Școala
informală
de IT

# Cascading Style Sheet
# (CSS)

## Agenda

- What is CSS 3?
- Selectors
- Box Model
- Fonts
- Positioning, display & overflow
- Colors
- Backgrounds
- Borders
- Animations

# CSS Intro

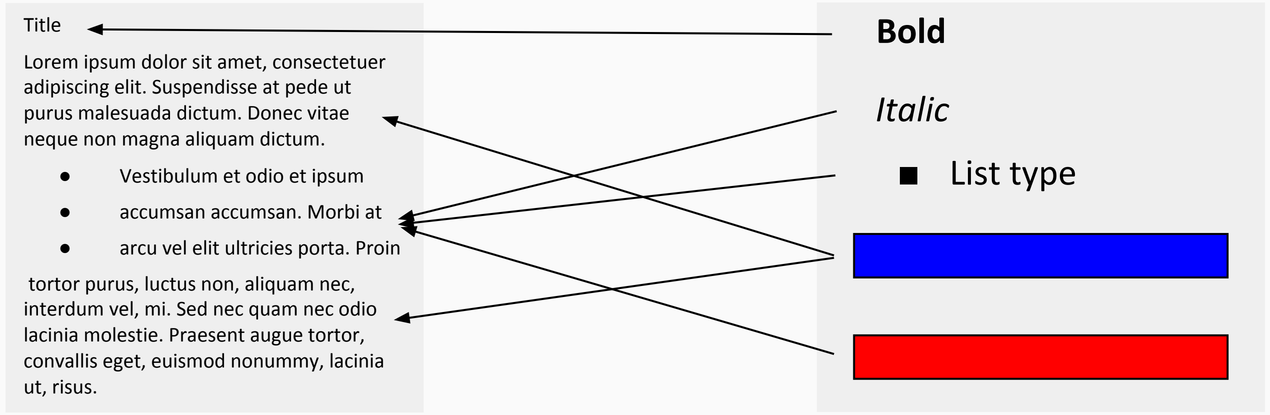
## Styling with Cascading Stylesheets

## CSS

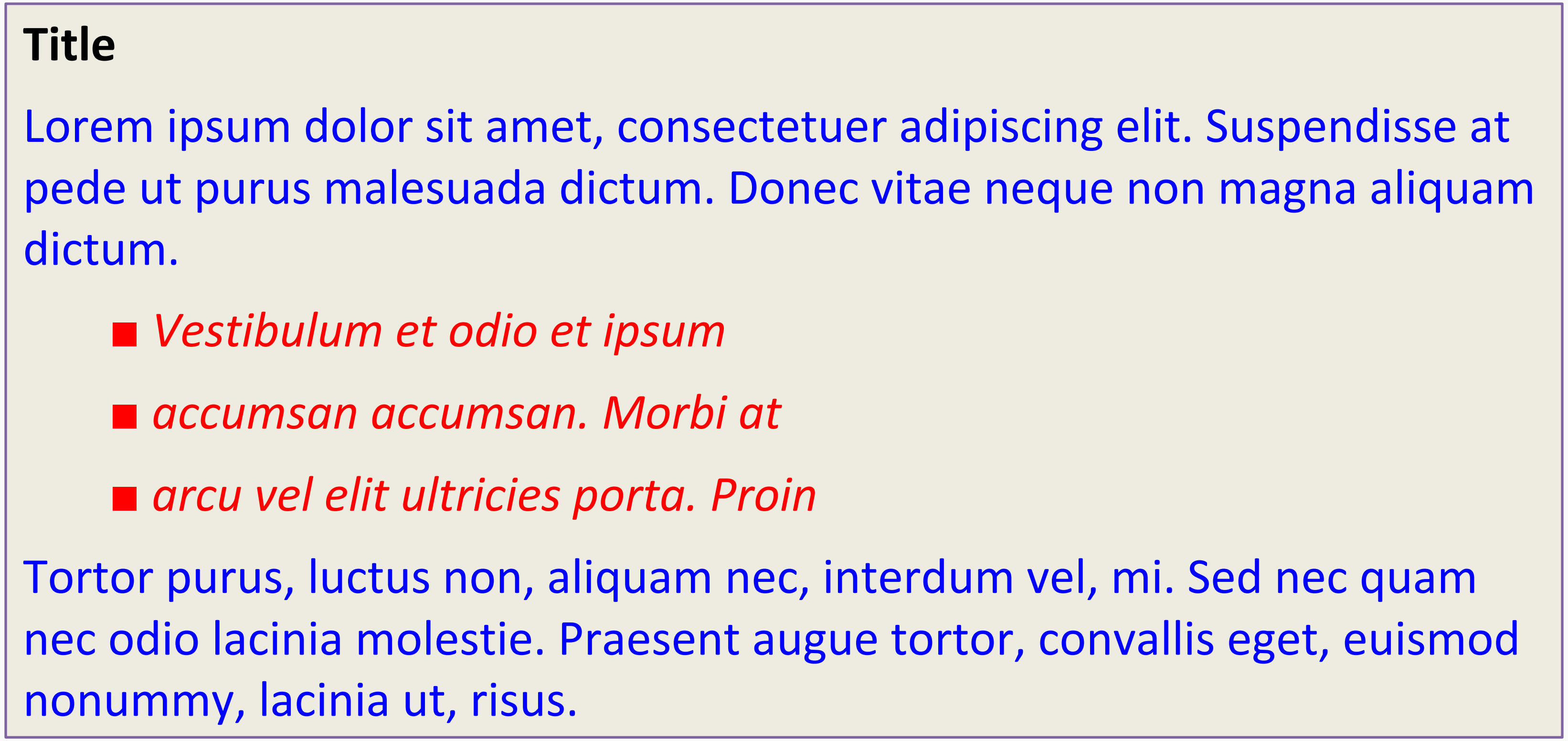
- **Separate content from presentation!**

**Content (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
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
Italic
List type

## What is rendered



**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 Introduction

- **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 specify different styles for different media
  - On-screen
  - In print
  - Tablets, smartphones, projectors, etc.
  - … even by voice or Braille-based reader
- Responsive design

## Why “Cascading”?

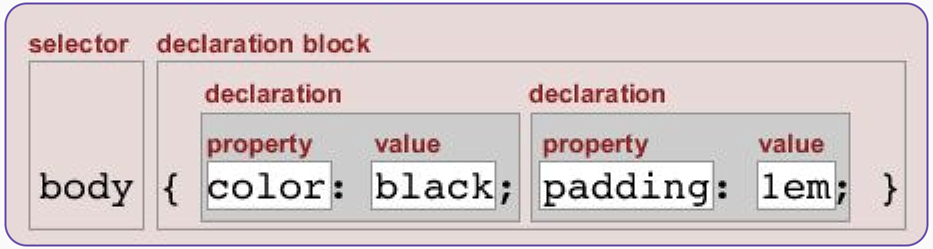
- Priority scheme determining which style rules apply to element
    - Cascade priorities or specificity (weight) are calculated and assigned to the rules
    - Child elements in the HTML DOM tree inherit styles from their parent
        - Can override them
        - Control via `!important` rule

## Why "Cascading"?

- 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

## StyleSheet Syntax

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



- Selectors are separated by commas
- Declarations are separated by semicolons
- Properties and values are separated by colons

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

## Selectors

## Selectors

- 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:

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

- Selectors can be combined with commas:

```
h1, .link, #top {font-weight: bold}
```

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

## 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: "«"; }
```

## Selectors

- 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" which immediately follows:

```
img + .link {border: 1px solid red}
```

  - This will match all siblings with class name `link` that appear immediately after `<img>` tag

## Selectors

- `>` 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 `<img>` tags with `alt` attribute containing the word `logo`
- `.class1.class2` (no space) - matches elements with both (all) classes applied at the same time

## Attribute Selectors

- `E[foo^="bar"]`
  - An E element whose "foo" attribute value begins exactly with the string "bar"
  - Example: `a[src^="https://"]`
- `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"

## 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

## 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

## 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 parent
- `E:empty`
  - An E element that has no children (including text nodes)
- More detailed descriptions:

[http://www.w3.org/TR/css3-selectors/#structural-pseudos](http://www.w3.org/TR/css3-selectors/#structural-pseudos)

## 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)

## Other CSS 3 Selectors

- `E:target`
  - A user interface element E, which is enabled
- `E:not(s)`
  - An E element that does not match simple selector
- E ~ F
  - An F element preceded by an E element

## CSS Specificity

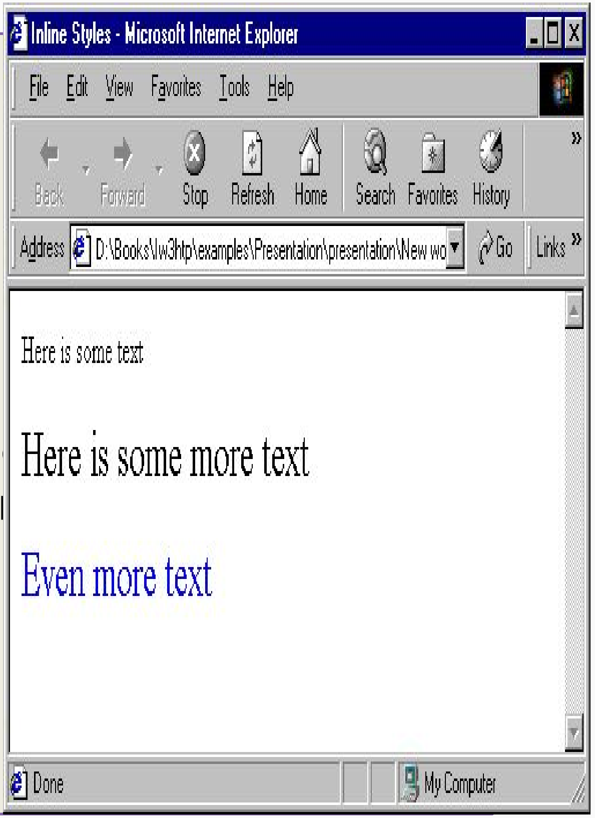
- CSS specificity is used to determine the precedence of CSS style declarations with the same origin. Selectors are what matters
  - Simple calculation: #id = 100, .class = 10, :pseudo = 10, [attr] = 10, tag = 1, * = 0
  - Same number of points? Order matters.
  - See also:
    - http://www.smashingmagazine.com/2007/07/27/css-specificity-things-you-should-know/
    - https://specificity.keegan.st/
    - http://css.maxdesign.com.au/selectutorial/advanced_conflict.htm

## 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, simplifies the HTML file)
    - Usually a file with `.css` extension
    - Linked via `<link rel="stylesheet" href=...>` tag or `@import` directive in embedded CSS block only

## Inline Styles: Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Inline Styles</title>
</head>
<body>
  <p>Here is some text</p>
  <!--Separate multiple styles with a semi
  <p style="font-size: 20pt">Here is some 
  <p style="font-size: 20pt;color: #0000FF
text</p>
</body>
</html>
```


Inline Styles - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address D:\Books\iw3htp\examples\Presentation\presentation\New wo
Go Links
Done
My Computer

Here is some text

Here is some more text

Even more text

## Embedded Styles

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

```
<style>
    ...
</style>
```

  - The `<style>` tag is placed in the <head> section of the document
  - Used for document-specific styles

## Embedded Styles: Example

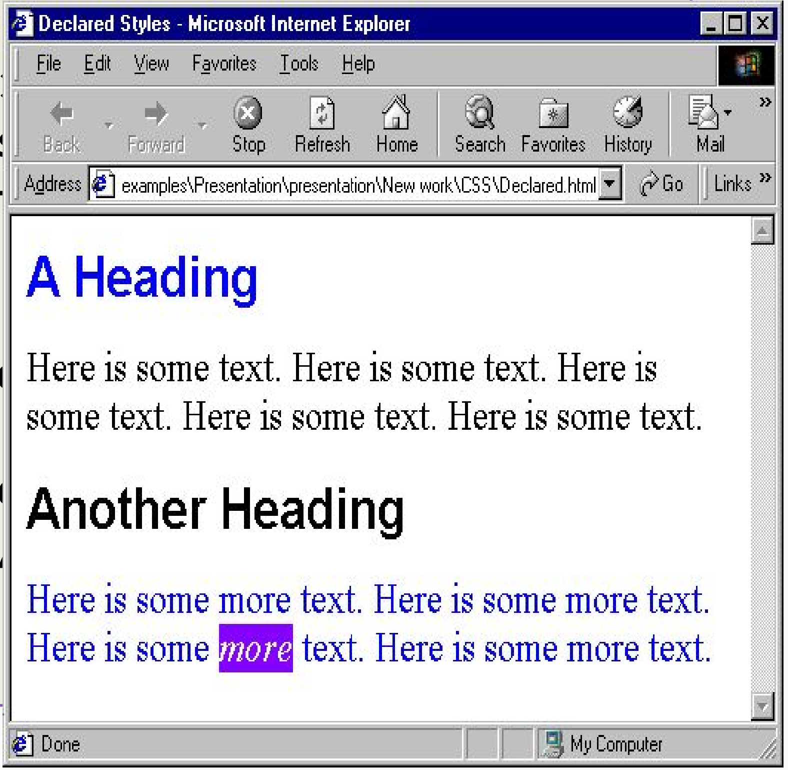
embedded-stylesheets.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Style Sheets</title>
  <style>
    em {background-color: #8000FF; color: white}
    h1 {font-family: Arial, sans-serif}
    p  {font-size: 18pt}
    .blue {color: blue}
  </style>
<head>
```

## Embedded Styles: Example

<u>**embedded-stylesheets.html**</u>

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


Declared Styles - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address examples\Presentation\presentation\New work\CSS\Declared.html
Go Links
Done
My Computer

A Heading

Here is some text. Here is some text. Here is some text. Here is some text. Here is some text.

Another Heading

Here is some more text. Here is some more text. Here is some *more* text. Here is some more text.

## 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">
```

  - `link` elements should be in the `<head>`

## External CSS Styles

**styles.css**

```
em {
    background-color: #8000FF;
    color: white
}
h1 {
    font-family: Arial, sans-serif
}
p {
    font-size: 18pt
}
.blue {
    color: blue
}
```

## Default Browser Styles

- 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; }
```

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

# Fonts

## Text-related CSS Properties

- `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]

## CSS Rules for Fonts

- `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`

## Compact Font Property

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

```
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;
```

## Font Embeds

- Use `@font-face` to declare font
- Point to font file on server
- Call font with font-family
- Currently not supported in IE
- Use font embedding instead of images

```
@font-face {
    font-family: Fancy;
    src: url('Fancy-Bold.ttf');
}
.my_class {
    font-family: Fancy;
    font-size: 3.2em;
}
```

## Text Shadow

- Applies shadow to text
- Syntax: `text-shadow:`

  `<horizontal-distance>`

  `<vertical-distance>`
  `<blur-radius> <shadow-color>;`
- Do not alter the size of a box

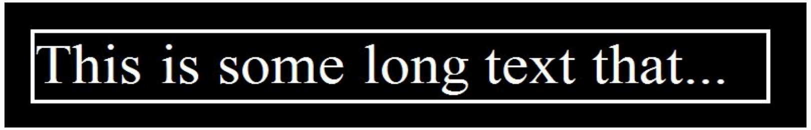
Some shadowed text

```
text-shadow: 2px 2px 7px #000000;
```

Some shadowed text

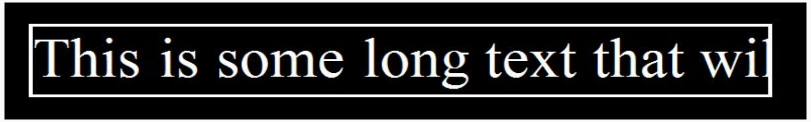
## Text Overflow

- Specifies what should happen when text overflows the containing element
- Syntax: `text-overflow: <value>;`
- Possible values:
  - `ellipsis` - Display ellipses to represent clipped text



This is some long text that...

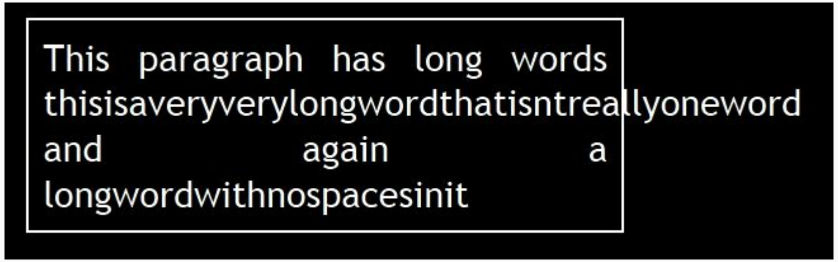
  - `clip` - Default value, clips text

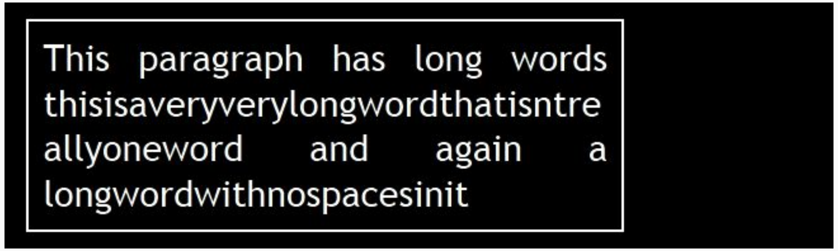


This is some long text that wil

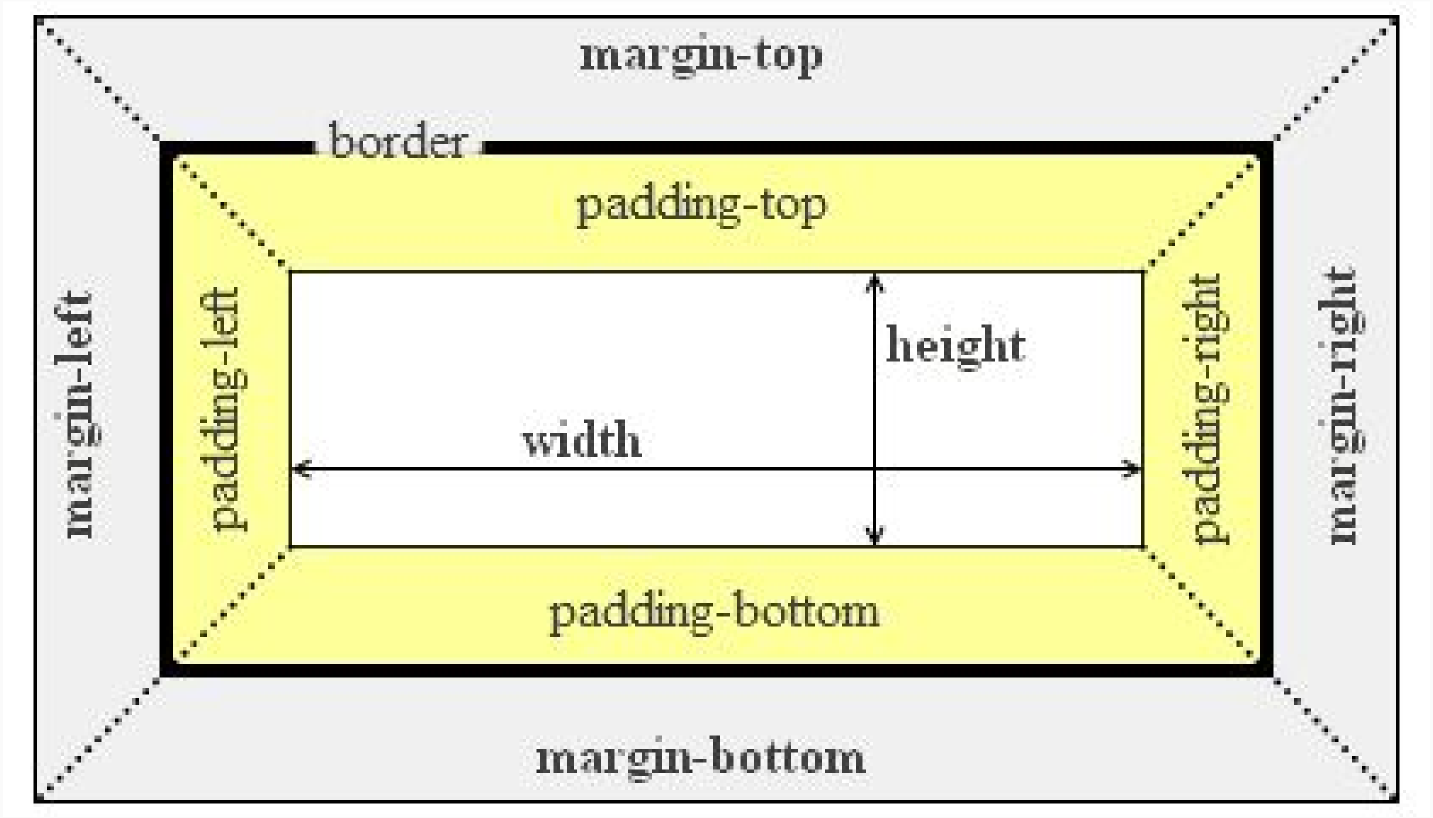
## Word Wrapping

- Allows long words to be able to be broken and wrap onto the next line
- Syntax: `word-wrap: <value>;`
- Possible values:
  - normal
  - break-word


This paragraph has long words thisisaveryverylongwordthatisntreallyoneword and again a longwordwithnospacesinit


This paragraph has long words thisisaveryverylongwordthatisntre allyoneword and again a longwordwithnospacesinit

# The Box Model


margin-top
border
padding-top
margin-left
padding-left
height
width
padding-right
margin-right
padding-bottom
margin-bottom

## Margin, Padding

- `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, Padding Short Rules

- `margin: 5px;`
  - Sets all four sides to have margin of 5 pixels;
- `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`

## Width and Height

- `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.

## CSS 3 box-sizing

- Determine whether you want an element to render its borders and padding within its specified width, or outside of it.
- Possible values:
  - `box-sizing: content-box` (default)
    box width: 288 pixels + 10 pixels padding and 1 pixel border on each side = 300 pixels
  - `box-sizing: border-box`
    box width: 300 pixels, including padding and borders

## Display & Overflow, Positioning

## Visibility

- `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

## 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`!)
  - `inline`: no breaks are placed before and after (`<span>` 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
  - `inline-block:` elements are like inline elements but they can have a width and a height.

## Overflow

- `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
  - `hidden` – any content that cannot fit is clipped

## Other CSS Properties

- `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

## Positioning

- `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

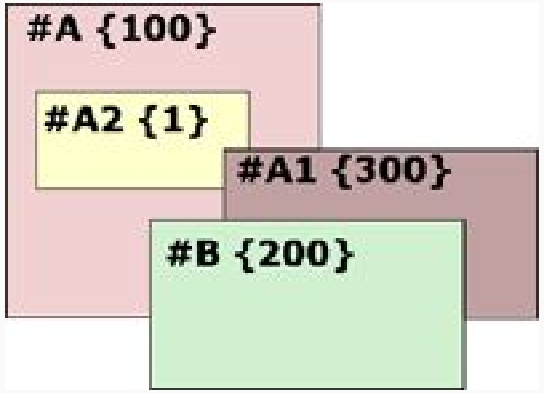
- Margin vs. relative positioning
- Fixed and absolutely positioned elements do not influence the page normal flow and usually stay on top of other elements
  - Their position and size is ignored when calculating the size of parent element or position of surrounding elements
  - Overlaid according to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

## Positioning

- `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. 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.
  - Good demo:

    http://www.barelyfitz.com/screencast/html-training/css/positioning/

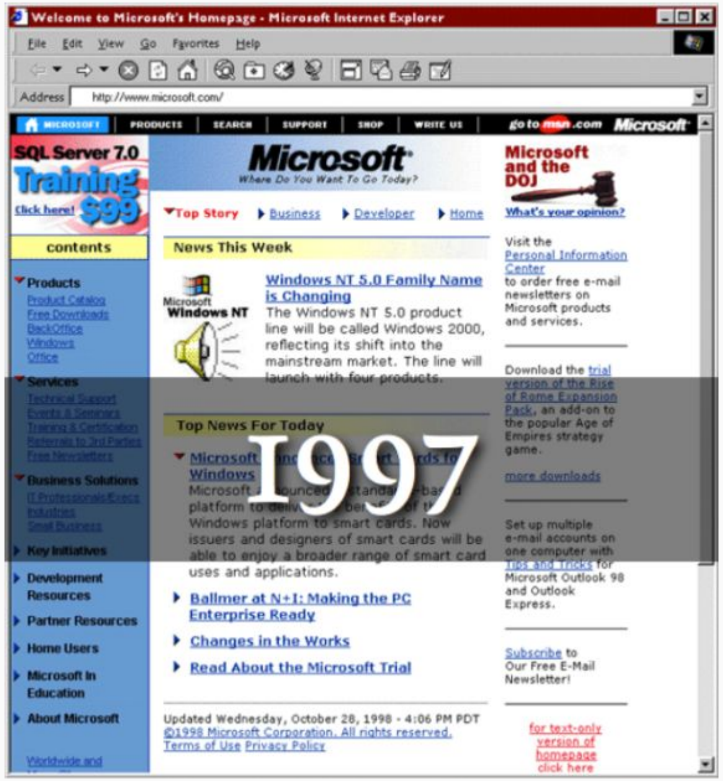


## Inline element positioning

- `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)

## What is CSS 3?

# What is CSS 3?



Welcome to Microsoft's Homepage - Microsoft Internet Explorer

File Edit View Go Favorites Help

Address http://www.microsoft.com/

MICROSOFT PRODUCTS SEARCH SUPPORT SHOP WRITE US go to msn.com Microsoft

SQL Server 7.0 Training Click here! $99

Microsoft Where Do You Want To Go Today?

Microsoft and the DOJ What's your opinion?

Top Story Business Developer Home

contents

Products Product Catalog Free Downloads BackOffice Windows Office

Services Technical Support Events & Seminars Training & Certification Referrals to 3rd Parties Free Newsletters

Business Solutions IT Professionals/Execs Industries Small Business

Key Initiatives

Development Resources

Partner Resources

Home Users

Microsoft in Education

About Microsoft

Worldwide and

News This Week

Microsoft Windows NT

Windows NT 5.0 Family Name is Changing

The Windows NT 5.0 product line will be called Windows 2000, reflecting its shift into the mainstream market. The line will launch with four products.

Top News For Today

Microsoft Announces Smart Cards for Windows

Microsoft announced standards-based platform to deliver the benefits of the Windows platform to smart cards. Now issuers and designers of smart cards will be able to enjoy a broader range of smart card uses and applications.

Ballmer at N+I: Making the PC Enterprise Ready

Changes in the Works

Read About the Microsoft Trial

Updated Wednesday, October 28, 1998 - 4:06 PM PDT
©1998 Microsoft Corporation. All rights reserved.
Terms of Use Privacy Policy

Visit the Personal Information Center to order free e-mail newsletters on Microsoft products and services.

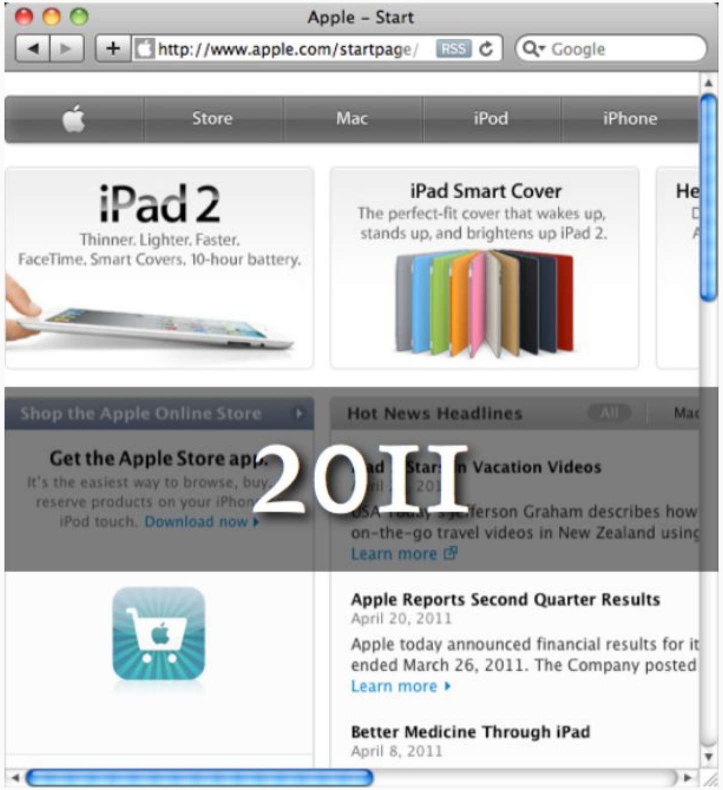
Download the trial version of the Rise of Rome Expansion Pack, an add-on to the popular Age of Empires strategy game.

more downloads

Set up multiple e-mail accounts on one computer with Tips and Tricks for Microsoft Outlook 98 and Outlook Express.

Subscribe to Our Free E-Mail Newsletter!

for text-only version of homepage click here



Apple – Start

http://www.apple.com/startpage/ RSS Google

Store Mac iPod iPhone

iPad 2
Thinner. Lighter. Faster.
FaceTime. Smart Covers. 10-hour battery.

iPad Smart Cover
The perfect-fit cover that wakes up, stands up, and brightens up iPad 2.

Shop the Apple Online Store

Get the Apple Store app.
It's the easiest way to browse, buy, reserve products on your iPhone or iPod touch. Download now

Hot News Headlines All Mac

iPad Stars in Vacation Videos
April 2011
USA Today's Jefferson Graham describes how on-the-go travel videos in New Zealand using
Learn more

Apple Reports Second Quarter Results
April 20, 2011
Apple today announced financial results for its ended March 26, 2011. The Company posted
Learn more

Better Medicine Through iPad
April 8, 2011

## What is CSS 3?

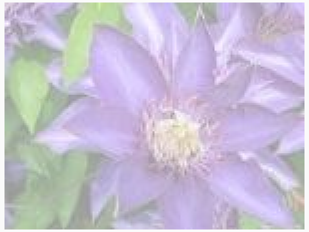
- Cascading Style Sheet level 3 is the most recent iteration of CSS.
- It is divided into several separate documents called “modules”.
- CSS 3 has not been approved as a specification, but there are already a lot of properties that are supported in various browsers.
- The earliest CSS 3 drafts were published in June 1999.

## Colors

## Opacity

- Sets the opacity level for an element
- Syntax: `opacity: <value>;`
- Value from `0.0` (fully transparent) to `1.0`
- The opacity is supported in all major browsers.
- Note: IE8 and earlier supports an alternative, the filter property: filter: Alpha(opacity=50).

```
<img src="img.jpg" style= " opacity: 0.4;
filter: alpha(opacity=40)" />
```

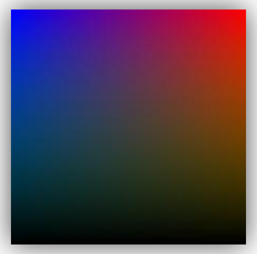


## Values in the CSS Rules

- 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%`

## RGBA Colors

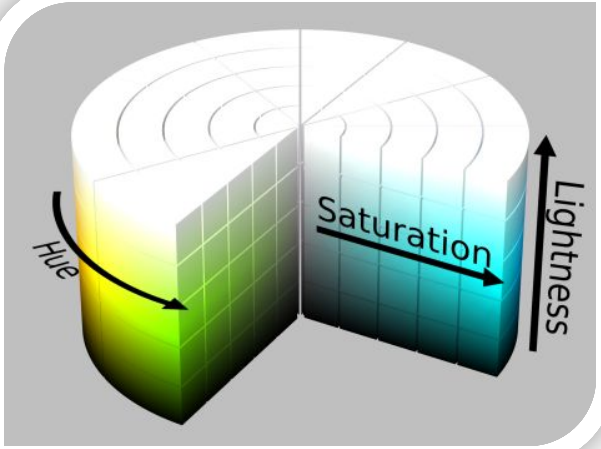
- Standard RGB colors with an opacity value for the color (alpha channel)



- 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)`

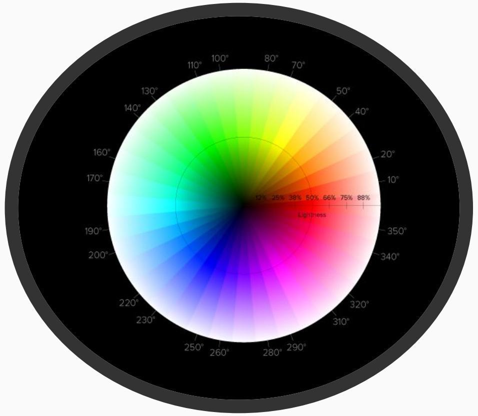
## HSL Colors

- 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



## 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: 🟥



## Backgrounds

## Backgrounds

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

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

- `background-color`
  - Using color and image and the same time
- `background-repeat`
  - `repeat-x`, `repeat-y`, `repeat`, `no-repeat`
- `background-attachment`
  - `fixed` / `scroll`

## Backgrounds

- `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%;
```

## Compact Backgrounds 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;
```

## Background-image or `<img>`?

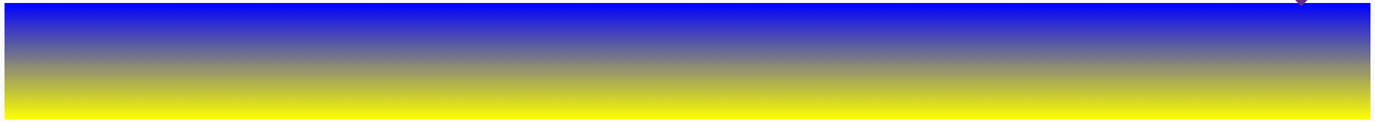
- Background images allow you to save many image tags from the HTML
  - Leads to less code
  - More content-oriented approach
- All images that are not part of the page content (and are used only for "beautification") should be moved to the CSS

## Gradient Backgrounds

- Gradients are smooth transitions between two or more specified colors
- Use of CSS gradients can replace images and reduce download time
- Create a more flexible layout, and look better while zooming
- Supported in all major browsers via different keywords

## Gradient Backgrounds Example

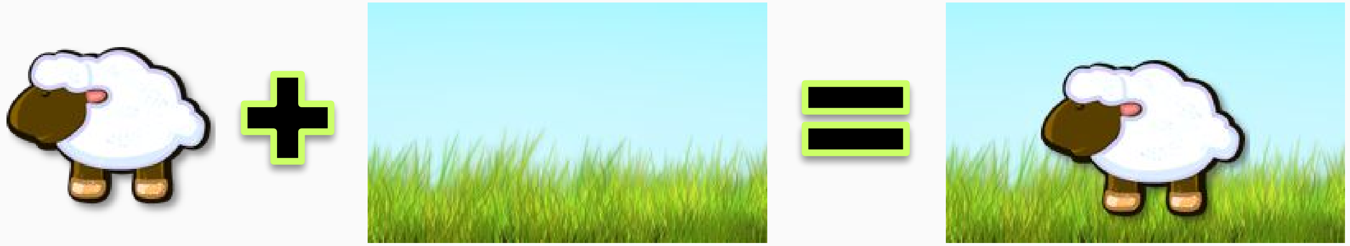
```
/* For Safari 5.1+ */
background: -webkit-linear-gradient(blue, yellow);

/* For Opera 11.1+ */
background: -o-linear-gradient(blue, yellow);

/* For Firefox 3.6+ */
background: -moz-linear-gradient(blue, yellow);

/* Standard syntax */
background: linear-gradient(blue, yellow);
```



## Multiple backgrounds

- CSS3 allows multiple background images
- Simple comma-separated list of images
- Comma separated list for the other properties

```
background-image: url(sheep.png), url(grass.png);
```



# Borders

## Borders

- `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`
- Each property can be defined separately for left, top, bottom and right
    - `border-top-style`, `border-left-color`, ...

## Border image

- Defines an image to be used instead of the normal border of an element
- Split up into a couple of properties
- Example:
    - The border-image property has 3 parts:

```
        border-image: url(border-image.png) 25% repeat;
```

- More detailed description:
    - https://bitsofco.de/understanding-border-image/

## Border radius

- Allows web developers to easily utilize rounder corners in their design elements
- Widespread browser support
- Syntax:

```
border-*-*-radius: [<length>|<%>][<length>|<%>]?
```

- Example:

```
-moz-border-radius: 15px;
border-radius: 15px;
background-color: #FF00FF;
```

## Box shadow

- Allows to easily implement multiple drop shadows (outer or inner) on box elements
- Specifying values for color, size, blur and offset
- Example:

```
-moz-box-shadow: 10px 10px 5px #888;
-webkit-box-shadow: 10px 10px 5px #888;
box-shadow: 10px 10px 5px #888;
```

# Animations

## Animations

- Works in all webkit browsers
- Example:

```
@keyframes resize {
    0% {...}
    50% {...}
    100% {...}
}
#box {
    animation-name: resize;
    animation-duration: 1s;
    animation-iteration-count: 4;
    animation-direction: alternate;  animation-timing-function:
ease-in-out;
}
```

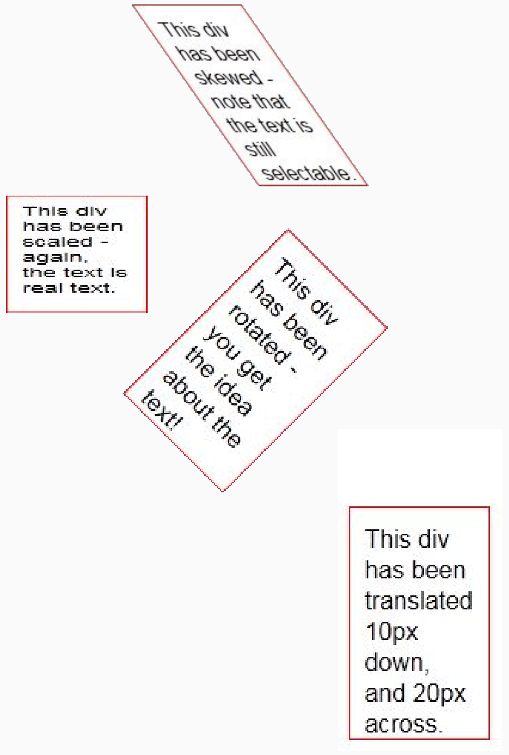
## Transitions

- **Add an effect when changing from one style to another**
- **Different timing functions:**
    - ease, ease-in, ease-out, ease-in-out, linear
- **Example:**

```
#id_of_element {
    -webkit-transition: all 1s ease-in-out;  -moz-transition: all 1s
ease-in-out;
    -o-transition: all 1s ease-in-out;
    -ms-transition: all 1s ease-in-out;  transition: all 1s
ease-in-out;
}
```

# 2D Transformations

- **2D transforms are widely supported**
- **Skew – skew element**
  - transform: skew(35deg);
- **Scale – scale element**
  - transform: scale(1,0.5);
- **Rotate – rotates element**
  - transform: rotate(45deg);
- **Translate – moves element**
  - transform: translate(10px, 20px);



This div has been skewed - note that the text is still selectable.

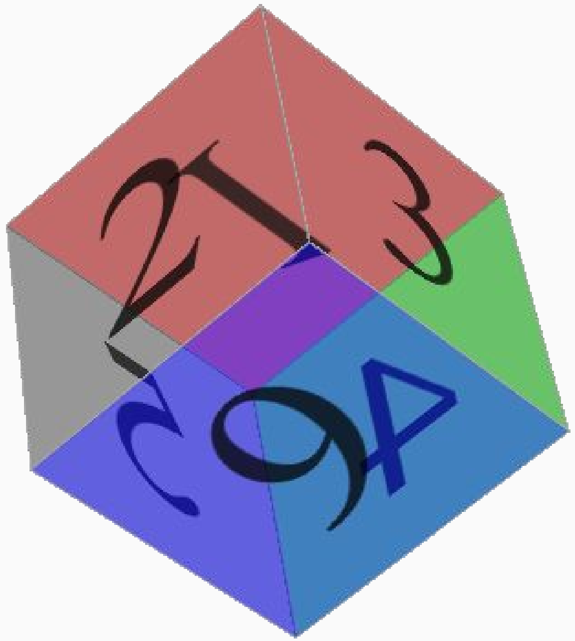
This div has been scaled - again, the text is real text.

This div has been rotated - you get the idea about the text!

This div has been translated 10px down, and 20px across.

## 3D Transformations

- 3D transforms are similar to 2D transforms
- Only work in Safari and Chrome
- X, Y and Z transformation
  - transform: rotateX(180deg);
  - transform: rotateY(180deg);
  - transform: rotateZ(180deg);
  - perspective: 800;
  - perspective-origin: 50% 100px;
  - translate3d, scale3d



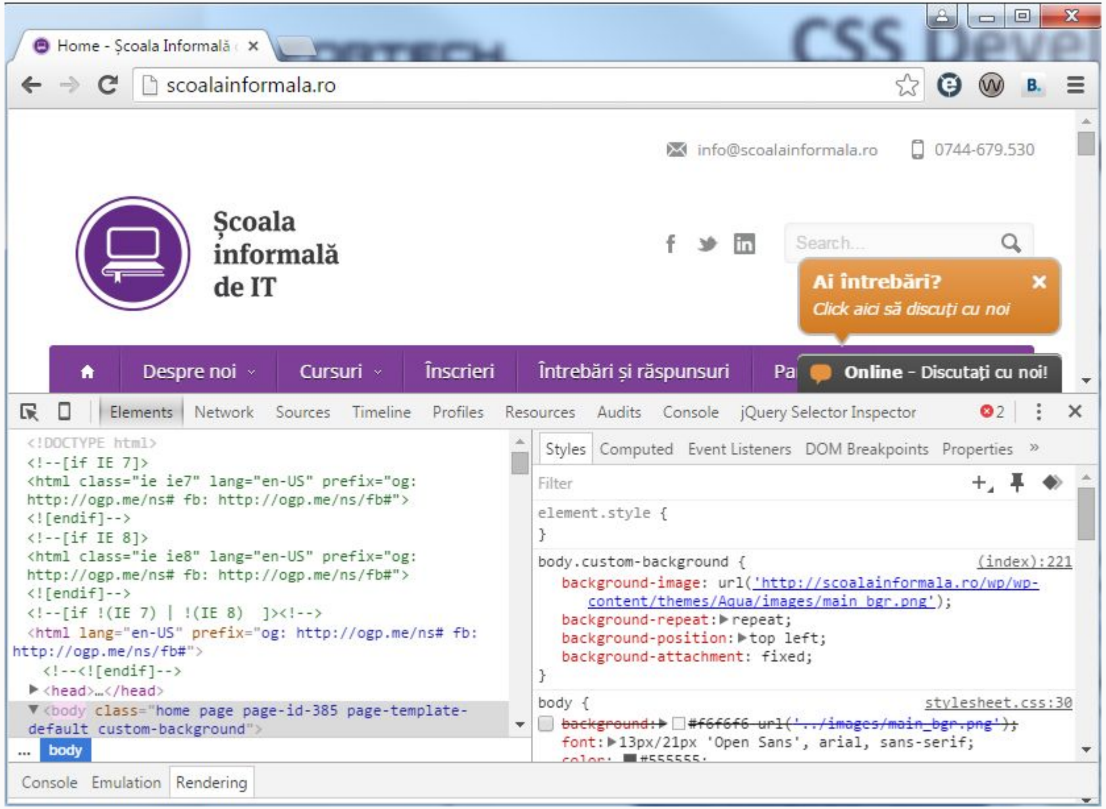
## Benefits of using  CSS

- 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

## CSS Development Tools

- **Chrome Dev Tools**

**– used to examine and adjust CSS and HTML**

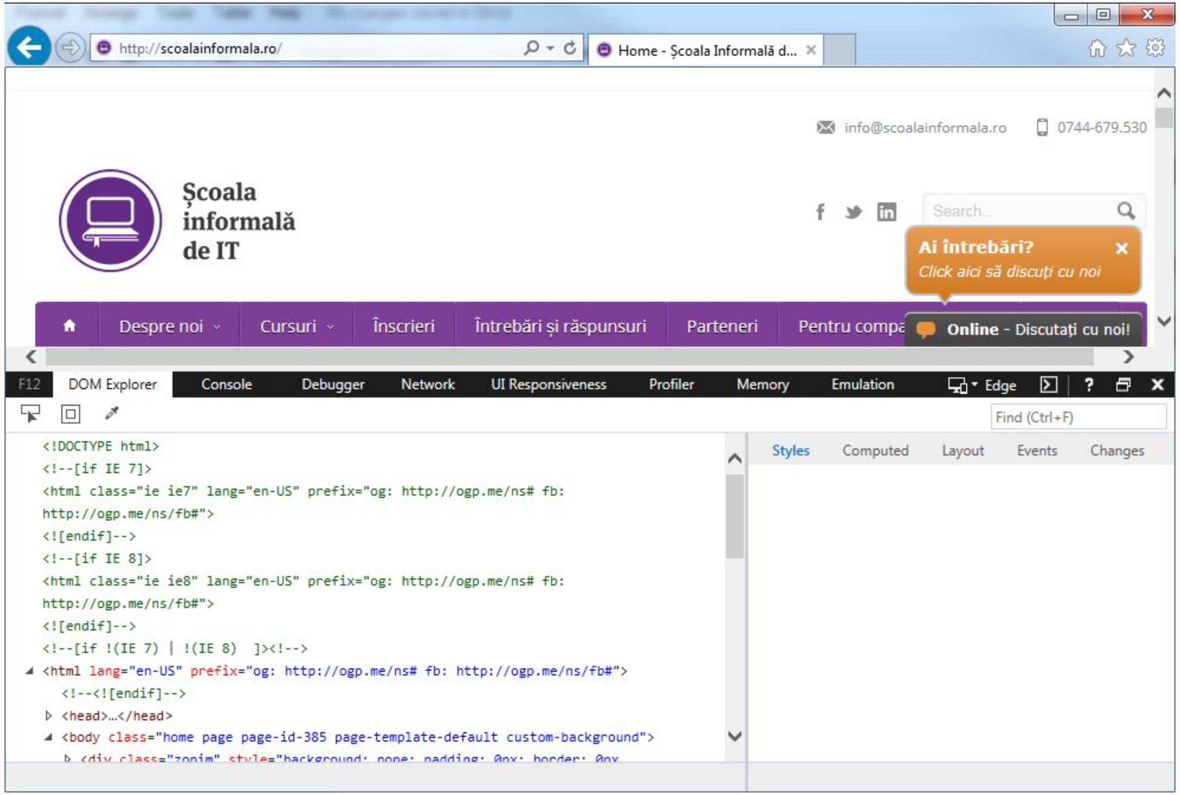


```
<!DOCTYPE html>
<!--[if IE 7]>
<html class="ie ie7" lang="en-US" prefix="og:
http://ogp.me/ns# fb: http://ogp.me/ns/fb#">
<![endif]-->
<!--[if IE 8]>
<html class="ie ie8" lang="en-US" prefix="og:
http://ogp.me/ns# fb: http://ogp.me/ns/fb#">
<![endif]-->
<!--[if !(IE 7) | !(IE 8)  ]><!-->
<html lang="en-US" prefix="og: http://ogp.me/ns# fb:
http://ogp.me/ns/fb#">
  <!--<![endif]-->
 ▶<head>…</head>
 ▼<body class="home page page-id-385 page-template-
default custom-background">
```

```
element.style {
}
body.custom-background {                                   (index):221
    background-image: url('http://scoalainformala.ro/wp/wp-
      content/themes/Aqua/images/main_bgr.png');
    background-repeat:▶repeat;
    background-position:▶top left;
    background-attachment: fixed;
}
body {                                                stylesheet.css:30
    ~~background:▶#f6f6f6 url('../images/main_bgr.png');~~
    font:▶13px/21px 'Open Sans', arial, sans-serif;
    color: #555555;
```

## CSS Development Tools

- **IE Developer Toolbar – to examine CSS and HTML (press [F12])**



```
<!DOCTYPE html>
<!--[if IE 7]>
<html class="ie ie7" lang="en-US" prefix="og: http://ogp.me/ns# fb:
http://ogp.me/ns/fb#">
<![endif]-->
<!--[if IE 8]>
<html class="ie ie8" lang="en-US" prefix="og: http://ogp.me/ns# fb:
http://ogp.me/ns/fb#">
<![endif]-->
<!--[if !(IE 7) | !(IE 8)  ]><!-->
◢ <html lang="en-US" prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb#">
    <!--<![endif]-->
  ▷ <head>…</head>
  ◢ <body class="home page page-id-385 page-template-default custom-background">
    ▷ <div class="zopim" style="background: none; padding: 0px; border: 0px
```