

Web Technologies

Kamran

Lecture 7-8

1. CSS

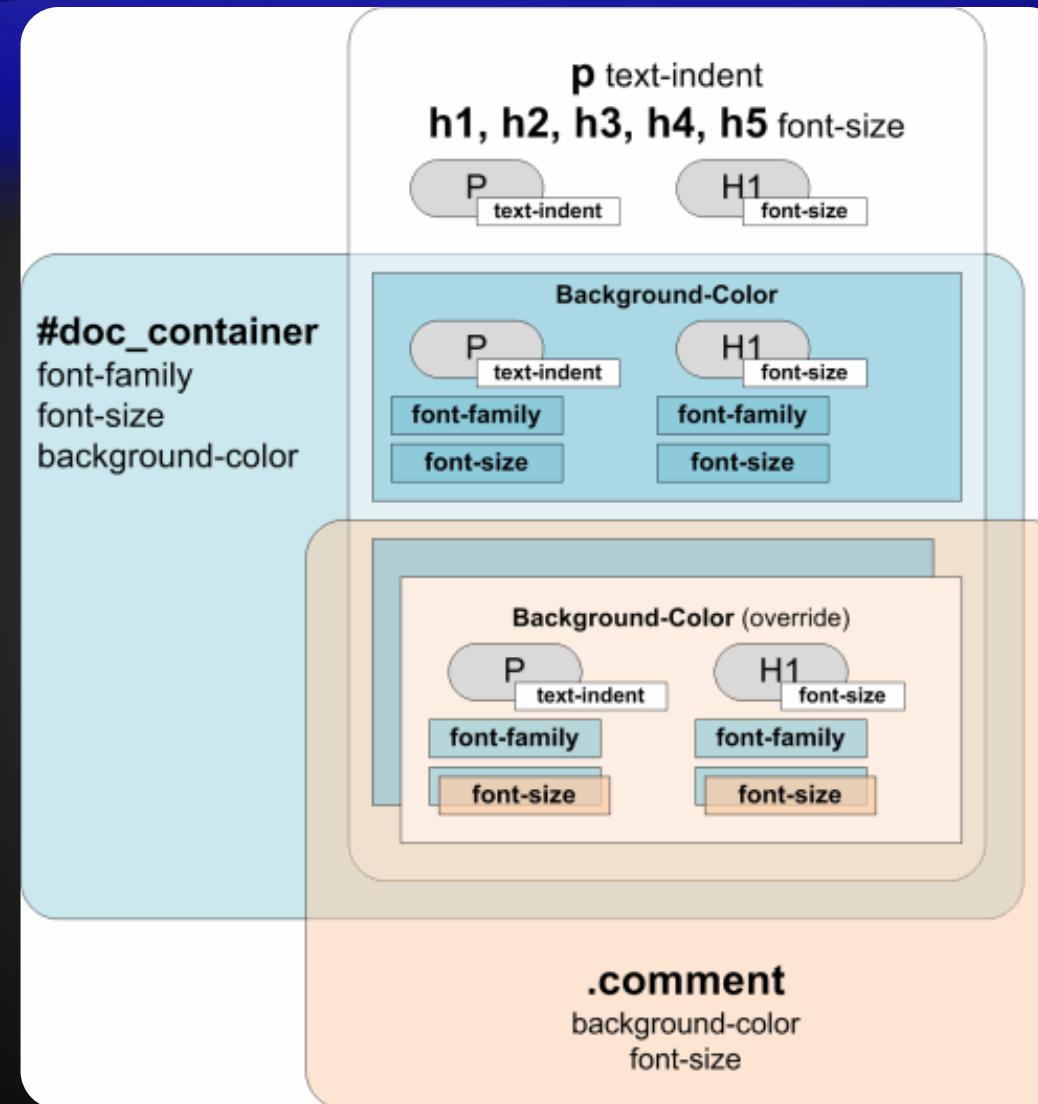


- ◆ Cascading Style Sheets (CSS)
 - ◆ Markup language, used to describe the presentation of document
 - ◆ Defines sizes, fonts, colors, layout, etc.
 - ◆ Improves content accessibility
 - ◆ Improves flexibility
- ◆ Designed to separate presentation from content
- ◆ Because of CSS all HTML presentation tags are deprecated, e.g. , <center>, etc.

Why “Cascading”?

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

Why “Cascading”? (2)



- ◆ CSS has simple syntax, based on English words
- ◆ Contains a set of cascading rules
- ◆ Each rule consists of one or more selectors and declaration block
- ◆ Declaration block consists of one or more semicolon-terminated declarations in curly braces
- ◆ Declaration consists of property, a colon and value

```
h1,h2,h3,h4,h5,h6 { color: green }
```

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

```
h1 .title { color: green }
```

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

- ◆ Pseudo-classes define further behavior
 - ◆ Appended to a selector
 - ◆ :hover, :first-letter, :visited, :active, :before, :after
 - ◆ Not all browsers support them fully

```
a:link {text-decoration: none}  
a:visited {text-decoration: none}  
a:active {text-decoration: none}  
a:hover {text-decoration: underline; color: red}  
.title:before { content: "»" }  
.title:after { content: "«" }
```

- ◆ Three primary types of selectors:

- By tag:

```
h1 {font-face: Verdana}
```

- By element id:

```
#element_id {color:#FF0000}
```

- By element class name (only for HTML):

```
.class_name {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

- ◆ Match relative to element placement:

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

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

- ◆ * – universal selector:

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

This will match all child nodes of `<p>` tag

- ◆ + selector – used to match “the following” tag:

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

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

- ◆ > selector – matches direct child nodes of element:

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

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

- ◆ [] – match tag attributes by regular expression:

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

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

- ◆ There are more rules to select attributes
 - ◆ Not well supported in all browsers

- ◆ **Browsers have default CSS styles**
 - ◆ Used when there is no CSS information or any other style information in the document
 - ◆ Silently inherited in all documents
- ◆ **Caution: default styles differ in browsers**
 - ◆ E.g. Firefox default page background is white, while IE7 uses about 5% gray background

- ◆ 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 HTML in `<style>` tag
 - ◆ External: CSS rules are in separate file
 - ◆ Usually a file with `.css` extension
 - ◆ Linked via `<link rel="stylesheet" href=...>` tag or `@import` directive in embedded CSS block

- ◆ **Inline styles have highest priority**
 - ◆ Then are the embedded styles
 - ◆ **External styles are last**
- ◆ **Using external files is highly recommended**
 - ◆ Simplify the HTML document
 - ◆ Benefit from browser's cache
- ◆ **Inline styles are about to be deprecated by the W3C**

♦ **Inline CSS styles**

- ♦ Individual element's style defined using **style** attribute

- ♦ Contains only declaration, no selectors:

```
<p style="font-size:20pt; color: #0000FF">
```

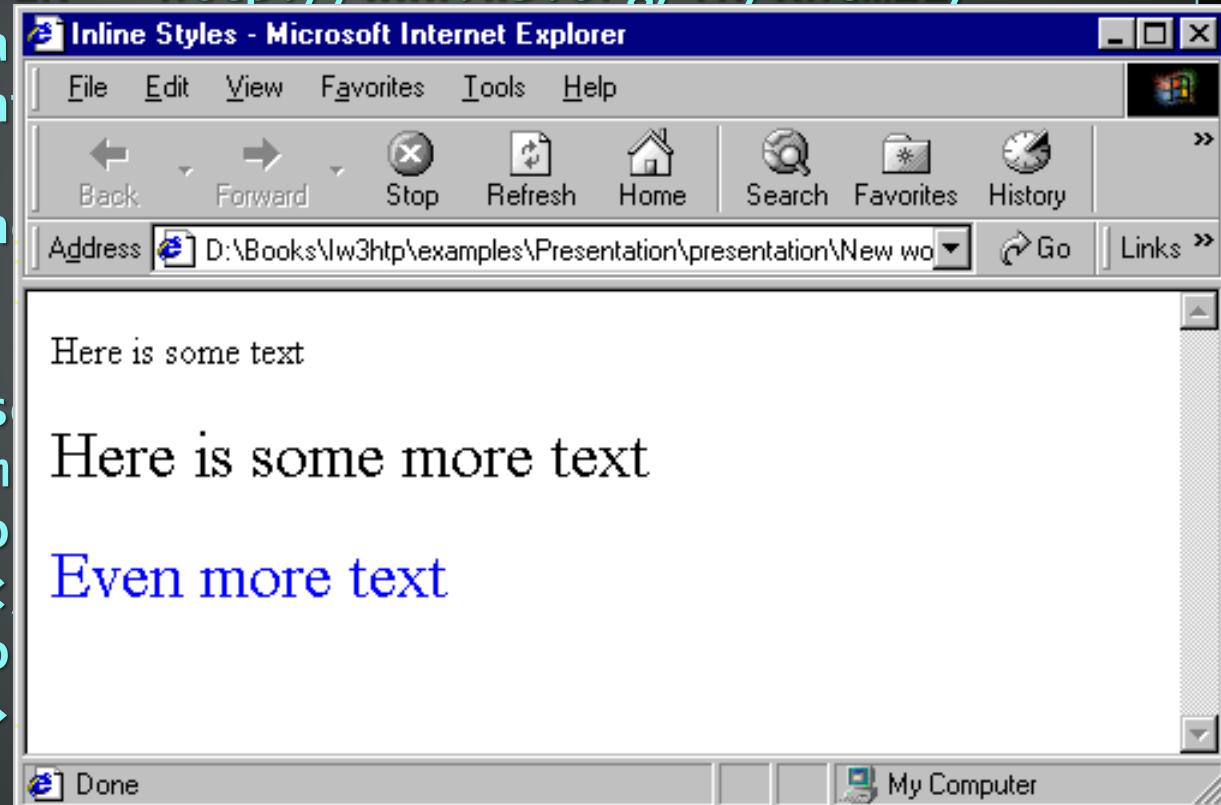
- ♦ Override any other styles
- ♦ Apply to all descendant elements
- ♦ Used for styles that are not needed anywhere else in the Web site

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.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: 1.5em; margin-left: 20px;">  
        more text</p>  
    <p style="font-size: 1.2em; color: #0000FF;">  
        Even more text</p>  
</body>  
</html>
```



- ♦ Inline CSS rules have precedence over the external CSS rules:

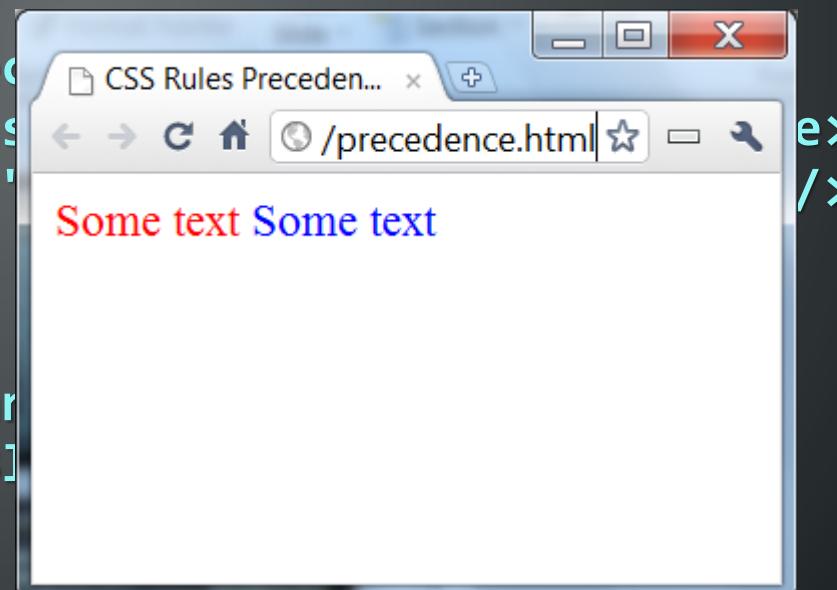
precedence.html

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>CSS Rules Precedence - Example</title>
    <style type="text/css"> span {color:red} </style>
    <link type="text/css" rel="stylesheet" href="" />
</head>
<body>
    <span>Some text</span>
    <span style="color:Blue">Some text</span>
</body>
</html>
```

- ♦ Inline CSS rules have precedence over the external CSS rules:

precedence.html

```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>CSS Rules Precedence</title>
    <style type="text/css">
        body {
            color: red;
        }
    </style>
    <link type="text/css" href="style.css" />
</head>
<body>
    <span>Some text</span>
    <span style="color:blue;">Some text</span>
</body>
</html>
```



- ◆ Embedded in the HTML in the `<style>` tag:

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

- ◆ The `<style>` tag is placed in the `<head>` section of the document
- ◆ Styles apply to the whole document
- ◆ type attribute specifies the MIME type
 - ◆ MIME is a describes the format of the content
 - ◆ Other MIME types include `text/html`, `image/gif`, `text/javascript` ...
- ◆ Used for document-specific styles

embedded-stylesheets.html

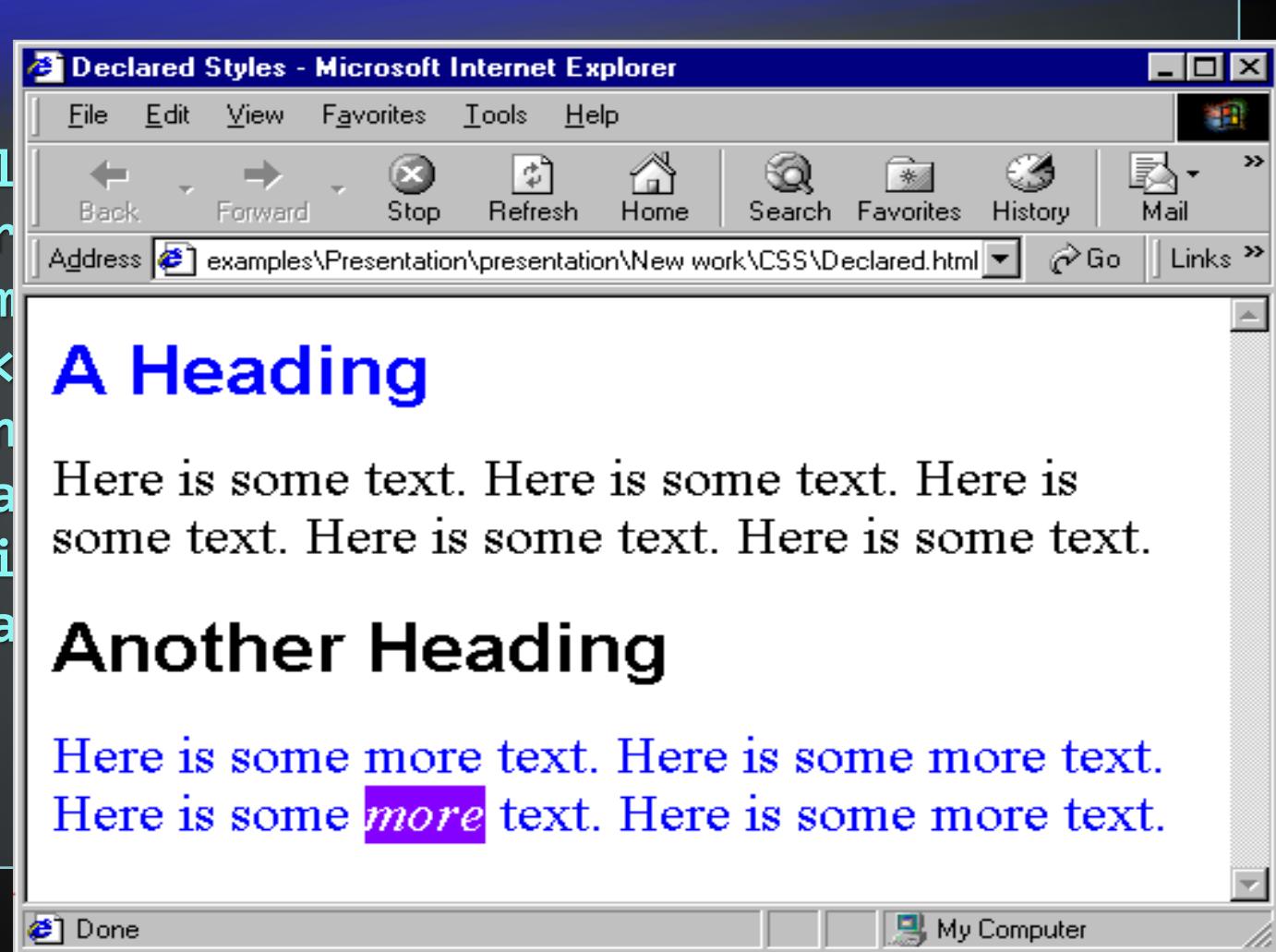
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Style Sheets</title>
    <style type="text/css">
        em {background-color:#8000FF; color:white}
        h1 {font-family:Arial, 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 (3)

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



- ◆ External linking
 - ◆ Separate pages can all use shared style sheet
 - ◆ Only modify a single file to change the styles across your entire Web site
 - ◆ link tag (with rel attribute)
 - ◆ Specifies a relationship between current document and another document
- ```
<link rel="stylesheet" type="text/css"
 href="styles.css">
```
- ◆ link element can stay only in the HTML header

## @import

- Another way to link external CSS files
- Example:

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

- Not all browsers recognize such rules
  - You can specify this way browser-specific styles (IE6 ignores the @import)

## 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>
 <h1>Shopping list for Monday:</h1>
 Milk
 ...
```

# External Styles: Example (3)

```
...
Bread

 White bread
 Rye bread
 Whole wheat bread

Rice
Potatoes
Pizza with mushrooms

<a href="http://food.com" title="grocery
 store">Go to the Grocery store
</body>
</html>
```

# External Styles: Example (4)

```
...
Bread

 White
 Rye b
 Whole

Rice
Potatoes<
Pizza

<a href="http://
 store">Go to
</body>
</html>
```



- ◆ Colors are specified in RGB format, in hex form:
  - ◆ Example: #A0A6AA
  - ◆ Predefined color aliases exist: black, blue, etc.
- ◆ Numeric values are specified in:
  - ◆ Pixels, e.g. 12px
  - ◆ Points, e.g. 10pt
  - ◆ Inches, centimeters, millimeters
    - ◆ E.g. 1in, 1cm, 1mm
  - ◆ Percentages, e.g. 50%
    - ◆ Percentage is relative to the parent element

- ◆ **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 serif font
- ◆ **font-weight** – **normal, bold, bolder, lighter or number in range [100 ... 900]**

- ◆ **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**

- ◆ font

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

```
font: italic normal bold 12px Verdana;
```

is equal to writing this:

```
font-style: italic;
font-variant: normal;
font-weight: bold;
font-size: 12px;
font-family: Verdana;
```



AMERICANA  
AUSTERE  
AVANT GARDE  
**BOOKMAN**  
COCHIN  
**COLLEGE**  
**COOPER**  
DOROVAR  
FRIZ QUAD  
Garamond  
**KABEL**  
**Fractur**  
Palatino  
RALEIGH  
**ROOST**  
**Tiffany**  
TIMES

*Ambiente*  
*Amaze*  
**Becka**  
*Bible Script*  
*Brush Script*  
*Candoo*  
*Caflish*  
*Churchward*  
*Dianna*  
*Flamenco*  
*Indy Script*  
*Nelson*  
*Palette*  
*Scrawlin*  
*Sentimental*  
*Zepher Script*



# Fonts

Live Demo

[font-rules.html](#)



- ◆ **background-color**
  - ◆ Solid color background
- ◆ **background-image**
  - ◆ URL of image to be used as background, e.g.:

```
background-image:url("back.gif");
```
- ◆ **background-repeat**
  - ◆ repeat-x, repeat-y, repeat, no-repeat
- ◆ **background-attachment**
  - ◆ 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:

```
background-position: top left;
```

```
background-position: -5px 50%;
```

- ◆ **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 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 should be moved to the CSS

```
* {
 margin: 0;
 padding: 0;

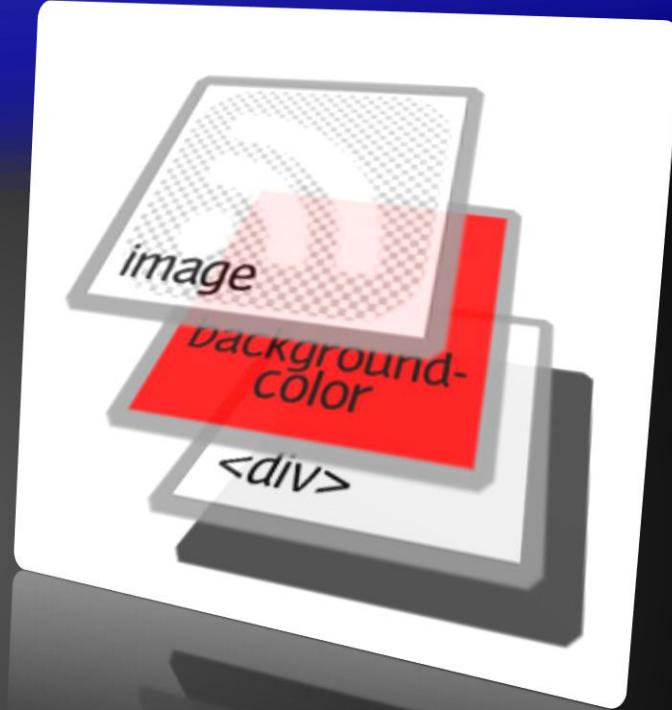
}

body, html {
 min-height: 101%;

}

body {
 background: #333;
 color: #333;
 padding: 13px 0 25px 0

}
}
```



# Background Styles

Live Demo

[background-rules.html](#)

- ◆ **border-width:** thin, medium, thick or numerical 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, ...**

- ◆ **border:** shorthand rule for setting border properties at once:

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

is equal to writing:

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

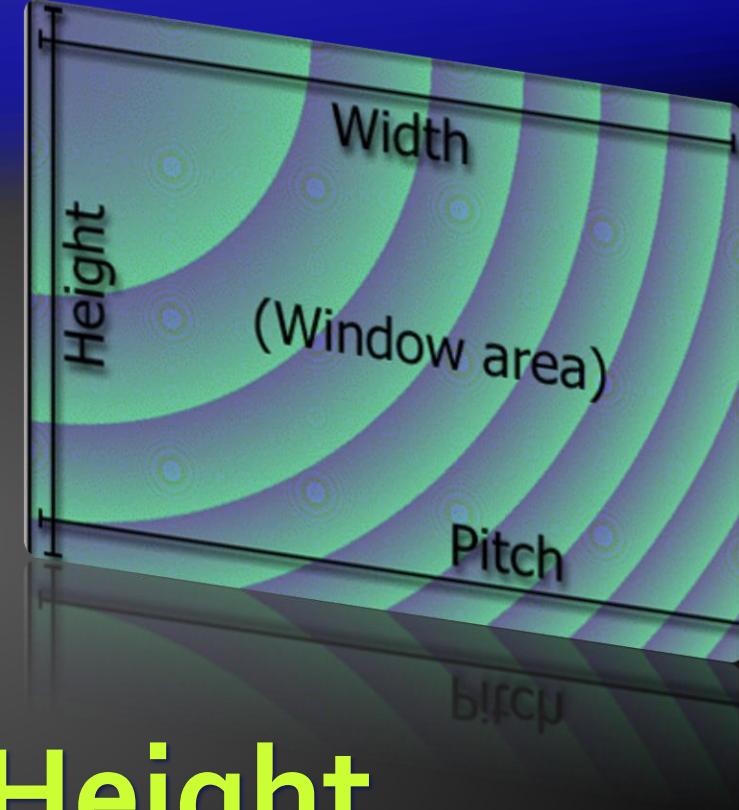
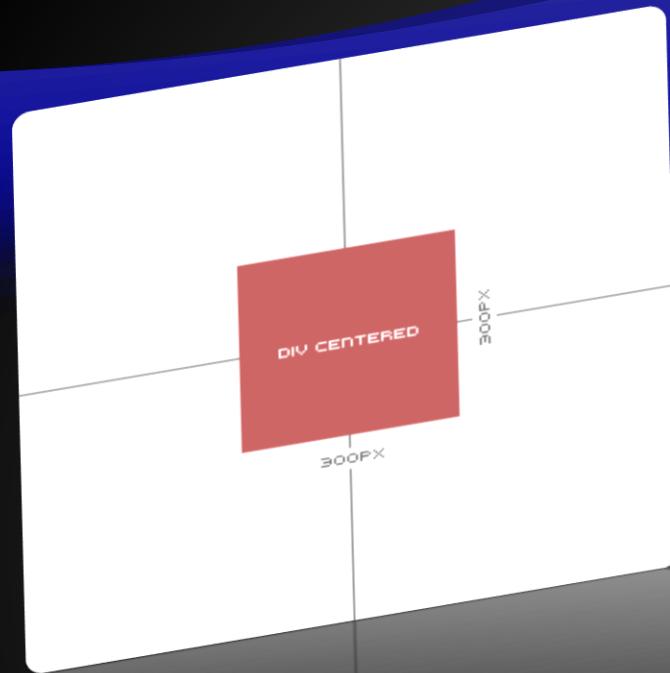
- ◆ Specify different borders for the sides via shorthand rules: **border-top**, **border-left**, **border-right**, **border-bottom**

# Borders

Live Demo

[border-rules.html](#)

- ◆ **width** – defines numerical value for the width of element, e.g. 200px
- ◆ **height** – defines numerical value for the height of element, e.g. 100px
  - ◆ Important: not all elements and browsers follow this value!
  - ◆ Usually the height of an element is defined by its content
  - ◆ Common mistake is to apply height to tables or table cells



# Width / Height

Live Demo

[size-rules.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

# **Margin and Padding: Short Rules**

- ◆ **margin: 5px;**
  - ◆ Sets all four sides to have margin of 5 px;
- ◆ **margin: 10px 20px;**
  - ◆ Sets margins: top and bottom to 10px, left and right to 20px;
- ◆ **margin: 1px 3px 5px 7px;**
  - ◆ Sets top, right, bottom, left margins to 1px, 3px, 5px and 7px respectively
- ◆ Same shorthand rules apply for padding

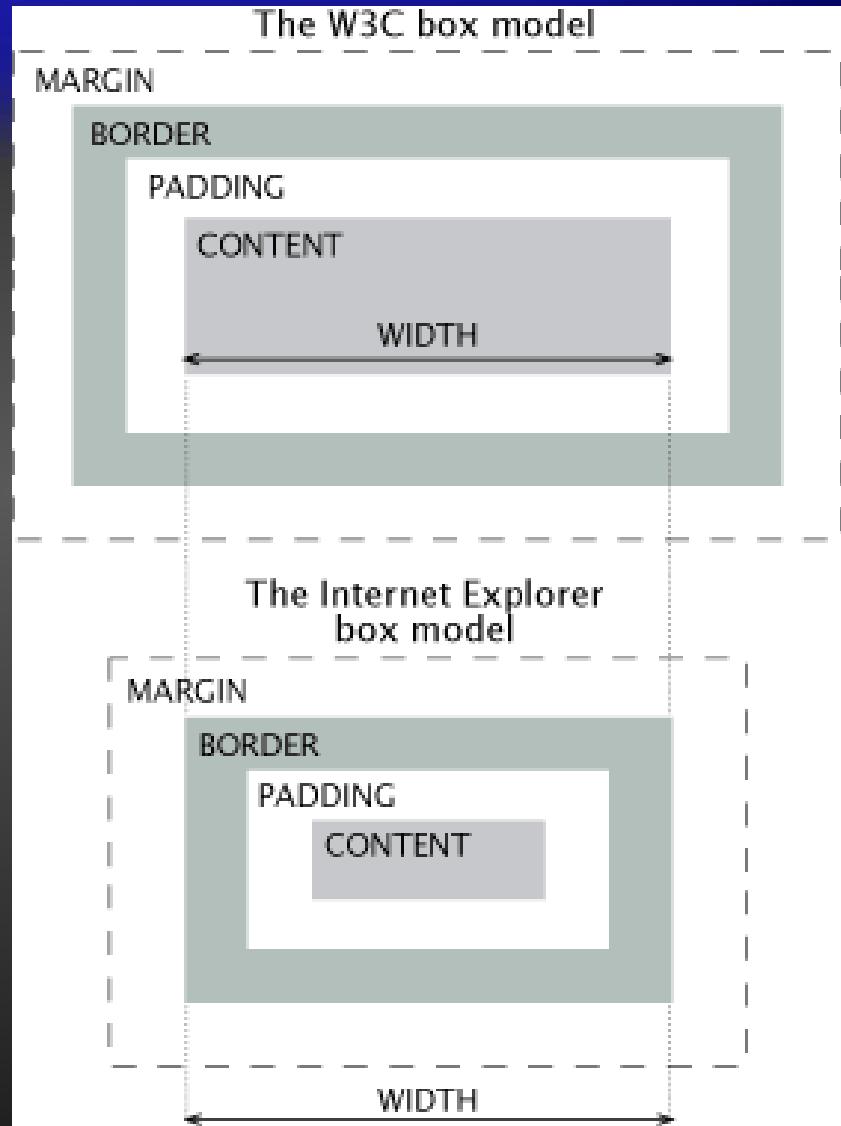
Margin

Border

Padding

Content

- ◆ Major difference between browsers when applying border, padding and width/height
- ◆ To avoid you need either “CSS hacks” or just don’t specify for the same element width/height and padding or border, different than 0





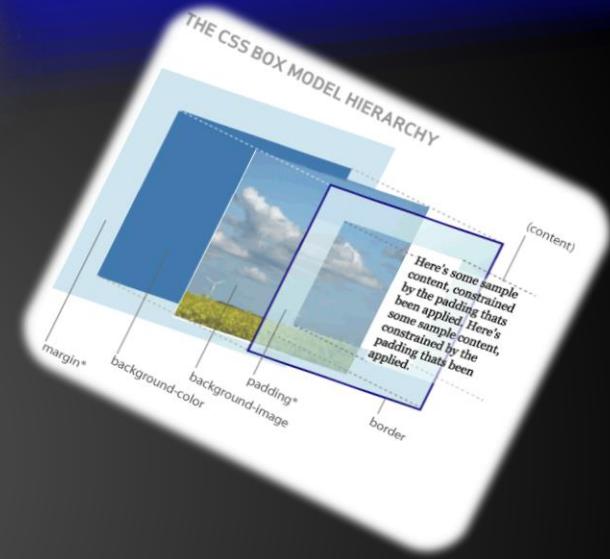
# Margins and Paddings

Live Demo

[margins-paddings-rules.html](#)

- ◆ **position:** defines the positioning of the element, depending on the parent elements
- ◆ The value is one of:
  - ◆ **static** (default)
  - ◆ **relative** – relative position according to where the element would appear with static position
  - ◆ **absolute** – position according the parent element
  - ◆ **fixed** – fix element on screen, ignore page scrolling

- ◆ Fixed and absolute positioned elements “float” over the rest of elements
  - ◆ Moved to separate document layer
  - ◆ Their position and size is ignored when calculating the size of parent element or position of surrounding elements
  - ◆ Ordered by their z-index

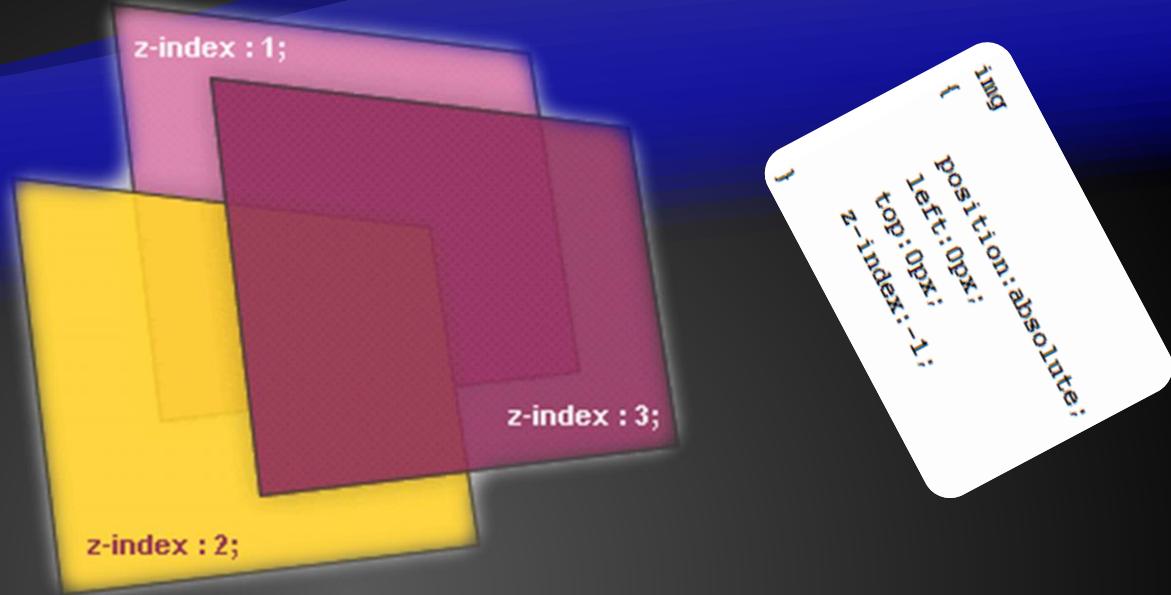


# Positioning

## Live Demo

positioning-rules.html

- ◆ **top, left, bottom, right:** specifies offset of absolute/fixed/relative positioned element as numerical values
- ◆ **vertical-align:** sets the vertical-alignment of an element
  - ◆ Usually used for table cells
  - ◆ Values: **baseline, sub, super, top, text-top, middle, bottom, text-bottom or numeric**
- ◆ **z-index:** specifies the depth-order of element



# Alignment and Z-Index

## Live Demo

[alignments-and-z-index-rules.html](#)

- ◆ **float:** the element “floats” above the elements
  - ◆ similar to position: absolute and the align HTML property of <img> tag
  - ◆ element is taken into account when rendering the surrounding text and elements
  - ◆ **left:** places the element on the left
  - ◆ **right:** places the element on the right

- ◆ **clear**

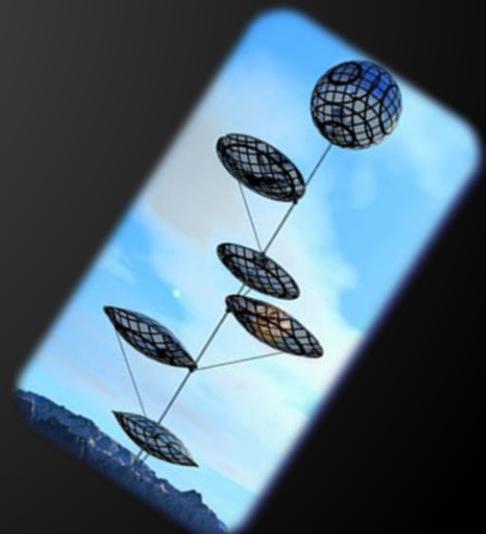
- ◆ Sets the sides of the element where other floating elements are NOT allowed
- ◆ Possible values: **left, right, both**
- ◆ This rule can be applied only to “floating” elements



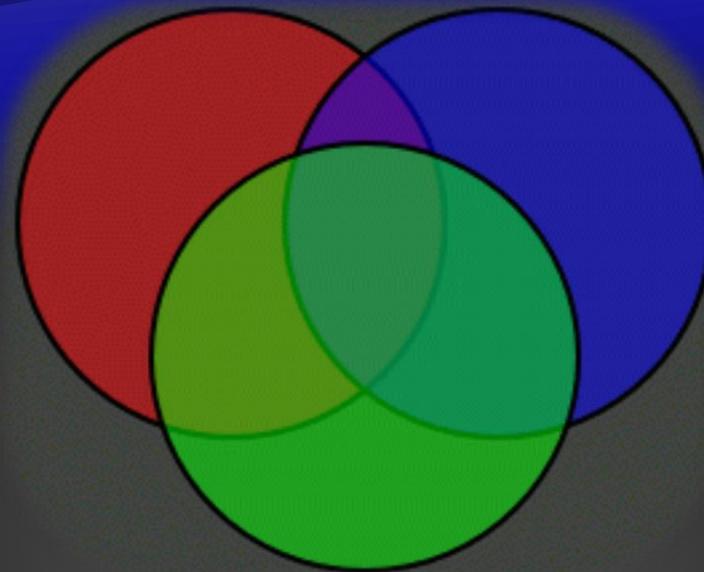
# Floating Elements

Live Demo

[float-rules.html](#)



- ◆ **opacity:** specifies the opacity of the element
  - ◆ Floating point number from 0 to 1
  - ◆ Supported only by Safari browser
  - ◆ For Mozilla use **-moz-opacity** CSS rule
  - ◆ For IE use **filter:alpha(opacity=value)**  
where value is from 0 to 100
  - ◆ Need to apply all three rules



# Opacity

Live Demo

[opacity-rule.html](#)

- ◆ **visibility**

- ◆ **Determines whether the element is visible**
- ◆ **hidden: element is not rendered, but still takes place in the rendering (acts like opacity:0)**
- ◆ **visible: element is rendered normally**

# Visibility

Live Demo

[visibility-rule.html](#)



- ◆ **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 (**<span>** is inline element)
  - ◆ **block:** breaks are placed before AND after the element (**<div>** is block 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

Live Demo

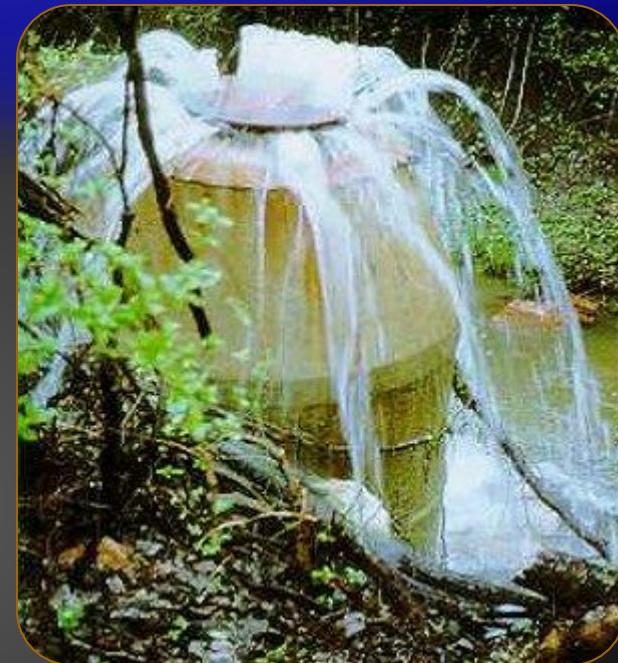
[display-rule.html](#)

- ◆ **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) – element size is increased to make space for content or the content “overflows” out of the element
  - **scroll** – show horizontal/vertical scroll bar in the element
  - **hidden** – any content in the element that cannot be placed inside is hidden

# Overflow

Live Demo

[overflow-rule.html](#)



- ◆ **cursor:** specifies the look of the mouse cursor when placed over the element
  - ◆ Values: crosshair, help, pointer, progress, move, hair, col-resize, row-resize, text, wait, copy, drop, and others
- ◆ **white-space** – controls the line breaking of text. Value is one of:
  - ◆ nowrap – keeps the text on one line
  - ◆ normal (default) – browser decides whether to break the lines if needed

- ◆ 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

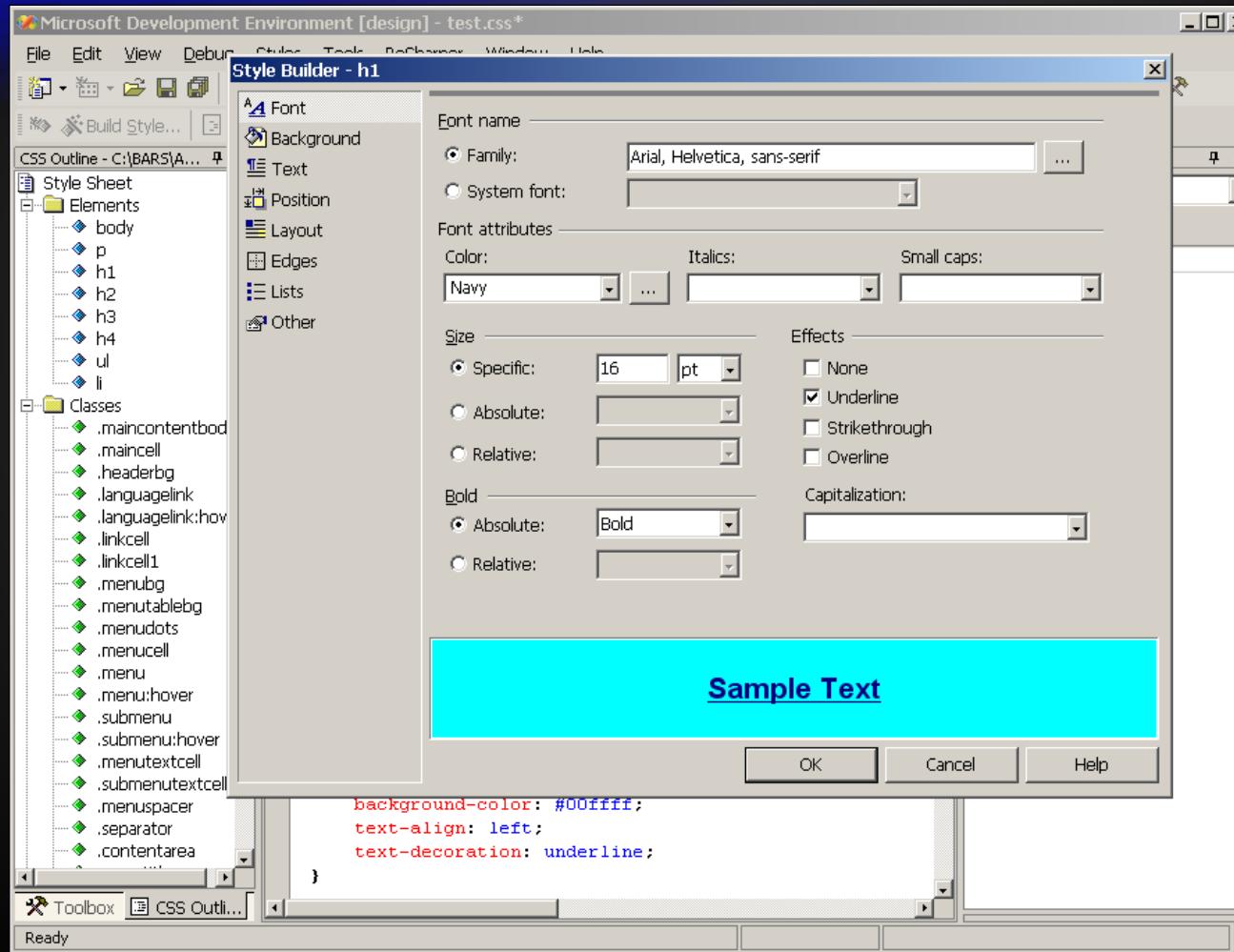
# Maintenance Example



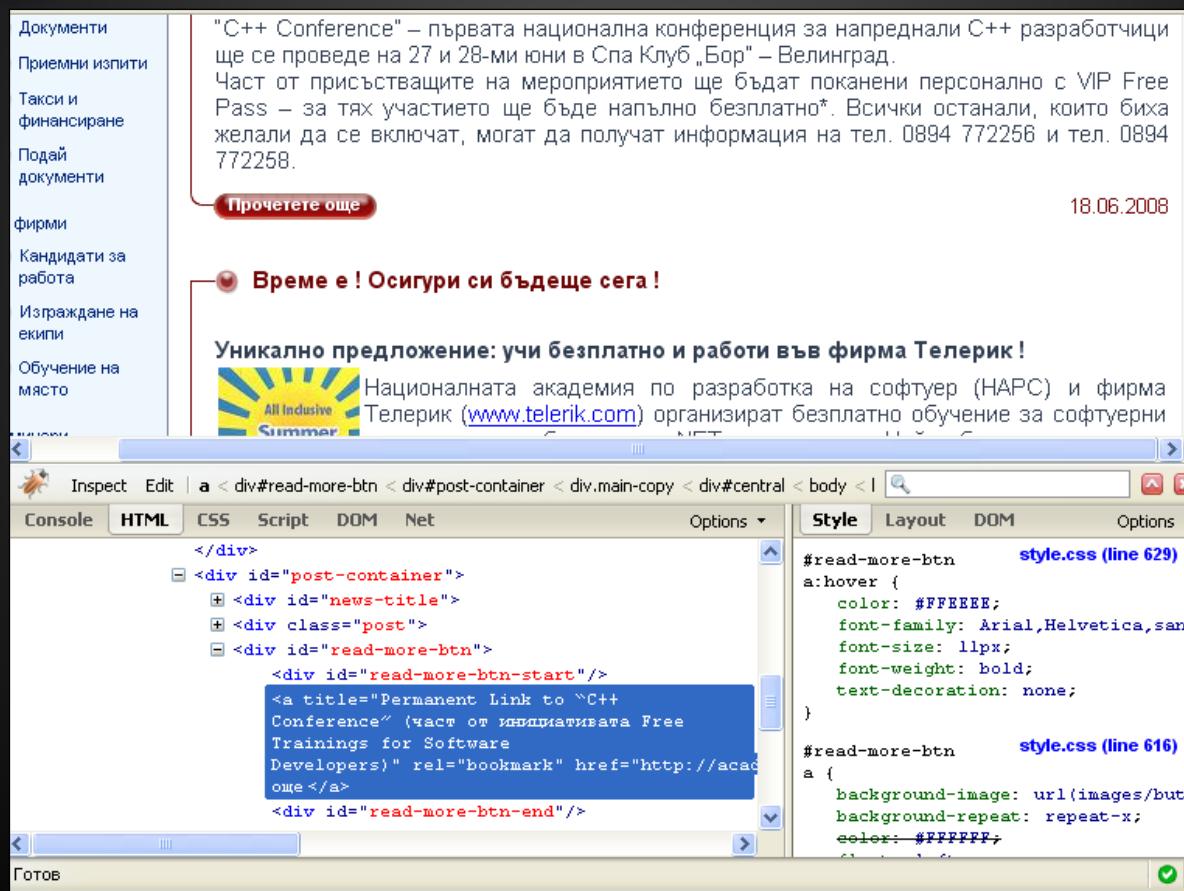
one  
CSS  
file

- ◆ Don't use <font> or other presentation tags to change the look of your text
  - ◆ <font color=...>
  - ◆ <b>text that is bold
  - ◆ <center> this text is centered
- ◆ Do not use complex CSS rules-sets because may slow down page rendering
- ◆ Move as much as possible to external files
- ◆ Avoid inline CSS

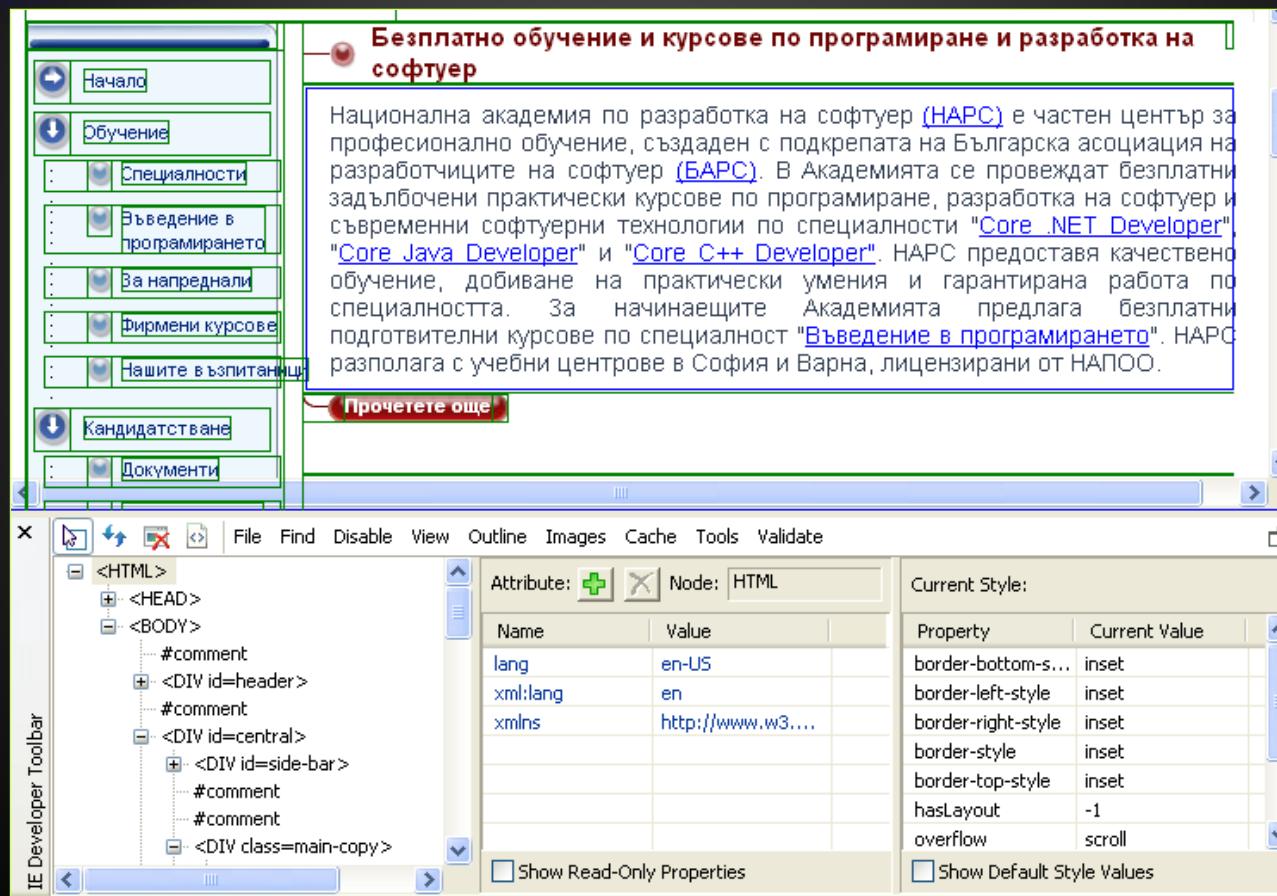
## ◆ Visual Studio – CSS Editor



- ◆ Firebug – add-on to Firefox used to examine and adjust CSS and HTML



- ◆ IE Developer Toolbar – add-on to IE used to examine CSS and HTML (press [F12])



## Questions?