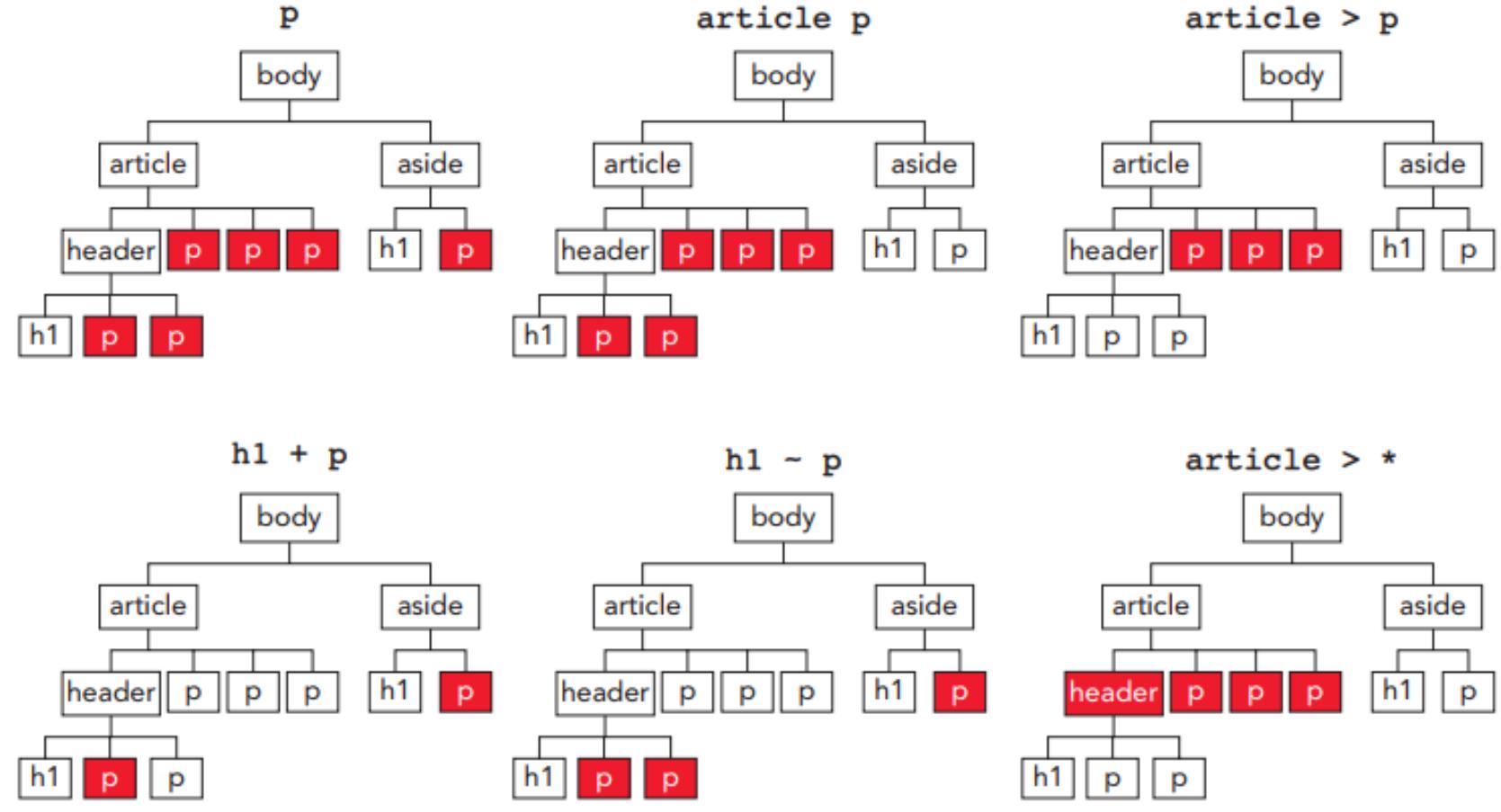
This will match all elements with class error, direct children of `<p>` tag

- [] – matches tag attributes by regular expression:

```
img[alt~="logo"] {border: none}
```

This will match all `` tags with `alt` attribute containing the word logo

- `.class1.class2` (no space) - matches elements with both (all) classes applied at the same time



VALUES IN THE CSS RULES

- ▶ Colors are set in RGB format (decimal or hex):
 - ▶ Example: #a0a6aa = rgb(160, 166, 170)
 - ▶ Predefined color aliases exist: black, blue, etc.
- ▶ Numeric values are specified in:
 - ▶ Pixels, ems, e.g. 12px , 1.4em
 - ▶ Points, inches, centimeters, millimeters
 - ▶ E.g. 10pt , 1in, 1cm, 1mm
 - ▶ Percentages, e.g. 50%
 - ▶ Percentage of what?...
 - ▶ Zero can be used with no unit: border: 0;

- ▶ Browsers have default 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 and usually developers reset them

* { margin: 0; padding: 0; }
DEFAULT BROWSER STYLES

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

LINKING HTML AND CSS

- ▶ HTML (content) and CSS (presentation) can be linked in three ways:
 - ▶ Inline: the CSS rules in the `style` attribute
 - ▶ No selectors are needed
 - ▶ Embedded: in the `<head>` in a `<style>` tag
 - ▶ External: CSS rules in separate file (best)
 - ▶ Usually a file with `.css` extension
 - ▶ Linked via `<link rel="stylesheet" href=...>` tag or`@import` directive in embedded CSS block

- ▶ Using external files is highly recommended
 - ▶ Simplifies the HTML document
 - ▶ Improves page load speed as the CSS file is cached

LINKING HTML AND CSS (2)

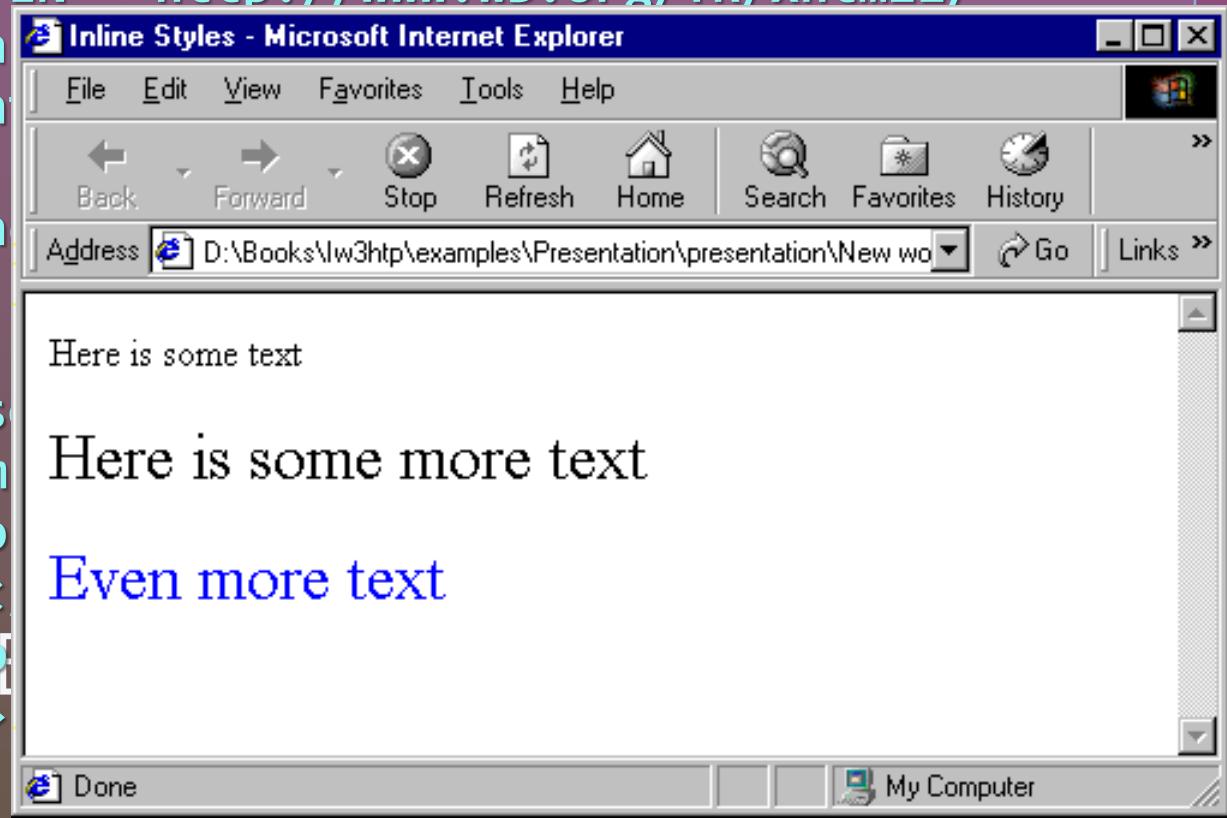
inline-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Inline Styles</title>
</head>
<body>
    <p>Here is some text</p>
    <!--Separate multiple styles with a semicolon-->
    <p style="font-size: 20pt">Here is some more text</p>
    <p style="font-size: 20pt; color: #0000FF">Even more text</p>
</body>
</html>
```

INLINE STYLES: EXAMPLE

inline-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/  
DTD/xhtml1-transitional.dtd"  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Inline Styles</title>  
</head>  
<body>  
    <p>Here is some text</p>  
    <!--Separate margin from content-->  
    <p style="font-size: 2em; color: blue;">  
        more text</p>  
    <!--Color from style-->  
    <p style="color: #0000FF;">  
        Even more text</p>  
    </body>  
</html>
```

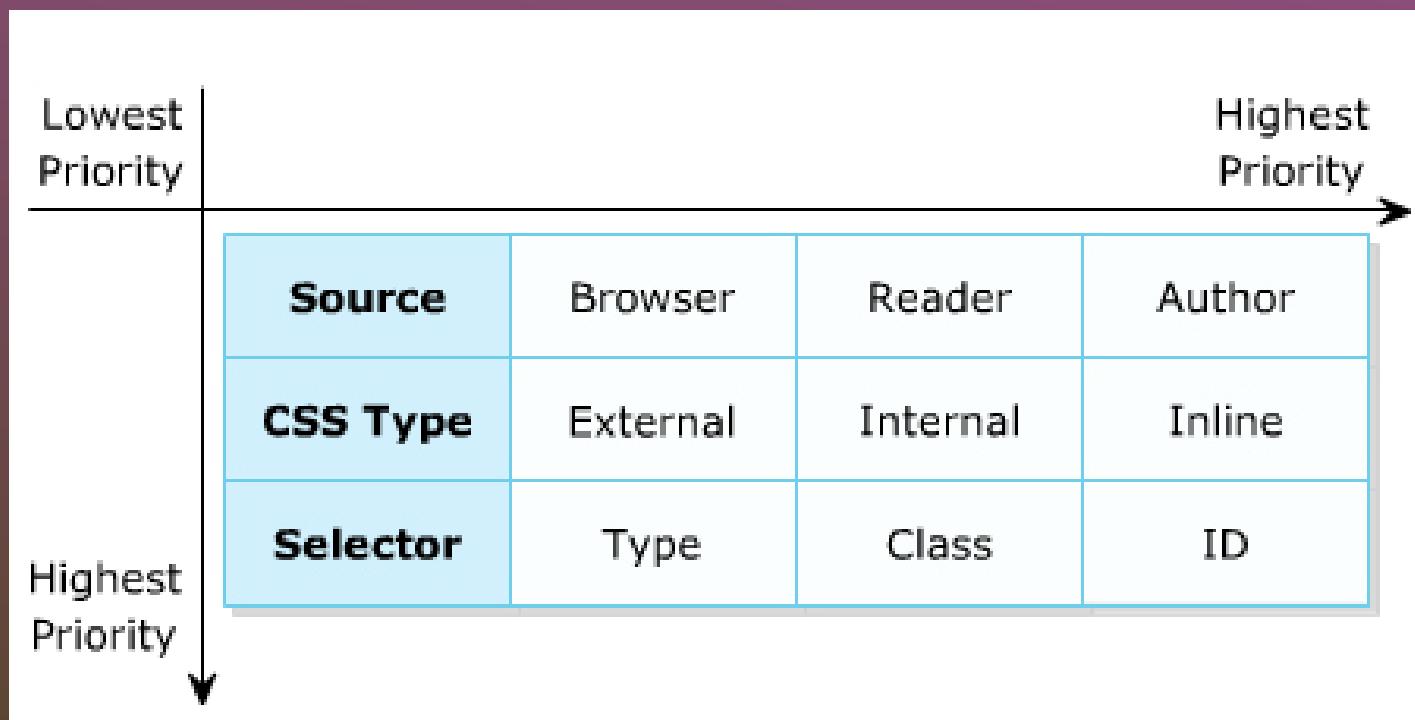


- ▶ There are browsers, user and author stylesheets with "normal" and "important" declarations
 - ▶ Browser styles (least priority)
 - ▶ Normal user styles
 - ▶ Normal author styles (external, in head, inline)
 - ▶ Important author styles
 - ▶ Important user styles (max priority)

CSS CASCADE (PRECEDENCE)

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

26



- ▶ CSS specificity is used to determine the precedence of CSS style declarations with the same origin. Selectors are what matters
 - ▶ 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

CSS SPECIFICITY

EMBEDDED STYLES

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

- ▶ Embedded in the HTML in the `<style>` tag:
 - ▶ The `<style>` tag is placed in the `<head>` section of the document
 - ▶ `type` attribute specifies the MIME type
 - ▶ MIME describes the format of the content
 - ▶ Other MIME types include `text/html`, `image/gif`, `text/javascript` ...
- ▶ Used for document-specific styles

EMBEDDED STYLES: EXAMPLE

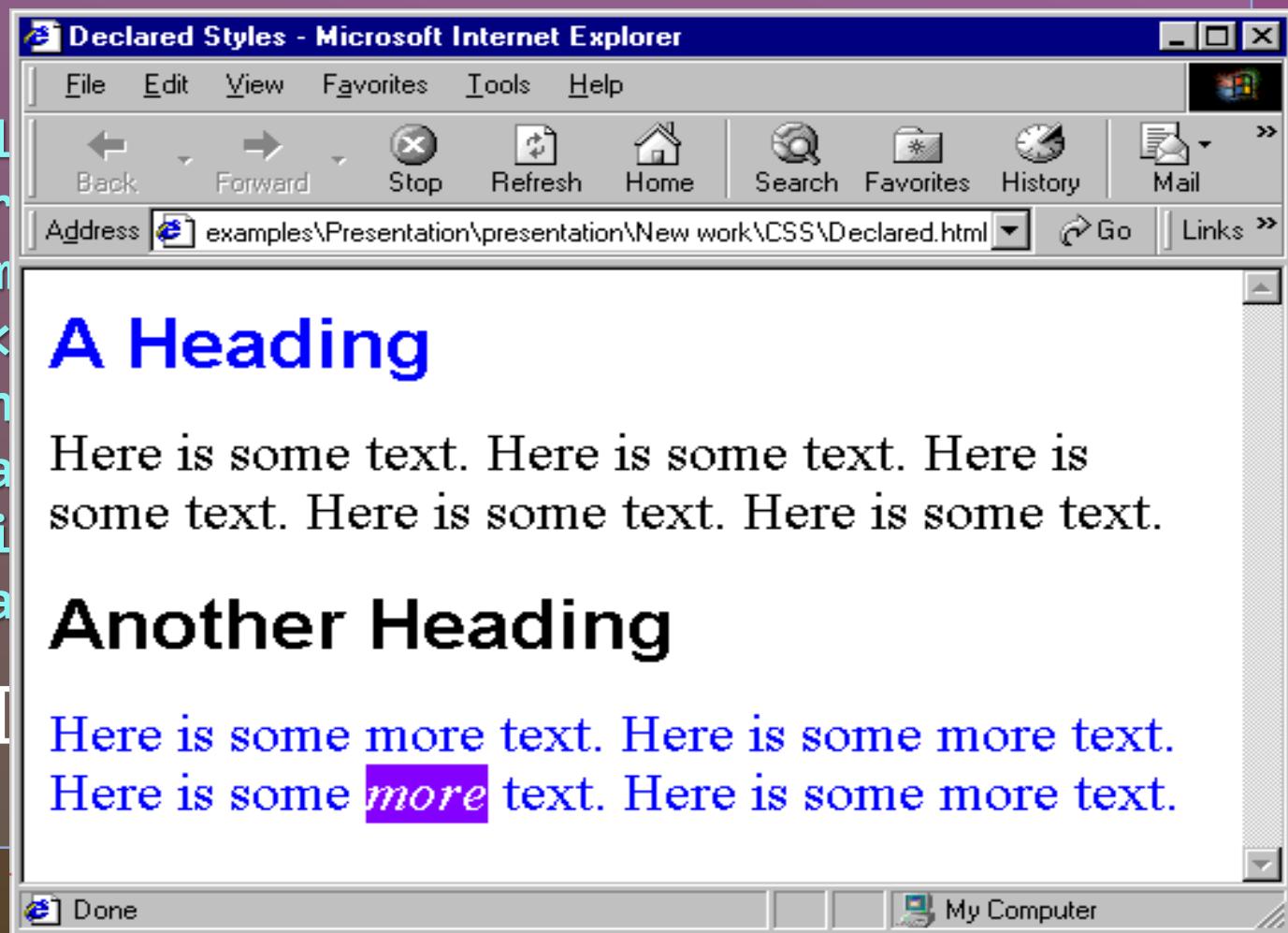
embedded-stylesheets.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Style Sheets</title>  
    <style type="text/css">  
        em {background-color:#8000FF; color:white}  
        h1 {font-family:Arial, sans-serif}  
        p {font-size:18pt}  
        .blue {color:blue}  
    </style>  
</head>
```

```
...
<body>
  <h1 class="blue">A Heading</h1>
  <p>Here is some text. Here is some text. Here
  is some text. Here is some text. Here is some
  text.</p>
  <h1>Another Heading</h1>
  <p class="blue">Here is some more text.
  Here is some more text.</p>
  <p class="blue">Here is some <em>more</em>
  text. Here is some more text.</p>
</body>
</html>
```

EMBEDDED STYLES: EXAMPLE (2)

```
...
<body>
  <h1 class="big">A Heading</h1>
  <p>Here is some text.</p>
  <h1>Another Heading</h1>
  <p>Here is some more text. Here is some more text.
</body>
</html>
```



EXTERNAL CSS STYLES

- ▶ External linking
 - ▶ Separate pages can all use a shared style sheet
 - ▶ Only modify a single file to change the styles across your entire Web site (see <http://www.csszengarden.com/>)
- ▶ link tag (with a rel attribute)
 - ▶ Specifies a relationship between current document and another document

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

33

- ▶ link elements should be in the <head>

@import

- ▶ Another way to link external CSS files

```
<style type="text/css">
  @import url("styles.css");
  /* same as */
  @import "styles.css";
</style>
```

- ▶ Ancient browsers do not recognize @import

EXTERNAL CSS STYLES (2)

Use @import in an external CSS file to workaround the IE 32 CSS file limit

EXTERNAL STYLES: EXAMPLE

styles.css

```
/* CSS Document */

a { text-decoration: none }

a:hover { text-decoration: underline;
           color: red;
           background-color: #CCFFCC }

li em { color: red;
         font-weight: bold }

ul { margin-left: 2cm }

ul ul { text-decoration: underline;
          margin-left: .5cm }
```

external-styles.html

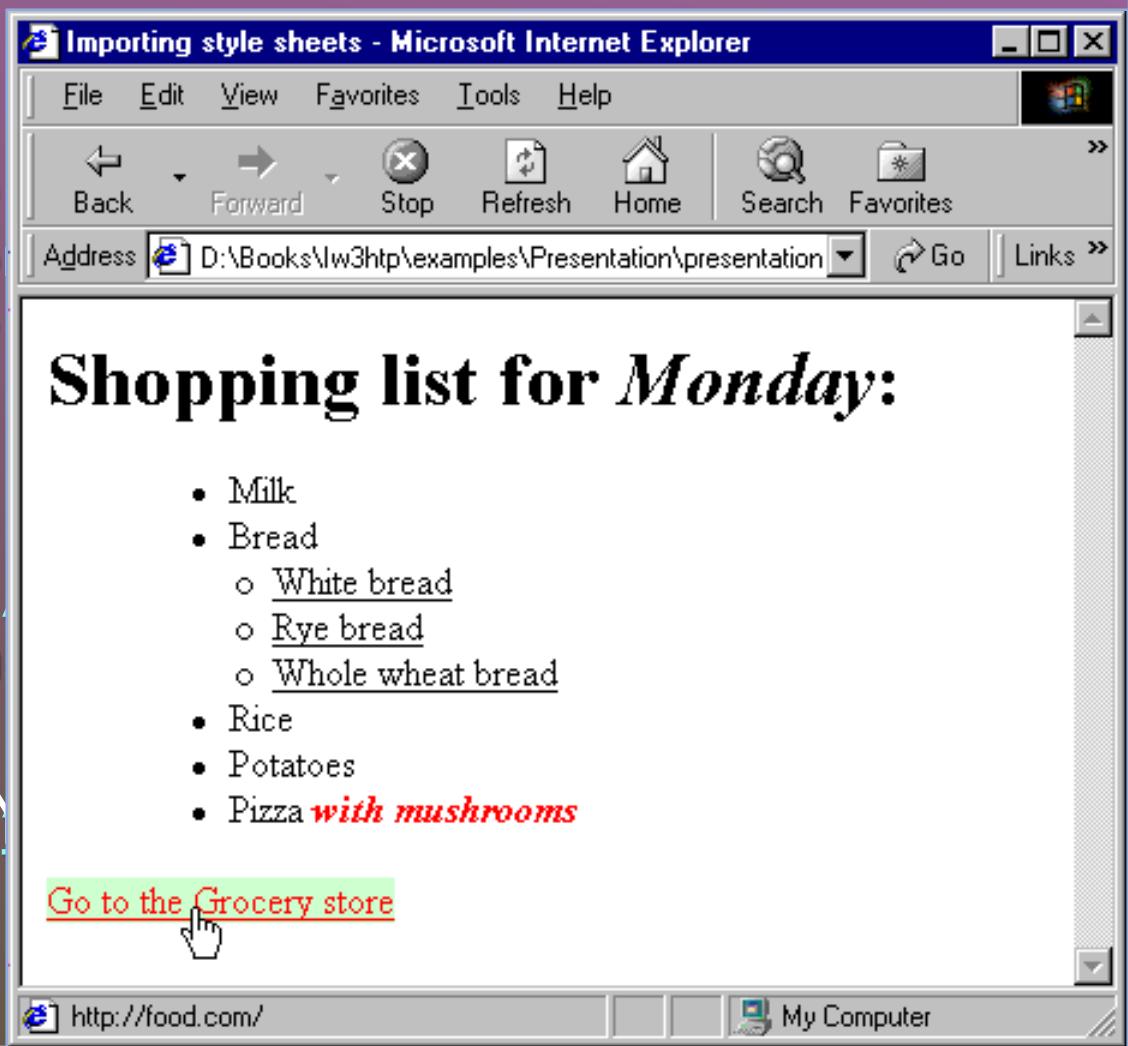
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
  <title>Importing style sheets</title>  
  <link type="text/css" rel="stylesheet"  
    href="styles.css" />  
</head>  
<body>  
EXTERNAL STYLES: EXAMPLE (2)  
  <h1>Shopping list for <em>Monday</em>:</h1>  
  <li>Milk</li>  
  ...
```

```
...
<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>
```

EXTERNAL STYLES: EXAMPLE (3)

```
...
<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>
```

EXTERNAL ST



- ▶ **color** – specifies the color of the text
- ▶ **font-size** – size of font: **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**, **smaller**, **larger** or numeric value
- ▶ **font-family** – comma separated font names
 - ▶ Example: **verdana**, **sans-serif**, etc.
 - ▶ The browser loads the first one that is available
 - ▶ There should always be at least one generic font
- ▶ **font-weight** can be **normal**, **bold**, **bolder**, **lighter** or a number in range [100 ... 900]

TEXT-RELATED CSS PROPERTIES

- ▶ **font-style** – styles the font
 - ▶ Values: `normal`, `italic`, `oblique`
 - ▶ **text-decoration** – decorates the text
 - ▶ Values: `none`, `underline`, `line-through`, `overline`, `blink`
 - ▶ **text-align** – defines the alignment of text or other content
 - ▶ Values: `left`, `right`, `center`, `justify`
 - ▶ **line-height**: defines the height of line
- ## CSS RULES FOR FONTS (2)

- ▶ font

- ▶ Shorthand rule for setting multiple font properties at the same time

is equal to writing this:

font:italic normal bold 12px/16px verdana

```
font-style: italic;  
font-variant: normal;  
font-weight: bold;  
font-size: 12px;  
line-height: 16px;  
font-family: verdana;
```

SHORTHAND FONT PROPERTY

TEXT PROPERTIES

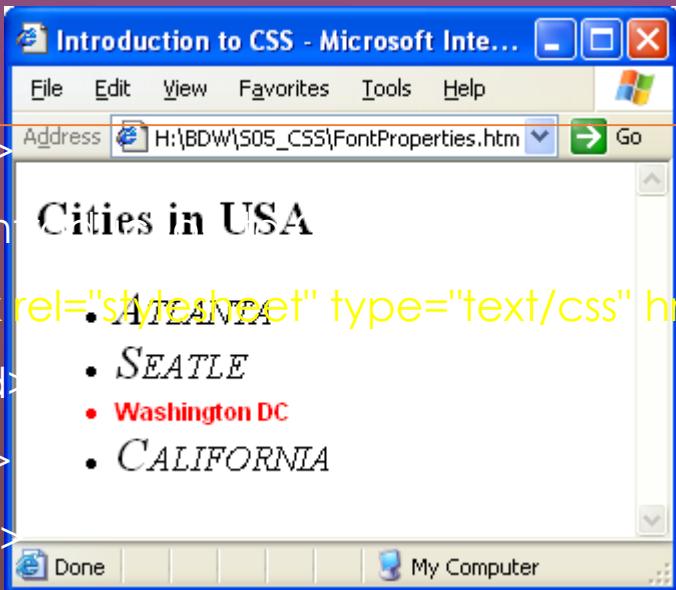
```
<html><head>
<title>Introduction to CSS - Microsoft Inte...
<head>
<title>Introduction to CSS - Microsoft Inte...
<link rel="stylesheet" type="text/css" href="myCSS2.css">
<body>
<div>
<h2>Notices</h2>
<p class="old">Old campus: Melbourne city.
<p>New campus: Sai gon, Vietnam
</div>
</body>
</html>
```

```
div {
    color: blue;
    text-align: left;
    text-indent: 2em;
    word-spacing: 2mm;
}

.old {
    color: gray;
    text-decoration: line-through;
}
```

```
<html>
<head>
<title>Introduction to CSS - Microsoft Internet Explorer</title>
<link rel="stylesheet" type="text/css" href="myCSS3.css"/>
</head>
<body>
<div>
<h2>Cities in USA</h2>
<ul>
<li>Atlanta</li>
<li>Seattle</li>
- Washington DC

<li>California</li>
</ul>
</div>
</body>
</html>
```



```
UL{
    font-family: "Times New Roman";
    font-size: large;
    font-style: italic;
    font-variant: small-caps;
}

#shorthand{
    font: bold 12px Arial;
    color: red;
}
```

- ▶ `background-image`

`background-image:url("back.gif");`

- ▶ URL of image to be used as background, e.g.:

- ▶ `background-color`

- ▶ Using color and image at the same time

- ▶ `background-repeat`

- ▶ repeat-x, repeat-y, repeat, no-repeat

- ▶ `background-attachment`

BACKGROUNDS

- ▶ fixed / scroll

- ▶ **background-position**: specifies vertical and horizontal position of the background image
 - ▶ Vertical position: top, center, bottom
 - ▶ Horizontal position: left, center, right
 - ▶ Both can be specified in percentage or other numerical values
 - ▶ Examples:

BACKGROUNDS (2)

background-position: top left;

background-position: -5px 5%;

BACKGROUND SHORTHAND PROPERTY

- **background:** shorthand rule for setting background properties at the same time:

```
background: #FFF0C0 url("back.gif") no-repeat  
fixed top;
```

is equal to writing:

```
background-color: #FFF0C0;  
background-image: url("back.gif");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: top;
```

- Some browsers will not apply BOTH color and image for background if using shorthand rule

```
background-image: url(Greeting.jpg);
```

BACKGROUND PROPERTIES

```
background-repeat: repeat;
```

```
background-attachment: fixed;
```

```
}
```

```
div {
```

```
background-color: #FFFF00;
```

```
text-align: right;
```

```
}
```

```
<html>
```

```
<head>
```

```
    <title>Welcome</title>
```

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

```
</head>
```

```
<body>
```

```
    <h2>Welcome to IUH.</h2>
```

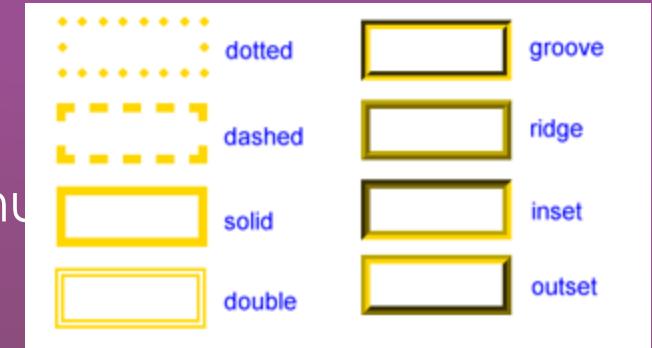
```
    <div><p>Dream of Innovation </p></div>
```

```
</body>
```



- ▶ Background images allow you to save many image tags from the HTML
 - ▶ Leads to less code
 - ▶ More content-oriented approach
- ▶ All images that are not part of the page content (and are used only for "beautification") should be moved to the CSS

BACKGROUND-IMAGE OR
?



- ▶ `border-width`: thin, medium, thick or numeric value (e.g. 10px)
- ▶ `border-color`: color alias or RGB value
- ▶ `border-style`: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
- ▶ Each property can be defined separately for left, top, bottom and right
 - ▶ `border-top-style`, `border-left-color`, ...

BORDERS

- ▶ border: shorthand rule for setting border properties at once:

```
border: 1px solid red
```

is equal to writing:

```
border-width:1px;  
border-style:solid;  
border-color:red;
```

BORDER SHORTHAND PROPERTY
Specify different borders for the sides via shorthand rules:
border-top, border-left, border-right, border-bottom

- ▶ When to avoid border:0

- ▶ **width** – defines numerical value for the width of element, e.g. 200px
- ▶ **height** – defines numerical value for the height of element, e.g. 100px
 - ▶ By default the height of an element is defined by its content
 - ▶ Inline elements do not apply height, unless you change their **display** style.

WIDTH AND HEIGHT

```
<head>
```

```
<title>Welcome</title>
```

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

```
</head>
```

CSS style border properties

```
<h2>CSS style border properties</h2>
```

```
<div class="tip">
```

1. Style sets the style appearance of the border.
2. Width sets the thickness of the elements border
3. Color sets the color of the border.

```
    <li><b>Style</b> sets the style appearance  
of the border.</li>
```

```
    <li><b>Width </b>sets the thickness of the  
elements border</li>
```

```
    <li><b>Color</b> sets the color of the  
border.</li>
```

```
</ol>
```

```
</div>
```

```
</body>
```

.tip {

```
background-color:  
lightcyan;  
border-top-color: #0000FF;  
border-right-color: #FF0000;  
border-bottom-color:  
#FF00FF;  
border-left-color: #FFFF00;  
border-style: double;  
border-width: 20px;
```

}

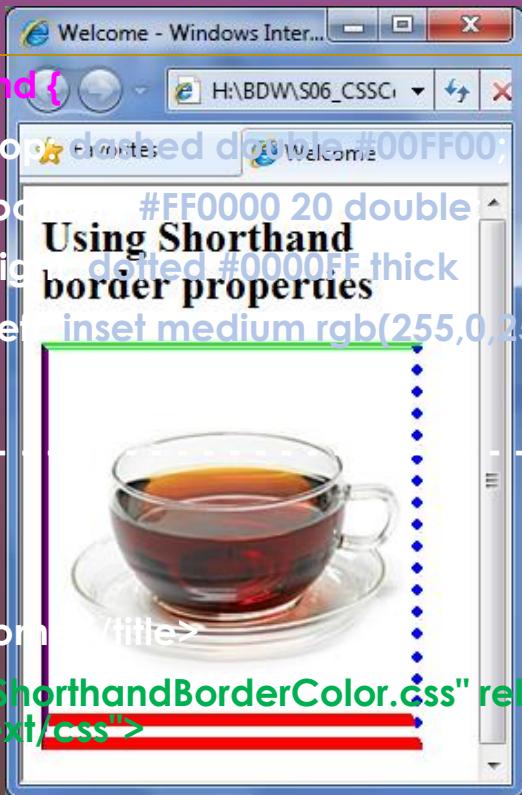
```
.myshorthand {  
border-top: dashed double #00FF00;  
border-bottom: #FF0000 20px double;  
border-right: dotted #0000FF thick;  
border-left: inset medium rgb(255,0,255);  
}
```

```
<html>  
<head>  
<title>Welcome</title>  
<link href="ShorthandBorderColor.css" rel="stylesheet"  
      type="text/css">  
</head>
```

```
<body>  


## SHORTHAND BORDER PROPERTIES

  
  
</body>  
</body>  
</html>
```



- ▶ **margin** and **padding** define the spacing around the element
 - ▶ Numerical value, e.g. 10px or -5px
 - ▶ Can be defined for each of the four sides separately - `margin-top`, `padding-left`, ...
 - ▶ **margin** is the spacing outside of the border
 - ▶ **padding** is the spacing between the border and the content
 - ▶ What are collapsing margins?

MARGIN AND PADDING

- ▶ `margin: 5px;`
 - ▶ Sets all four sides to have margin of 5 px;
- ▶ `margin: 10px 20px;`
 - ▶ top and bottom to 10px, left and right to 20px;
- ▶ `margin: 5px 3px 8px;`
 - ▶ top 5px, left/right 3px, bottom 8px
- ▶ `margin: 1px 3px 5px 7px;`
 - ▶ top, right, bottom, left (clockwise from top)
- ▶ Same for padding

MARGIN AND PADDING: SHORT RULES

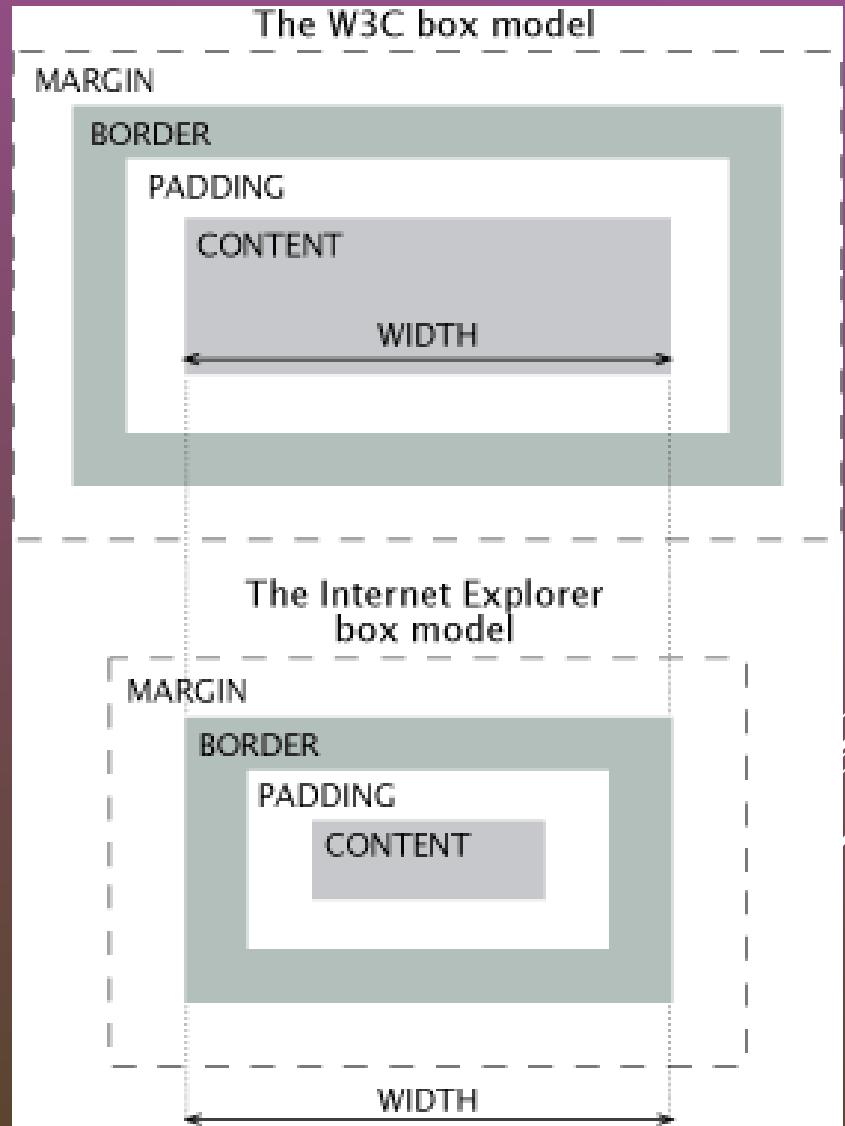
Margin

Border

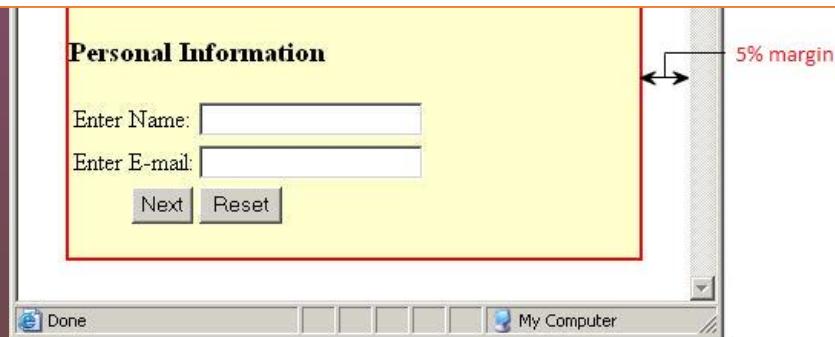
Padding

Content

- When using quirks mode (pages with no DOCTYPE or with a HTML 4 Transitional DOCTYPE), Internet Explorer violates the box model standard

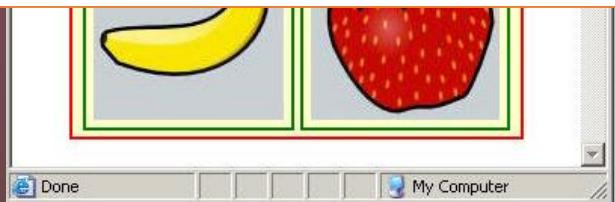


```
.format  
{  
    background-color: #FFFFCC;  
    border: solid thin red;  
    margin-top: 5%;  
    margin-left: 5%;  
    margin-right: 5%;  
}
```



MARGIN

```
BODY
{
    margin-left: 10%;
    margin-right: 10%;
}
.fruits
{
    padding-left: 5px;
    padding-top: 5px;
    padding-right: 5px;
    padding-bottom: 5px;
    border: solid thin green;
}
```



PADDING

- ▶ **position**: defines the positioning of the element in the page content flow
- ▶ The value is one of:
 - ▶ **static** (default)
 - ▶ **relative** – relative position according to where the element would appear with static position
 - ▶ **absolute** – position according to the innermost positioned parent element
 - ▶ **fixed** – same as absolute, but ignores page scrolling

POSITIONING

- ▶ Margin VS relative positioning
- ▶ Fixed and absolutely positioned elements do not influence the page normal flow and usually stay on top of other elements
 - ▶ Their position and size is ignored when calculating the size of parent element or position of surrounding elements
 - ▶ Overlaid according to their z-index
 - ▶ Inline fixed or absolutely positioned elements can apply height like block-level elements

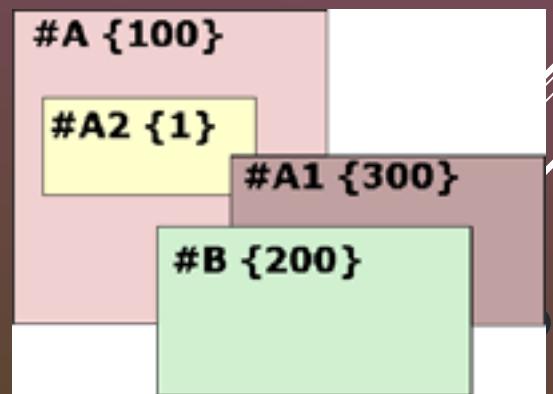
POSITIONING (2)

- ▶ **top, left, bottom, right**: specifies offset of absolute/fixed/relative positioned element as numerical values
- ▶ **z-index** : specifies the stack level of positioned elements
 - ▶ Understanding stacking context

Each positioned element creates a stacking context.

POSITIONING (3)

Elements in different stacking contexts are overlapped according to the stacking order of their containers. For example, there is no way for #A1 and #A2 (children of #A) to be placed over #B without increasing the z-index of #A.



- ▶ `vertical-align`: sets the vertical-alignment of an inline element, according to the line height
 - ▶ Values: `baseline`, `sub`, `super`, `top`, `text-top`, `middle`, `bottom`, `text-bottom` or numeric
- ◆ Also used for content of table cells (which apply `middle` alignment by default)

INLINE ELEMENT POSITIONING

Positioning

```
#pos1
{
    position:relative;
    left:-50px;
    top:30px;
}
#pos2
{
    position:absolute;
    left:10px;
    top:50px;
}
```

html code

```
<body>
<p>This is the text that you would like the image position...
</p>
```

```
<p>This is the text that you would like the image position...
</p>
</body>
```

This is the text that you would like the image position...



This is the text that you would like the image position...



Positioning

- The “clip” property specifies the shape of the element’s content that is displayed on the Web page.
- It is widely used when the content is too large to fit in the specified space.

```
.clip  
{  
    position:absolute;  
    clip:rect(0px, 120px, 200px, 0px);  
}
```

Before



After



Positioning

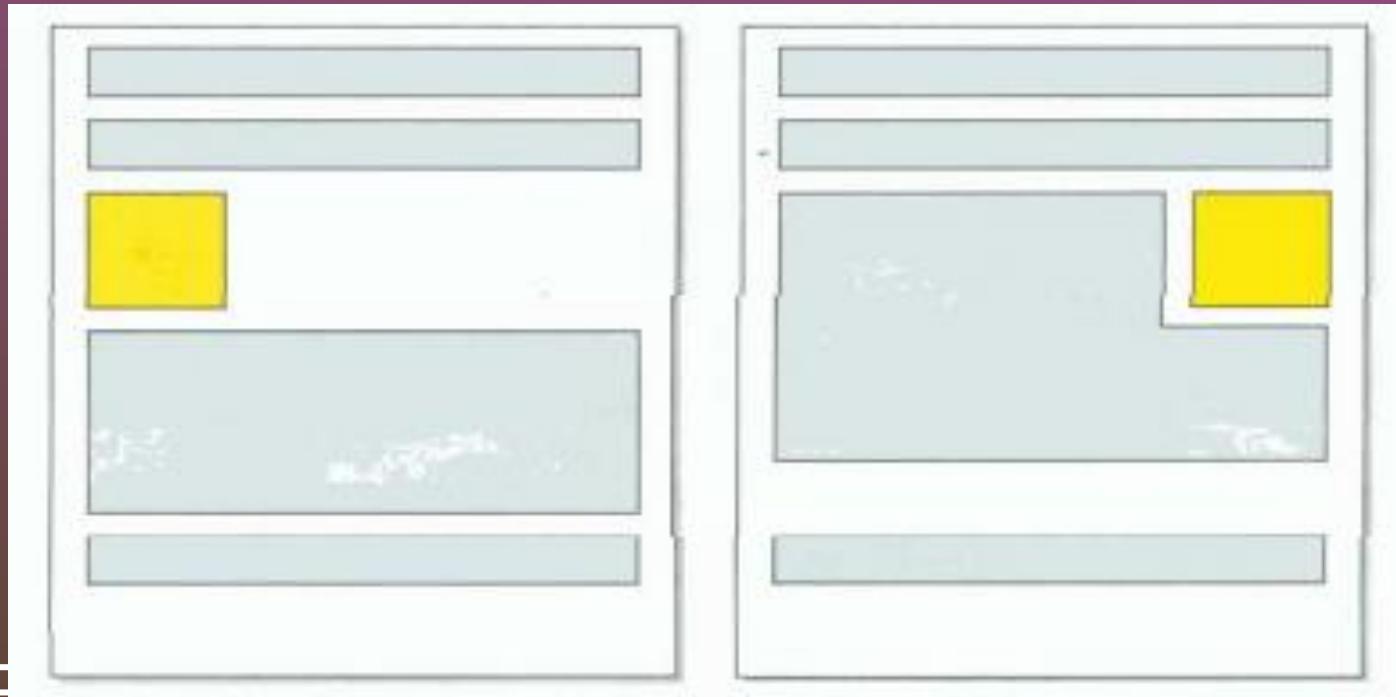
- The “z-index” property specifies the stack order of an element.
- An element with greater stack order is always in front of an element with a lower stack order.
- Note:** z-index only works if position: absolute; is used.

```
.showa  
{  
    position:absolute;  
    left:50px;  
    top:20px;  
    z-index:2;  
}  
.showb  
{  
    position:absolute;  
    left:0px;  
    top:0px;  
    z-index:1;  
}
```



- ▶ **float**: the element “floats” to one side
 - ▶ **left**: places the element on the left and following content on the right
 - ▶ **right**: places the element on the right and following content on the left
 - ▶ floated elements should come before the content that will wrap around them in the code
 - ▶ margins of floated elements do not collapse
 - ▶ floated inline elements can apply height

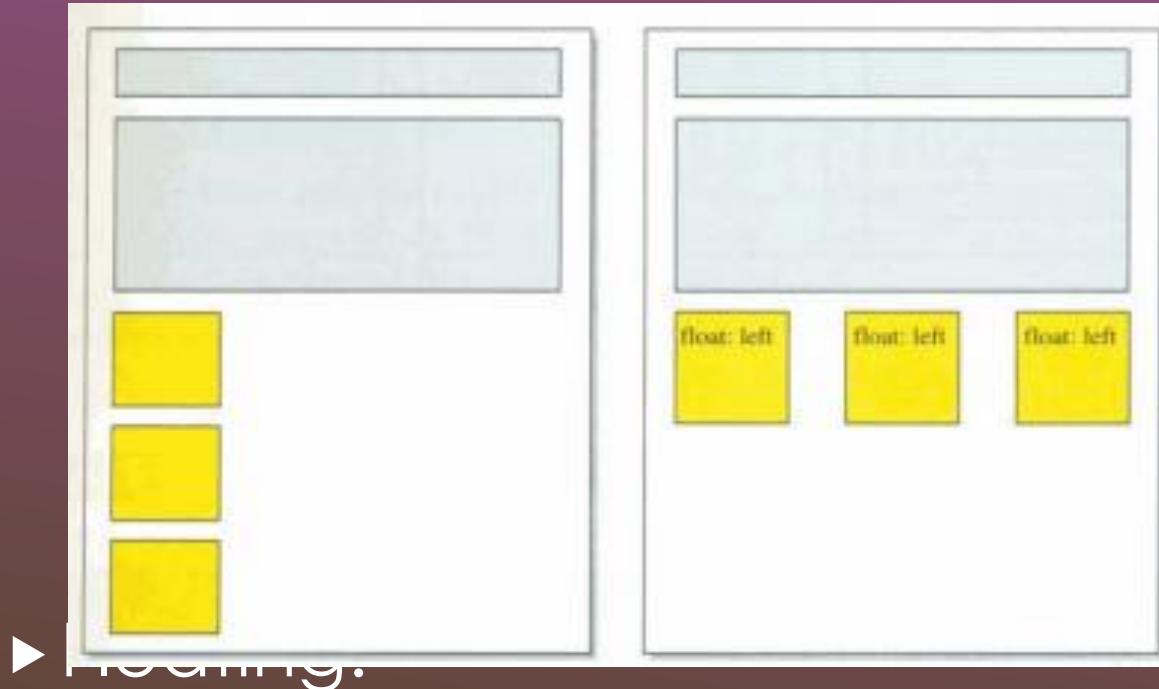
FLOAT



FLOAT

- ◆ Floating: normal
- Floating: right

Float

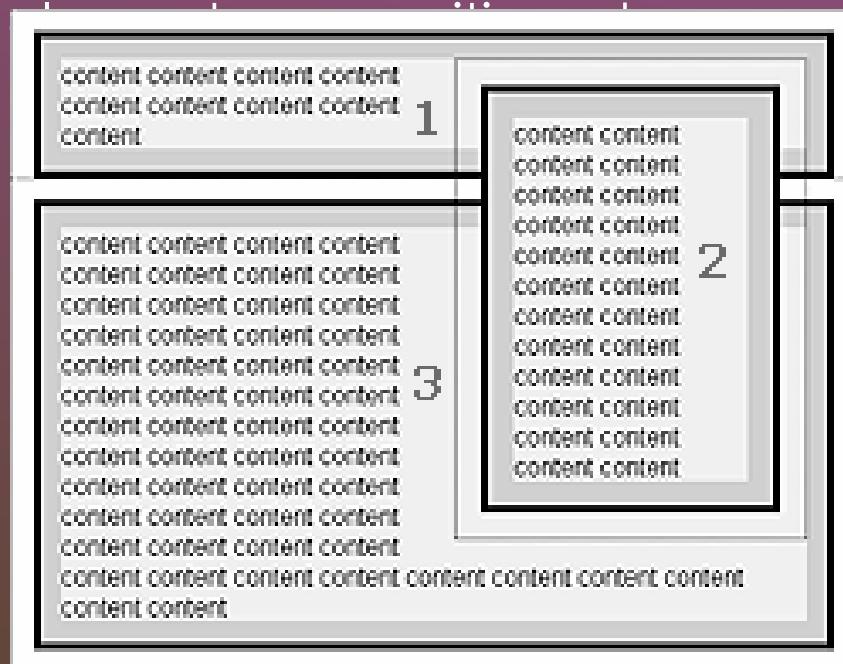


▶ Meaning:

normal (3
elements)

◆ **Floating: left (3
elements)**

► How floated



FLOAT (2)

THE STUFF SHOP

The stuff Shop is your online home for the buying, selling, and trading of used merchandise and unique collectibles.

[Books](#)

[Clothes](#)

[Electronics](#)

[Furniture](#)

[Jewelry](#)

[Music and Videos](#)

[Sporting Goods](#)

Click a link on the left to browse the store.



► clear

- ▶ Sets the sides of the element where other floating elements are NOT allowed
- ▶ Used to "drop" elements below floated ones or expand a container, which contains only floated children
- ▶ Possible values: `left`, `right`, `both`

► Clearing floats

- ▶ additional element (`<div>`) with a clear style

CLEAR

► Clearing floats (continued)

- ▶ `:after { content: ""; display: block; clear: both; height: 0; }`
- ▶ Triggering hasLayout in IE expands a container of floated elements
 - ▶ `display: inline-block;`
 - ▶ `zoom: 1;`

CLEAR (2)

- ▶ **opacity**: specifies the opacity of the element
 - ▶ Floating point number from 0 to 1
 - ▶ For old Mozilla browsers use `-moz-opacity`
 - ▶ For IE use `filter:alpha(opacity=value)` where value is from 0 to 100; also, "binary and script behaviors" must be enabled and `hasLayout` must be triggered, e.g. with `zoom:1`

OPACITY

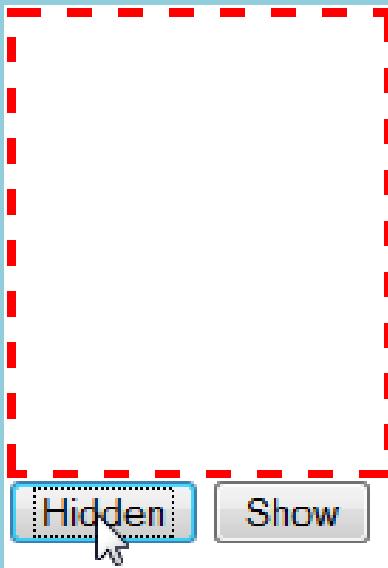
► **visibility**

- ▶ Determines whether the element is visible
- ▶ **hidden**: element is not rendered, but still occupies place on the page (similar to **opacity:0**)
- ▶ **visible**: element is rendered normally

VISIBILITY

Visibility

```
.frame  
{  
    border: dashed medium red;  
    width:130px;  
}  
  
html file  
<body>  
<div class="frame">  
<p></p>  
</div>  
<input type="button"  
       value="Hidden"  
       onclick="myImg.style.visibility='hidden'">  
  
<input type="button"  
       value="Show"  
       onclick="myImg.style.visibility='visible'">  
</body>
```



- ▶ **display**: controls the display of the element and the way it is rendered and if breaks should be placed before and after the element
 - ▶ **inline**: no breaks are placed before and after (**** is an inline element)
 - ▶ **block**: breaks are placed before AND after the element (**<div>** is a block element)

DISPLAY

- ▶ **display:** controls the display of the element and the way it is rendered and if breaks should be placed before and after the element
 - ▶ **none:** element is hidden and its dimensions are not used to calculate the surrounding elements rendering (differs from **visibility: hidden!**)
 - ▶ There are some more possible values, but not all browsers support them
 - ▶ Specific displays like **table-cell** and **table-row**

DISPLAY (2)

- ▶ **overflow**: defines the behavior of element when content needs more space than you have specified by the size properties or for other reasons. Values:
 - ▶ **visible** (default) – content spills out of the element
 - ▶ **auto** - show scrollbars if needed
 - ▶ **scroll** – always show scrollbars
- ▶ **OVERFLOW**
- ▶ **hidden** – any content that cannot fit is clipped

Pseudo Classes

- Pseudo classes allow apply different styles to elements such as button, hyperlinks and so on.

```
Selector : pseudo-class { property: value; }
```

Or

```
Selector.class : pseudo-class { property : value; }
```

Where:

- Selector: is an element name
- Pseudo-class: is one of the states of element: active, hover, link, visited.
- Property: in any CSS property such as color, border, font, ...

Pseudo Classes

```
a:link  
{  
    color:#FF0000;  
}  
a:visited  
{  
    color:#00FF00;  
}  
a:hover  
{  
    color:#FF00FF;  
}  
a:active  
{  
    color:#0000FF;  
}
```

```
a:link  
{  
    color:#FF0000;  
}  
a:visited  
{  
    color:#00FF00;  
}  
a:hover  
{  
    color:#FF00FF;  
}  
a:active  
{  
    color:#0000FF;  
}
```

```
a:link  
{  
    text-decoration:none;  
    background-color:#B2FF99;  
}  
a:visited  
{  
    text-decoration:none;  
    background-color:#FFFF85;  
}  
a:hover  
{  
    text-decoration:underline;  
    background-color:#FF704D;  
}  
a:active  
{  
    text-decoration:underline;  
    background-color:#FF704D;  
}
```

Pseudo Classes

```
p:first-line  
{  
    font-family: tahoma;  
    font-weight: bold;  
    background-color: #FFFFCC;  
}
```

E-Commerce

E-commerce (Electronic commerce) is defined as the sale and purchase of products over the Internet. E-mail, accounting, shipment information, and enterprise information reporting are the some common applications of e-commerce.

```
p:first-letter  
{  
    font-family: fantasy;  
    font-size: xx-large;  
    font-weight: bold;  
}
```

Physical Chemistry

Physical chemistry is a branch of chemistry which deals with the physical properties of chemicals.

Table Properties

```
table
{
background-color: #f0f0f0;
border: 1px solid #C1DAD7;
border-collapse: collapse;
table-layout: fixed;
}
```

```
th
{
font: bold 11px "Trebuchet MS", Verdana, Arial, Helvetica, sans-serif;
color: #4f6b72;
border-right: 1px solid #C1DAD7;
border-bottom: 1px solid #C1DAD7;
border-top: 1px solid #C1DAD7;
letter-spacing: 2px;
text-transform: uppercase;
text-align: left;
padding: 6px 6px 6px 12px;
background: #CAE8EA url(images/bg_header.jpg) no-repeat;
}
caption
{
padding: 0 0 5px 0;
```

Login Info:

User Name:

Password:

C

A

CONFIGURATIONS	DUAL 1.8GHZ	DUAL 2GHZ	DUAL 2.5GHZ
MODEL	M9454LL/A	M9455LL/A	M9457LL/A
G5 PROCESSOR	Dual 1.8GHz PowerPC G5	Dual 2GHz PowerPC G5	Dual 2.5GHz PowerPC G5
FRONTSIDE BUS	900MHz per processor	1GHz per processor	1.25GHz per processor
LEVEL2 CACHE	512K per processor	512K per processor	512K per processor

List Properties

- List properties allow set the type of the list-item marker, apply image to the marker and position the marker

List-style-type: {none, circle, square, disc}

List-style-image: url (images/bullet.gif)}

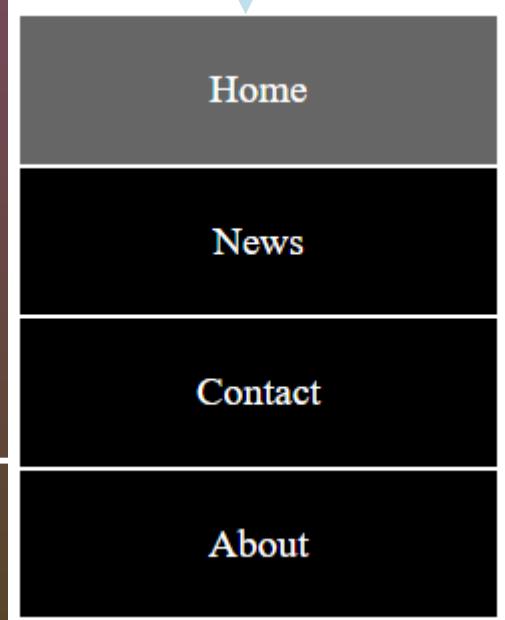
List-style-position: [inside/outside]

```
UL
{
    list-style-type: square;
    list-style-position: inside;
    list-style-image: url ('smile.jpg');
}
```



- Home
- News
- Contact
- About

```
<ul>
  <li>Home </li>
  <li>News</li>
  <li>Contact</li>
  <li>About</li>
</ul>
```



```
ul {
  padding-left:0;
}
ul li
{
  list-style-type:none;
  width:150px;
  border:1px solid #fff;
  display:block;
  background-color:#000;
  text-align:center;
  color:#FFF;
  padding:20px;
}
```

Home

News

Contact

About

```
ul {  
    padding-left:0;  
    background-color:#000;  
}  
ul li  
{  
    list-style-type:none;  
    width:150px;  
    border-right:1px solid #fff;  
    display:inline-block;  
    background-color:#000;  
    text-align:center;  
    color:#FFF;  
    padding:20px;  
}
```

Selector:

- ▶ Input: Apply for all items
- ▶ Input [type=text]: apply for textbox
- ▶ Input [type=text]:focus
- ▶ Input[type=number] : apply for number
- ▶ Select{ }: apply for listbox

CSS FORM

► Padding:

```
input[type=text] {  
    width: 100%;  
    padding: 12px 20px;  
    margin: 8px 0;  
    box-sizing: border-box;  
}
```

First Name

Last Name

```
input[type=text] {  
    border: 2px solid red;  
    border-radius: 4px;  
}
```

First Name

Last Name

- Background :
- Color:

```
<style>
input[type=text] {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
    border: 1px solid #555;
    outline: none;
}

input[type=text]:focus {
    background-color: lightblue;
}
</style>
```

First Name

Last Name

Scrollbar Properties

- Scroll bars allow the browser window to display continuous text.
- CSS allows customize the appearance of the scrollbar by using different scrollbar properties.
- The scrollbar properties are not part of the CSS 2.0 specification but these are supported by IE 5.5 or higher versions.

Property	Description
scrollbar-face-color	Specifies the color of scroll bar

```
textarea
{
background-color:#FFFFCC;
scrollbar-3dlight-color:#ffd700;
scrollbar-arrow-color:#ff0;
scrollbar-base-color:#ff6347;
scrollbar-darkshadow-color:#ffa500;
scrollbar-face-color:#008080;
scrollbar-highlight-color:#ff69b4;
scrollbar-shadow-color:#f0f;
scrollbar-track-color:#800080;
}
```

scrollbars with CSS using the scrollbar property.

Scrollbar colors show up in Internet Explorer 5.5. Other browsers will see the default gray scrollbars.



Screen Properties

- The cursor property: specifies the type of cursor to be displayed when the mouse pointer is placed over the content
- Values of the cursor property:

```
a  
{  
cursor: wait;  
}
```

[Click here to register](#)

Icon	Value	Live example (move mouse over box)
	auto	<i>The User Agent determines the cursor to use.</i>
	default	style="cursor: default;"
	hand	style="cursor: hand;"
	pointer	style="cursor: pointer;"
	hand & pointer	style="cursor: pointer; cursor: hand;"
	crosshair	style="cursor: crosshair;"
	text	style="cursor: text;"
	wait	style="cursor: wait;"
	help	style="cursor: help;"

Screen Properties

- The zoom property: is used to enlarge the contents displayed on the Web page

```
img  
{  
    zoom: 200%;  
}
```



```
<style type="text/css">
nav {
    display: block;
    position: absolute;
    top: 0;
    width: 100%;
    background-color: green;
}
li{
    list-style-type: none;
    display: inline;
    margin-right: 20px;
    font-size: 25px
}
a:link {
    color: #fff;
    text-decoration: none;
}
a:hover {
    color: orange;
    text-decoration: none;
}
li > ul { display: none }
li:hover ul {
    display: block; position: absolute;
    left: 200px;
    background-color: green; margin: 0;
}
li:hover ul li a:link{
    display: block; margin-left: -30px;
}
</style>
```

```
<body>
<nav>
<ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li> <a href="#">Products</a>
        <ul>
            <li><a href="#">Engineering</a></li>
            <li><a href="#">Telecom</a></li>
            <li><a href="#">Energy</a></li>
            <li><a href="#">Finance</a></li>
            <li><a href="#">Consultancy</a></li>
        </ul>
    </li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Contact</a></li>
</ul>
</nav>
</body>
```

- ▶ More powerful formatting than using presentation tags
- ▶ Your pages load faster, because browsers cache the .css files
- ▶ Increased accessibility, because rules can be defined according given media
- ▶ Pages are easier to maintain and update

BENEFITS OF USING CSS

CSS3 - BORDER

- ▶ border-radius: [Example 1](#), [example 2](#), [example 3](#)
- ▶ box-shadow:
- ▶ border-image: [example 1](#), [example 2](#)

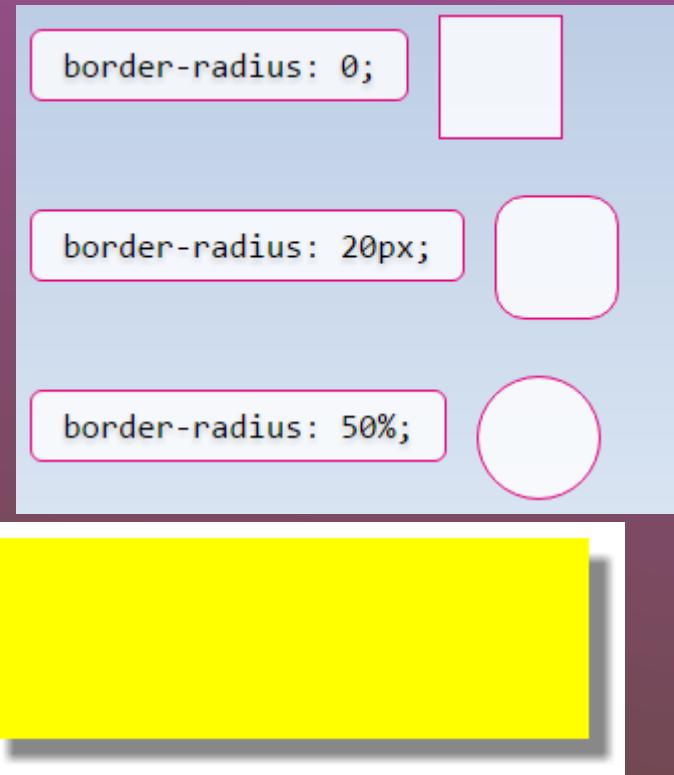
► Background:

- ▶ background-size
- ▶ background-origin

```
#example1 {  
    border: 10px solid black;  
    padding: 35px;  
    background: url(img_flwr.gif);  
    background-repeat: no-repeat;  
    background-origin: content-box;  
}
```

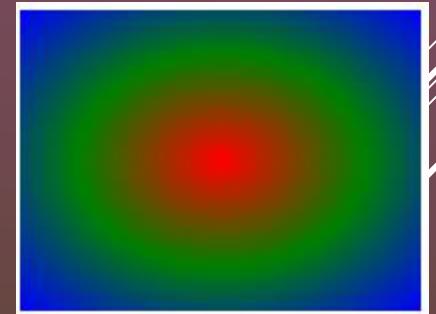
```
#example1 {  
    background: url(img_flwr.gif) left top no-repeat, url(img_flwr.gif) right bottom no-repeat,  
    url(paper.gif) left top repeat;  
    background-size: 50px, 130px, auto;  
}
```

```
div  
{  
box-shadow: 10px 10px 5px  
#888888;  
}
```



- ▶ Linear Gradients (goes down/up/left/right/diagonally)
- ▶ Radial Gradients (defined by their center)
- ▶ Example Radial Gradients

```
#grad1
{
height:150px;
width:200px;
background: -webkit-radial-gradient(red, green,
blue);
background: -o-radial-gradient(red, green, blue);
background: -moz-radial-gradient(red, green, blue);
background: radial-gradient(red, green, blue);
}
```



- ▶ text-shadow:
- ▶ word-wrap: [example](#)

Example

```
▶ h1  
{  
text-shadow: 5px 5px 5px #FF0000;  
}
```

Text shadow effect!

CSS3 TEXT EFFECTS
@font-face: [example](#)

- ▶ translate(): [example](#)
- ▶ rotate(): [example](#)
- ▶ scale(): [example](#)
- ▶ skew(): [example1](#), [example2](#)
- ▶ matrix(): [example](#)

```
div
{
-ms-transform: rotate(30deg);
-webkit-transform: rotate(30deg);
transform: rotate(30deg);
}
```

ON



► rotateX(): [example](#)

► rotateY(): [example](#)

► rotateZ(): [example](#)

```
div
{
-webkit-transform: rotateX(120deg);
transform: rotateX(120deg);
}
```



CSS3- 3D TRANSFORMATION

► CSS3- TRANSITION

- transition:
 - transition-delay
 - transition-duration
 - transition-property
 - transition-timing-function
- [example](#)
- Animation:
 - @keyframes: [example](#)
 - Animation: [example](#)

- **Multiple Columns:**

- column-count
- column-gap
- column-rule

[example](#)

- **User Interface:**

- Resize: [example](#)
- box-sizing: [example](#)
- outline-offset: [example](#)

CSS BASICS

Questions?