Școala
informală
de IT

# HTML

## Tags, Attributes, Form, Tables, Forms, Validation, Audio, Video, SVG, Canvas

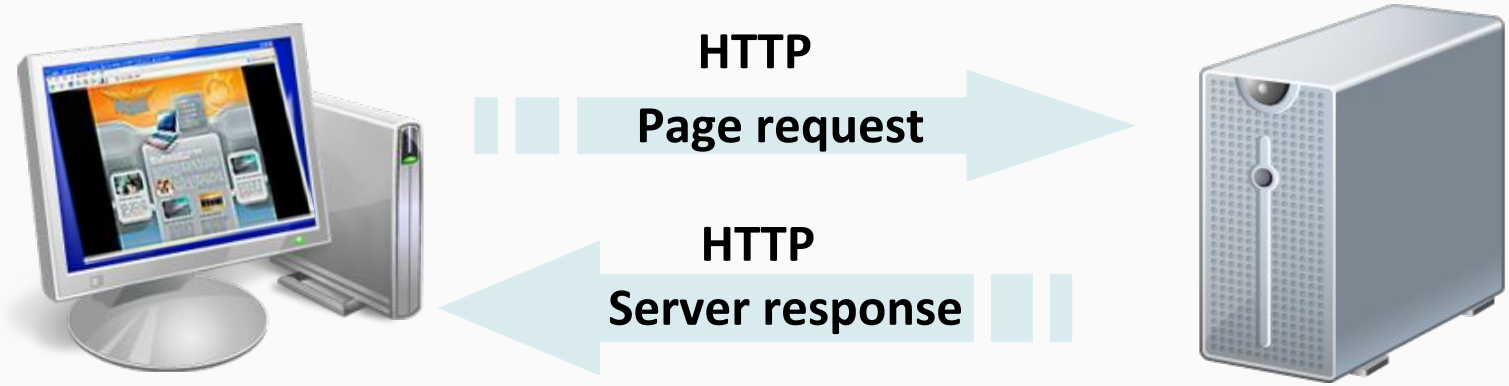
# Agenda

- Introduction to HTML
- New Tags
  - Audio and Video Tags
  - Semantic Tags (<header>, <footer>, …)
  - New Form Elements
- Form Validation and new attributes
- Canvas vs SVG

# Introduction to HTML

# How the Web Works ?

- WWW use classical client / server architecture
  - HTTP is text-based request-response protocol



# What is a Web Page

- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
  - A notation for describing
    - document structure (semantic markup)
    - formatting (presentation markup)
- The markup tags provide information about the page content structure

# Creating HTML Pages

- An HTML file must have an .htm or .html file extension
- HTML files can be created with editors:
  - Notepad, Notepad++ etc.
- Or HTML editors (WYSIWYG Editors):
  - Visual Studio
  - Macromedia Dreamweaver
  - Microsoft Word

# HTML Basics

## Text, Images, Tables, Forms

# HTML Structure

- HTML consists in "elements" and "tags"
  - Begins with `<html>` and ends with `</html>`
- Elements are nested one inside another:

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

- Elements have tags and attributes:

```
<img src="logo.jpg" alt="logo" />
```

- HTML describes structure using two main sections: `<head>` and `<body>`

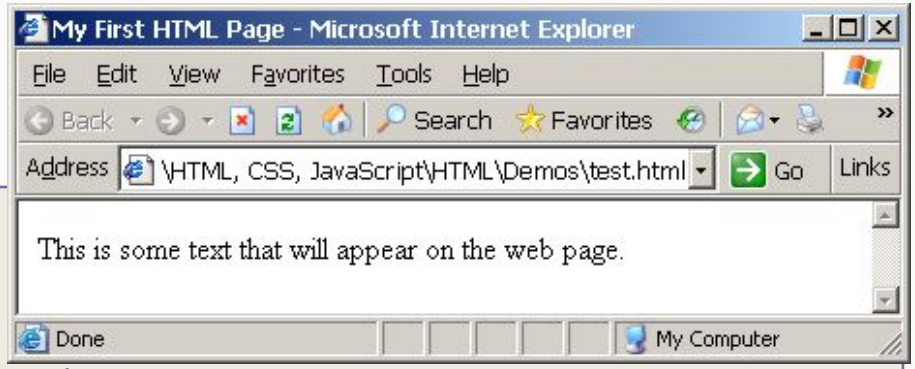
# HTML Code Formatting

- The HTML source code should be further improved and formatted to increase readability and facilitate debugging.
  - Every block element should start on a new line.
  - Every nested (block) element should be indented.
  - Browsers ignore multiple whitespaces in the page source, so formatting is harmless.
- For performance reasons, formatting can be sacrificed

# First HTML Page

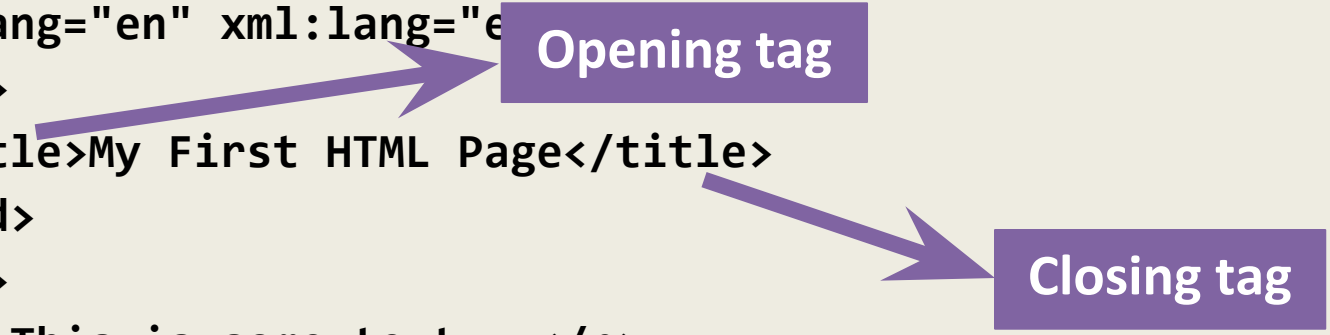
**test.html**

```
<!DOCTYPE HTML>
<html lang="en" xml:lang="en">
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text that will appear on the web page.</p>
  </body>
</html>
```



# First HTML Page: Tags

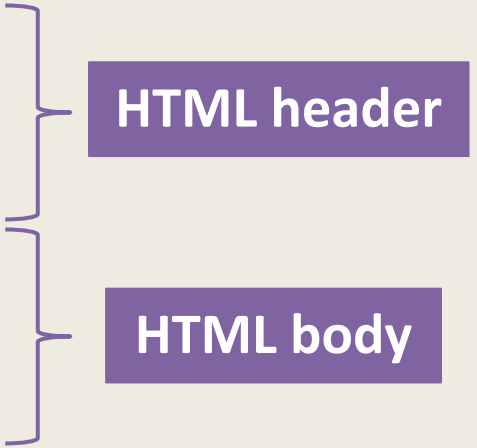
```
<!DOCTYPE HTML>
<html lang="en" xml:lang="e
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```




**An HTML element consists of an opening tag, a closing tag and the content inside.**

# First HTML Page: Header and Body

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```




**The header can contain other information like applied CSS source paths, script links etc.**

# Tags Attributes

- Tags can have attributes
  - Attributes specify properties and behavior

Attribute `alt` with value *"logo"*

```
<img src="logo.gif" alt="logo" />
```

  - Few attributes can apply to every element:
    - `id`, `style`, `class`, `title`
    - The `id` is unique in the document
    - Content of `title` attribute is displayed as hint when the element is hovered with the mouse
    - Some elements have obligatory attributes

# HTML Document Structure in Depth

# The <!DOCTYPE> Declaration

- HTML documents must start with a document type definition (DTD)
  - DTD tells web browsers what type is the served code
  - Example for HTML 5:

```
<!DOCTYPE html>
```

# The <head> Section

- Contains information that doesn't show directly on the viewable page
- Starts after the `<!doctype>` declaration
- Begins with `<head>` and ends with `</head>`
- Contains `<title>` tag
- Can contain some other tags, e.g.
  - `<meta>`
  - `<script>`
  - `<style>`
  - `<!-- comments -->`

# <title> tag

- Title should be placed between <head> and </head> tags

```
<title>Home - Scoala Informala</title>
```

- Used to specify a title in the window title bar
- Search engines and people rely on titles

# <head> Section: <meta>

- Meta tags additionally describe the content contained within the page

```
<meta name="description" content="HTML tutorial" />

<meta name="keywords" content="html, web design, styles" />

<meta name="author" content="John Smith" />

<meta http-equiv="refresh" content="5; url=http://www.google.ro"/>
```

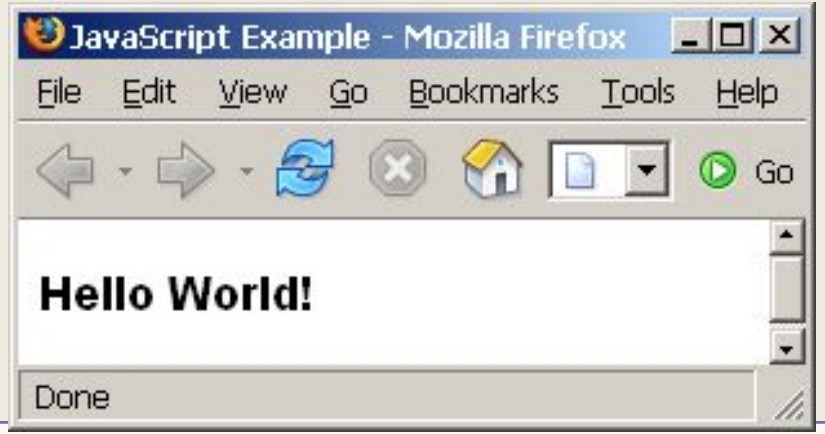
# <head> Section: <script>

- The <script> element is used to embed scripts into an HTML document
  - Script are executed in the client's Web browser
  - Scripts can live in the <head> and in the <body> sections
- Supported client-side scripting languages:
  - JavaScript (it is not Java!)
  - ECMAScript (standard)

# The <script>Tag - Example

**scripts-example.html**

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>JavaScript Example</title>
    <script>
      function sayHello() {
        document.write("<p>Hello World!<\/p>");
      }
    </script>
  </head>
  <body>
    <script>
      sayHello();
    </script>
  </body>
</html>
```

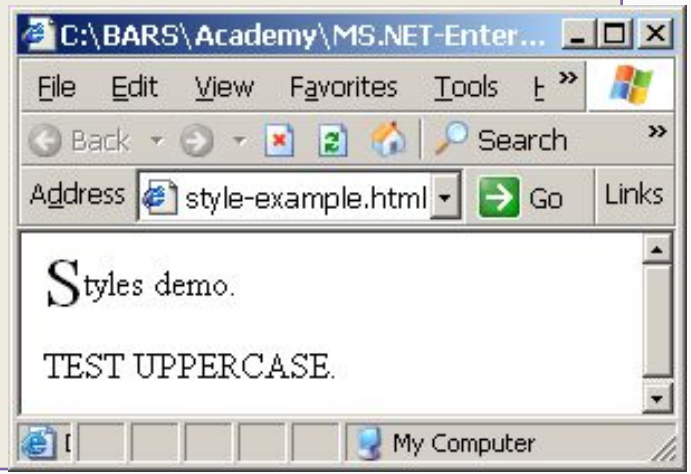


# The <script>Tag - Example

- The <style> element embeds formatting information (CSS styles) into an HTML page

```
<html>
  <head>
    <style>
      p { font-size: 12pt; line-height: 12pt; }
      p:first-letter { font-size: 200%; }
      span { text-transform: uppercase; }
    </style>
  </head>
  <body>
    <p>Styles demo.<br />
       <span>Test uppercase</span>.
    </p>
  </body>
</html>
```

**style-example.html**



# Comments: <!-- --> Tag

- Comments can exist anywhere between the <html></html> tags
- Comments start with <!-- and end with -->

```
<!-- Logo (a JPG file) -->
<img src="logo.jpg" alt="Logo">
<!-- Hyperlink to the web site -->
<a href="http://google.com/">Google</a>
<!-- Show the news table -->
<table class="newstable">
...
```

# <body> Section: Introduction

- The <body> section describes the viewable portion of the page
- Starts after the <head> section

```
<html>
  <head>
    <title>Test page</title>
  </head>
  <body>
    <!-- This is the Web page body -->
  </body>
</html>
```

# Hyperlinks: <a> Tag

- Link to a document called `form.html` on the same server in the same directory:

```
<a href="form.html">Fill Our Form</a>
```

- Link to a document called `parent.html` on the same server in the parent directory:

```
<a href="../parent.html">Parent</a>
```

- Link to a document called `cat.html` on the same server in the subdirectory stuff:

```
<a href="stuff/cat.html">Catalog</a>
```

# Hyperlinks: <a> Tag

- Link to an external Web site:

```
<a href="http://www.nasa.org" target="_blank">NASA</a>
```

  - Always use a full URL, including "http://", not just "www.somesite.com"
  - Using the target="_blank" attribute opens the link in a new window

- Link to an e-mail address:

```
<a href="mailto:bugs@example.com?subject=Bug+Report">
Please report bugs here (by e-mail only)</a>
```

# Hyperlinks: <a> Tag

- Link to a document called `apply-now.html`
  - On the same server, in same directory
  - Using an image as a link button:

```
<a href="apply-now.html"><img src="apply-now-button.jpg" /></a>
```

- Link to a document called `index.html`
  - On the same server, in the subdirectory `english` of the parent directory:

```
<a href="../english/index.html">Switch to English version</a>
```

# Hyperlinks: <a> Tag

- Link to another location in the same document:

```
<a href="#section1">Go to Introduction</a>
...
<h2 id="section1">Introduction</h2>
```

- Link to a specific location in another document:

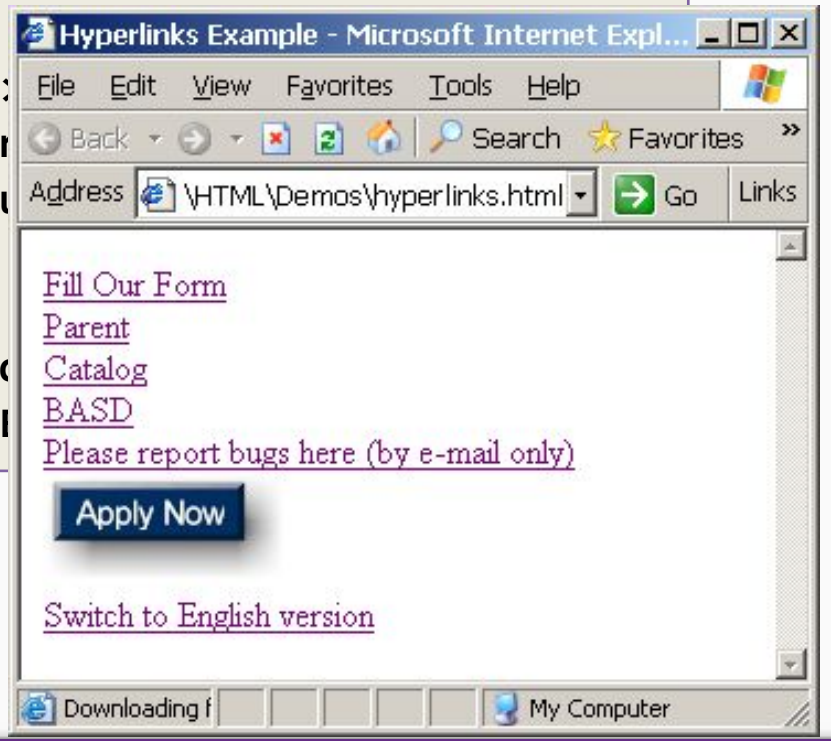
```
<a href="chapter3.html#section3.1.1">Go to Section 3.1.1</a>

<!-- In chapter3.html -->
...
<div id="section3.1.1">
  <h3>3.1.1. Technical Background</h3>
</div>
```

# Hyperlinks - Example

**hyperlinks.html**

```
<a href="form.html">Fill Our Form</a> <br />
<a href="../parent.html">Parent</a> <br />
<a href="stuff/cat.html">Catalog</a> <br />
<a href="http://www.nasa.org" target="_blan
<a href="mailto:bugs@example.com?subject=Bu
bugs here (by e-mail only)</a>
<br />
<a href="apply-now.html"><img src="apply-no
<a href="../english/index.html">Switch to E
```



# Hyperlinks - Example

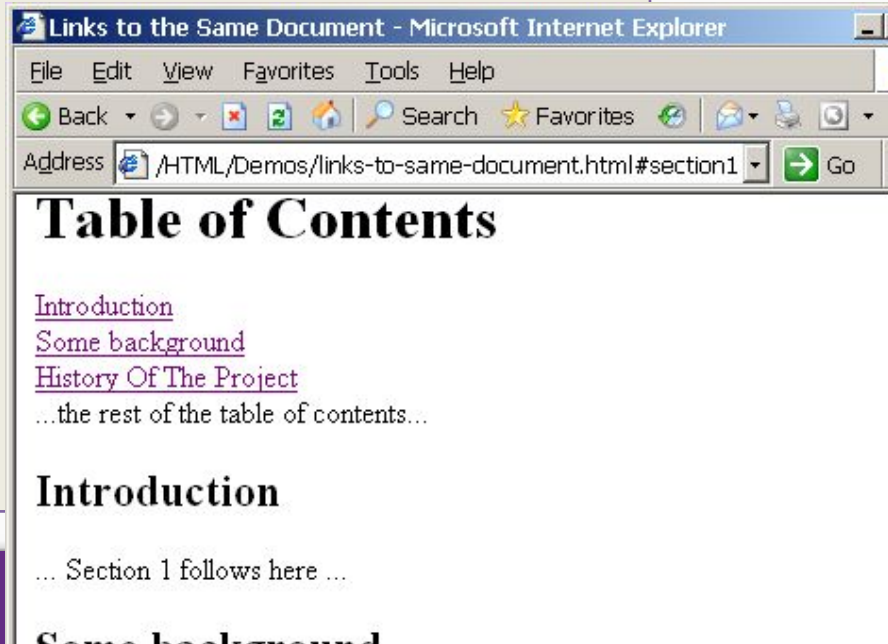
links-to-same-document.html

```
<h1>Table of Contents</h1>

<p><a href="#section1">Introduction</a><br />
<a href="#section2">Some background</A><br />
<a href="#section2.1">Project History</a><br />
...the rest of the table of contents...

<!-- The document text follows here -->

<h2 id="section1">Introduction</h2>
... Section 1 follows here ...
<h2 id="section2">Some background</h2>
... Section 2 follows here ...
<h3 id="section2.1">Project History</h3>
... Section 2.1 follows here ...
```



# Headings and Paragraphs

- Heading Tags (h1 – h6)

```
<h1>Heading 1</h1>
<h2>Sub heading 2</h2>
<h3>Sub heading 3</h3>
```

- Paragraph Tags

```
<p>This is my first paragraph</p>
<p>This is my second paragraph</p>
```

- Sections: `div` and `span`

```
<div style="background: skyblue;">
   <span style="color: red;">This is</span> a div</div>
```

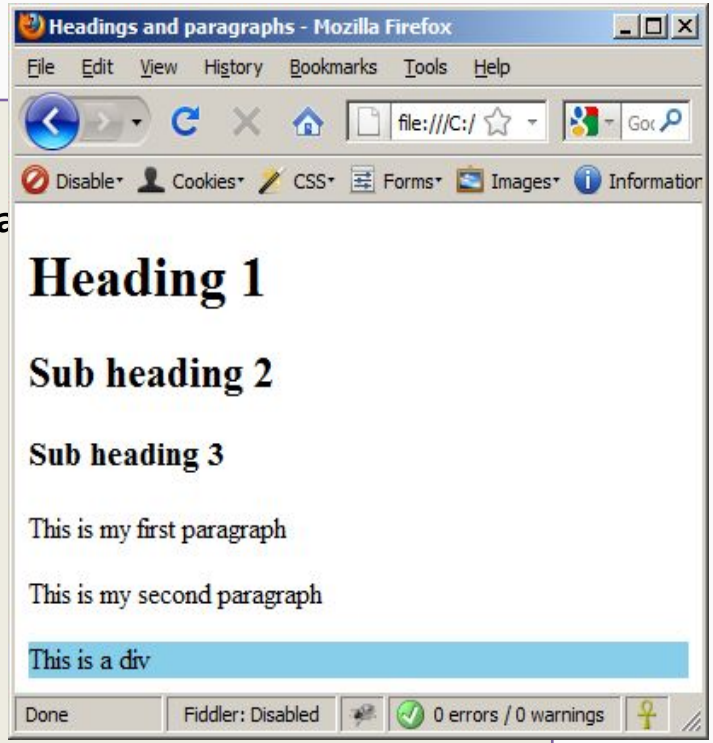
# Headings and Paragraphs

## headings.html

```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and paragraphs</title></hea
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

    <p>This is my first paragraph</p>
    <p>This is my second paragraph</p>

    <div style="background:skyblue">
      This is a div</div>
  </body>
</html>
```



# Images: <img> Tag

- Inserting an image with <img> tag:

```
<img src="/img/some-logo.png">
```

- Image attributes:

| | |
|---|---|
| src | Location of image file (relative or absolute) |
| alt | Substitute text for display (e.g. in text mode) |
| height | Number of pixels of the height |
| width | Number of pixels of the width |
| border | Size of border, 0 for no border |

- Example:

```
<img src="./logo.png" alt="Some Logo" />
```

# Ordered Lists: <ol> Tag

- Create an Ordered List using <ol></ol>:

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Pear</li>
</ol>
```

- Attribute values for type are

1. Apple
2. Orange
3. Pear

**1**

a. Apple
b. Orange
c. Pear
A. Apple
B. Orange
C. Pear

**A, a**

i. Apple
ii. Orange
iii. Pear
I. Apple
II. Orange
III. Pear

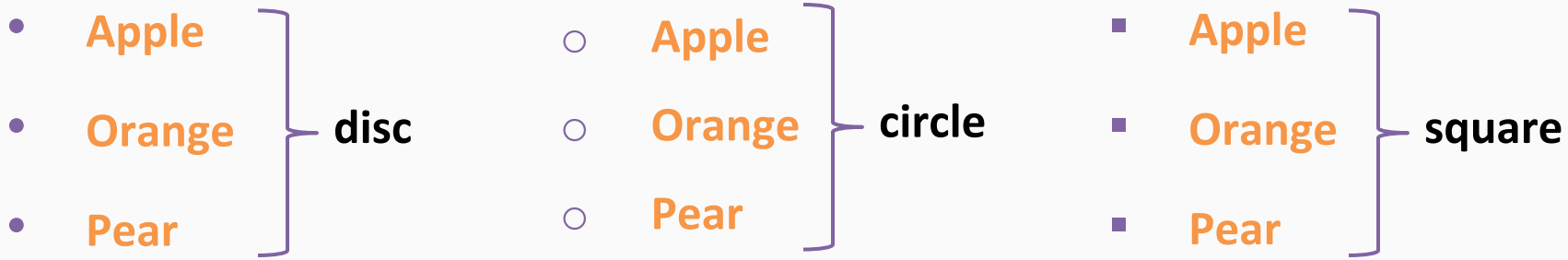
**I, i**

# Unordered Lists: <ul> Tag

- Create an Unordered List using <ul></ul>:

```
<ul type="disc">
  <li>Apple</li>
  <li>Orange</li>
  <li>Pear</li>
</ul>
```

- Attribute values for type are:



# Definition lists: <dl> tag

- Create definition lists using <dl>
  - Pairs of text and associated definition; text is in <dt> tag, definition in <dd> tag

```
<dl>
    <dt>HTML</dt>
    <dd>A markup language ...</dd>
    <dt>CSS</dt>
    <dd>Language used to ...</dd>
</dl>
```

  - Renders without bullets
  - Definition is indented

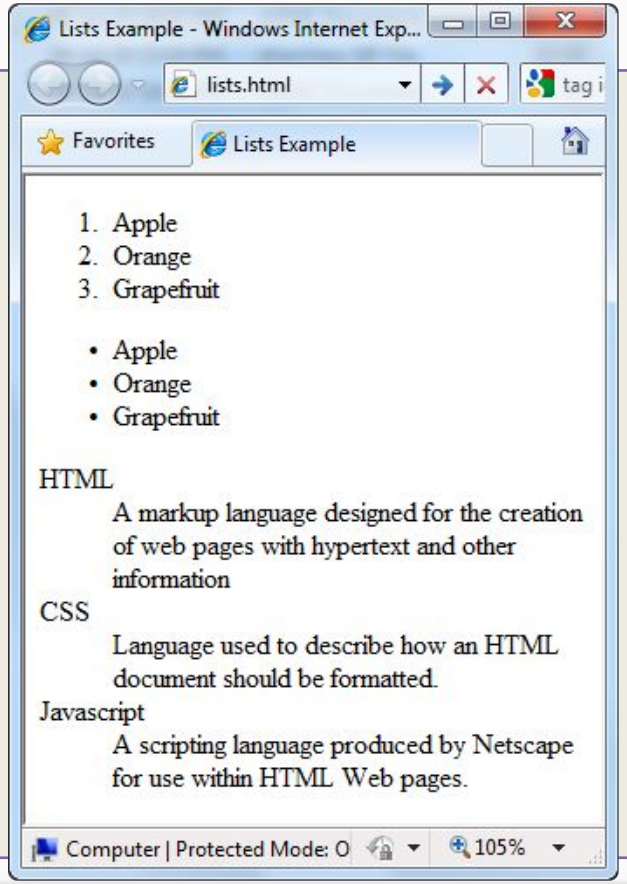
# Lists

**lists.html**

```
<ol type="1">
    <li>Apple</li>
    <li>Orange</li>
    <li>Grapefruit</li>
</ol>

<ul type="disc">
    <li>Apple</li>
    <li>Orange</li>
    <li>Grapefruit</li>
</ul>

<dl>
    <dt>HTML</dt>
    <dd>A markup lang...</dd>
</dl>
```



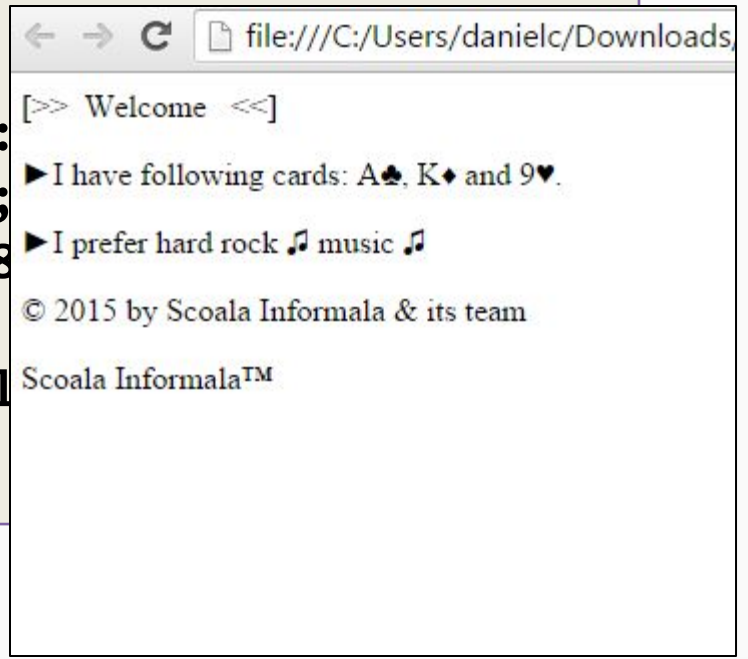
# HTML Special Characters

| Symbol Name | HTML Entity | Symbol |
|---|---|---|
| Copyright Sign | © | © |
| Registered Trademark Sign | ® | ® |
| Trademark Sign | ™ | ™ |
| Less Than | < | < |
| Greater Than | > | > |
| Ampersand | & | & |
| Non-breaking Space |   | |
| Em Dash | — | — |
| Quotation Mark | " | " |
| Euro | € | € |
| British Pound | £ | £ |

# Special Characters

**special-chars.html**

```
<p>[&gt;&gt;  Welcome
      &lt;&lt;]</p>
  <p>&#9658;I have following cards:
    A&#9827;, K&#9830; and 9&#9829;
  <p>&#9658;I prefer hard rock &#98
    music &#9835;</p>
  <p>&copy; 2015 by Scoala Informal
  <p>Scoala Informala&trade;</p>
```



# Using <DIV> and <SPAN>

## Block and Inline Elements

# Block and Inline Elements

- Block elements add a line break before and after them
  - `<div>` is a block element
  - Other block elements are `<table>`, `<hr>`, headings, `<ol>`, `<ul>`, `<dl>`, `<hr>`, `<p>` and etc.
- Inline elements don't break the text before and after them
  - `<span>` is an inline element
  - Most HTML elements are inline, e.g. `<a>`, `<input>`, `<label>`, `<select>`

# The  Tag

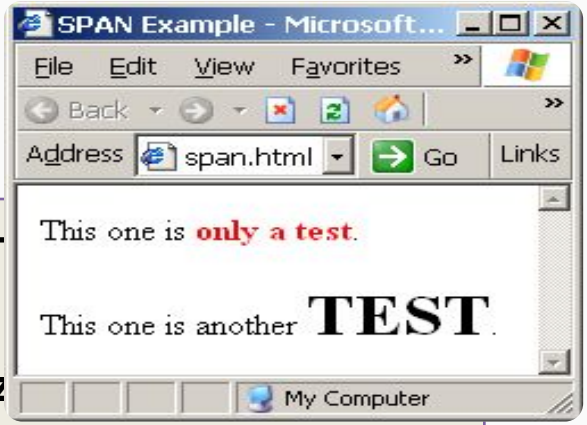
-  creates logical divisions within a page
- Block style element
- Used with CSS
- Example:
  div-and-span.html


```
<div style="font-size:24px; color:red">DIV example</div>

<p>This one is <span style="color:red; font-weight:bold">only
a test</span>.</p>
```

# The  Tag

- Inline style element
- Useful for modifying a specific portion of text
  - Don't create a separate area (paragraph) in the document
- Very useful with CSS




**div-and-span.html**

```
<p>This one is <span style="color:red; font-
a test</span>.</p>

<p>This one is another <span style="font-siz
font-weight:bold">TEST</span>.</p>
```

# HTML Tables

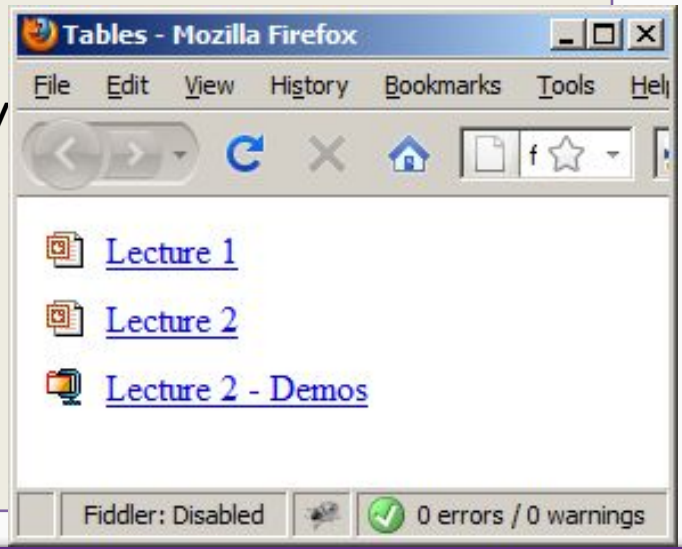
# HTML Tables

- Tables represent tabular data
  - A table consists of one or several rows
  - Each row has one or more columns
- Tables comprised of several core tags:

  `<table></table>`: begin / end the table
  `<tr></tr>`: create a table row
  `<td></td>`: create tabular data (cell)
- Tables should not be used for layout. Use CSS floats and positioning styles instead

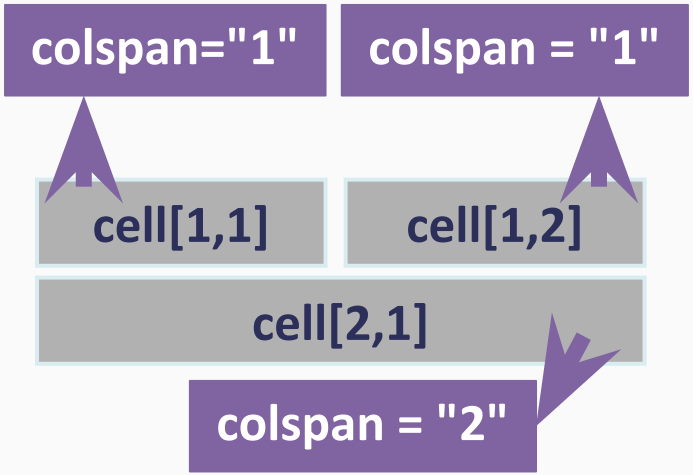
# Simple HTML Tables

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture2.ppt">Lecture 2</
  </tr>
  <tr>
    <td><img src="zip.gif"></td>
    <td><a href="lecture2-demos.zip">
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```

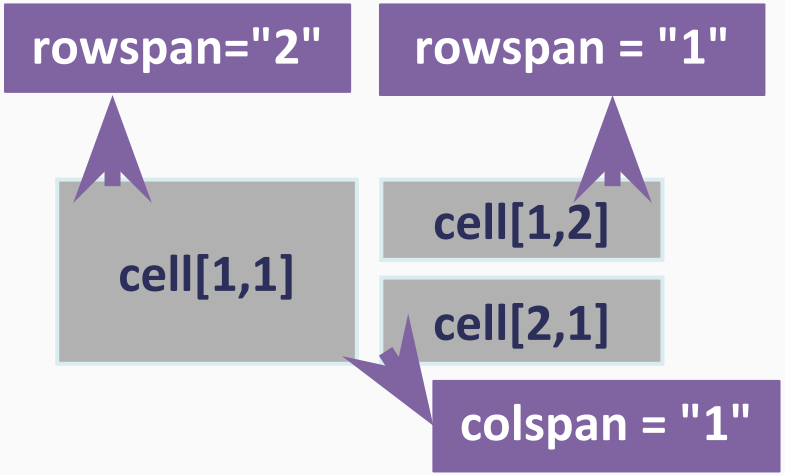


# Column and Row Span

- Tables have two important attributes:
  - `colspan`
  - `rowspan`




Defines how many columns the cell occupies

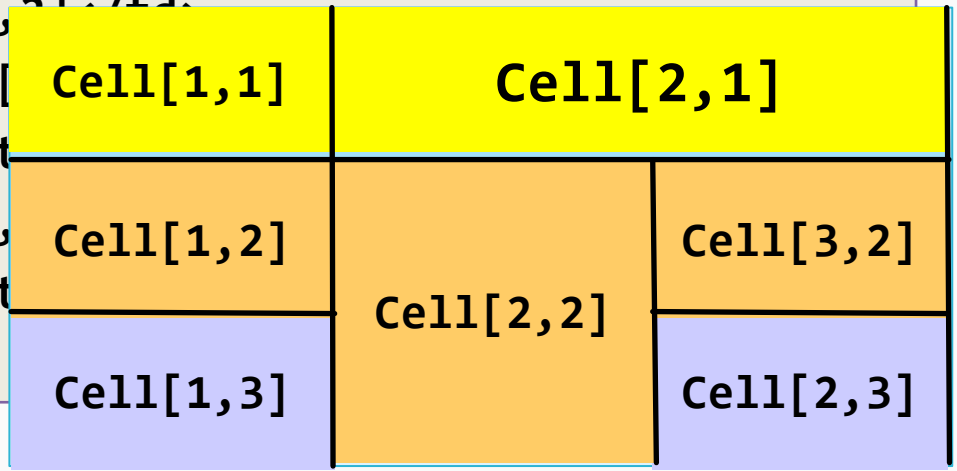



- Defines how many rows the cell occupies

# Column and Row Span

table-colspan-rowspan.html

```
<table cellspacing="0">
    <tr class="1"><td>Cell[1,1]</td>
        <td colspan="2">Cell[2,1]</td></tr>
    <tr class="2"><td>Cell[1,
        <td rowspan="2">Cell[
        <td>Cell[3,2]</td></t
    <tr class="3"><td>Cell[1,
        <td>Cell[2,3]</td></t
</table>
```



# Column and Row Span

- **Do not to try to code complex table by hand, use online table generators instead**
  - http://www.tablesgenerator.com/html_tables

# HTML Forms

## Entering User Data from a Web Page

# HTML Forms

- Forms are the primary method for gathering data from site visitors
- Create a form block with

```
<form></form>
```

# Form Fields

- Single-line text input fields:

```
<input type="text" name="FirstName" value="This is a text field" />
```

- Multi-line textarea fields:

```
<textarea name="Comments">This is a multi-line text field</textarea>
```

- Hidden fields contain data not shown to the user:

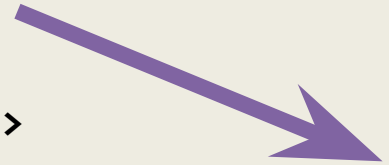
```
<input type="hidden" name="Account" value="This is a hidden text field" />
```

  - Often used by JavaScript code

# Fieldsets and legends

- Fieldsets are used to enclose a group of related form fields:

```
<form>
   <fieldset>
      <legend>Client Details</legend>
      <input type="text" id="Name" />
      <input type="text" id="Phone" />
   </fieldset>
   <fieldset>
      <legend>Order Details</legend>
      <input type="text" id="Quantity" />
      <textarea cols="40" rows="10"
         id="Remarks"></textarea>
   </fieldset>
</form>
```



The <legend> is the fieldset's title

# Form Input Controls

- Checkboxes:

```
<input type="checkbox" name="fruit" value="apple" />
```

- Radio buttons:

```
<input type="radio" name="title" value="Mr." />
```

- Radio buttons can be grouped, allowing only one to be selected from a group:

```
<input type="radio" name="city" value="Cluj" />
<input type="radio" name="city" value="Huedin" />
```

# Other Form Controls

- Dropdown menu

```
<select name="level">
  <option value="1" selected>Beginner</option>
  <option value="2">Intermediate</option>
  <option value="3">Advanced</option>
</select>
```

- Button – used for Javascript, no default action

```
<input type="button" value="click me" />
```

# Other Form Controls

- Submit button – submits the content to server

```
<input type="submit" name="submit_button" value="Apply Now" />
```

- Reset button – brings the form to its initial state for values

```
<input type="reset" name="resetBtn" value="Reset the form" />
```

- Image button – acts like submit but image is displayed and click coordinates are sent

```
<input type="image" src="submit.gif" name="submitBtn"
alt="Submit" />
```

# Other Form Controls

- Password input – a text field which masks the entered text with *

```
<input type="password" name="pass" />
```

- Multiple select field – displays the list of items in multiple lines, instead of one

```
<select name="products" multiple>
  <option value="1" selected>keyboard</option>
  <option value="2">mouse</option>
  <option value="3">speakers</option>
</select>
```

# Labels

- Form labels are used to associate an explanatory text to a form field using the field's ID.

```
<label for="first_name">First Name</label>
<input type="radio" id="first_name" />
```

- Clicking on a label focuses its associated field (checkboxes are toggled, radio buttons are checked)
- Labels are both a usability and accessibility feature and are required in order to pass accessibility validation.

# HTML Forms - Example

**form.html**

```
<form>
  <input name="subject" type="hidden" value="Class" />
  <fieldset><legend>Academic information</legend>
    <label for="degree">Degree</label>
    <select name="degree" id="degree">
      <option value="BA">Bachelor of Art</option>
      <option value="BS">Bachelor of Science</option>
      <option value="MBA" selected="selected">Master of
        Business Administration</option>
    </select>
    <br />
```

# HTML Forms - Example

**form.html** *(continued)*

```
    <label for="studentid">Student ID</label>
    <input type="password" name="studentid" />
    <label for="classes">Classes Attended</label>
    <select multiple name="classes" id="classes">
      <option value="1">Geography</option>
      <option value="2">Mathematics</option>
      <option value="3">English</option>
    </select>
  </fieldset>
  <fieldset><legend>Personal Details</legend>
    <label for="fname">First Name</label>
    <input type="text" name="fname" id="fname" />
```

# HTML Forms - Example

**form.html** *(continued)*

```
    <br />
    <label for="lname">Last Name</label>
    <input type="text" name="lname" id="lname" />
    <br />
    Gender:
    <input name="gender" type="radio" id="gm" value="m" />
    <label for="gm">Male</label>
    <input name="gender" type="radio" id="gf" value="f" />
    <label for="gf">Female</label>
    <br />
  <label for="email">Email</label>
  <input type="text" name="email" id="email" />
</fieldset>
```

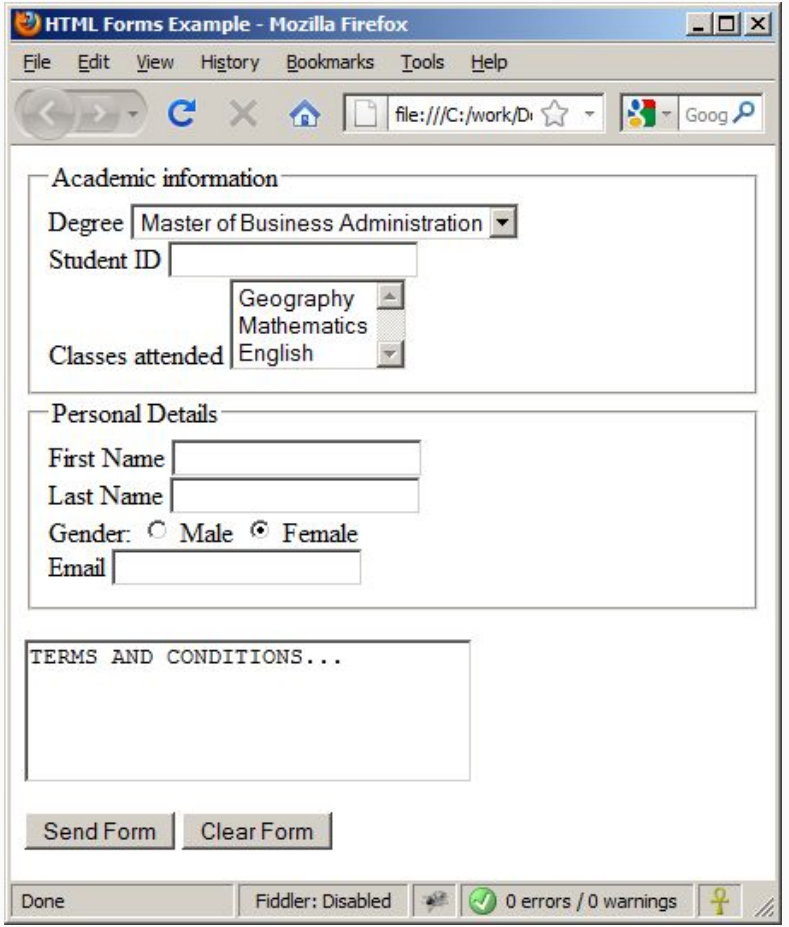
# HTML Forms - Example

**form.html** *(continued)*

```
  <p>
    <textarea name="terms" cols="30" rows="4"
      readonly="readonly">TERMS AND CONDITIONS...</textarea>
  </p>
  <p>
    <input type="submit" name="submit" value="Send Form" />
    <input type="reset" value="Clear Form" />
  </p>
</form>
```

# HTML Forms - Example

**form.html** ***(continued)***



# Tabindex

- The tabindex HTML attribute controls the order in which form fields and hyperlinks are focused when repeatedly pressing the TAB key
  - `tabindex="0"` (zero) - "natural" order
  - If x > y, then elements with tabindex="x" are iterated before elements with tabindex="y"
  - Elements with negative tabindex are skipped, however, this is not defined in the standard

```
<input type="text" tabindex="10" value="tab stop" />
<input type="text" tabindex="-1" value="tab no stop" />
```

# HTML Frames

## <iframe>

# HTML Frames

- Frames provide a way to show multiple HTML documents in a single Web page
- The page can be split into separate views (frames) horizontally and vertically
- Frames were popular in the early ages of HTML development, but now their usage is rejected

# Inline Frames: <iframe>

- Inline frames provide a way to show one website inside another website:

iframe-demo.html

```
<iframe name="iframeGoogle"
    width="600" height="400"
    src="http://www.google.com"
    frameborder="yes"
    scrolling="yes">
</iframe>
```

# Introduction to HTML 5

# What is HTML 5?

- HTML5 – the next major revision of the HTML
  - Currently under development
    - Far away from final version
  - Developed under the HTML 5 working group of the World Wide Web Consortium (W3C) in 2007
  - First Public Working Draft of the specification
    - January 22, 2008
  - Parts of HTML5 are being implemented in browsers before the whole specification is ready

# HTML - Past, Present, Future

- 1991 – HTML first mentioned – Tim Berners-Lee – HTML tags
- 1993 – HTML (first public version, published at IETF)
- 1993 – HTML 2 draft
- 1995 – HTML 2 – W3C
- 1995 – HTML 3 draft
- 1997 – HTML 3.2 – “Wilbur”
- 1997 – HTML 4 – ”Cougar” – CSS
- 1999 – HTML 4.01 (final)
- 2000 – XHTML draft
- 2001 – XHTML (final)
- 2008 – HTML5 / XHTML5 draft
- 2011 – feature complete HTML5
- 2022 – HTML5 – final specification

# HTML 5 Goals

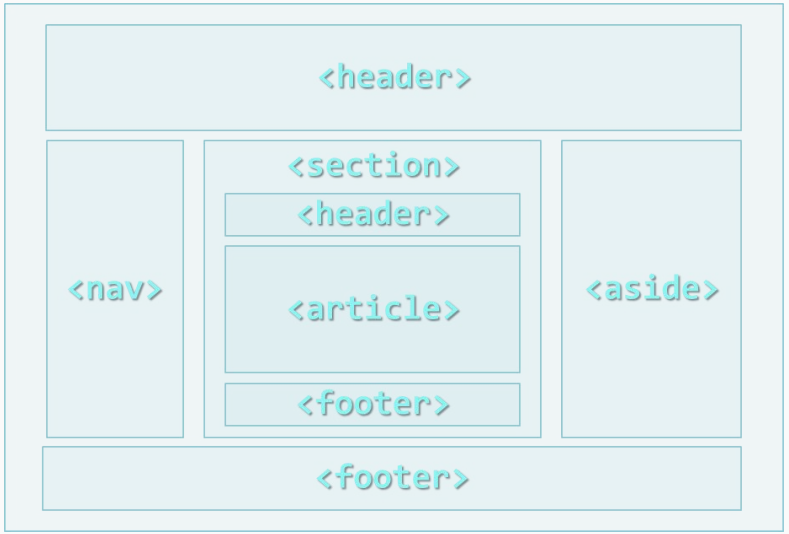
- Latest version is HTML5
  - Aimed to have all of the power of native applications
  - Run on any platform (Windows, Linux, iPhone, Android, etc.)
- New features should be based on HTML, CSS, DOM and JavaScript
- Reduce the need for external plugins
- Better error handling
- More markup to replace scripting

# Tags in HTML 5

## What should we keep in mind with HTML 5

# New Layout Structure

- Better layout structure: Semantic tags
  - <header> = 
  - <nav> = 
  - <section>
  - <article>
  - <aside>
  - <footer>



# New Layout Structure (2)

- Elements like header and footer are not meant to be only at the top and bottom of the page
- Header and footer of each document section
  - Not very different from `<DIV>` tag but are more semantically well defined in the document structure

# New Layout Structure - Exemple

```
<body>

  <header>
    <hgroup>
      <h1>HTML 5 Presentation</h1>
      <h2>New Layout Structure</h2>
    </hgroup>
  </header>

  <nav>
    <ul>
      <li>Lecture</li>
      <li>Demos</li>
      <li>Trainers</il>
    </ul
  </nav>
```

*(continued on next slide)*

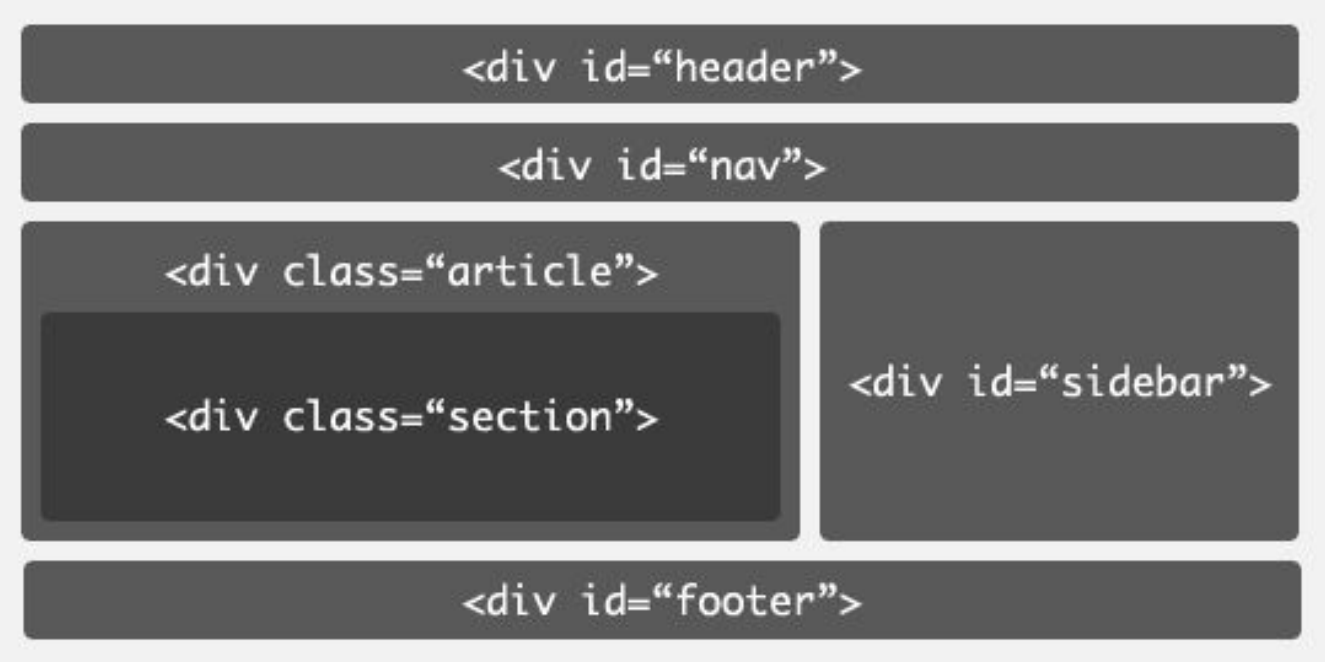
# New Layout Structure - Example(2)

```
  <section>
    <article>
      <header>
        <h1>First Paragraph</h1>
      </header>
      <section>
        Some text
      </section>
    </article>
  </section>

  <aside>
    <a href="http://scoalainformala.ro"> more info</a>
  </aside>

  <footer>
    Done by...
  </footer>
</body>
```

# New Layout Structure using divs



# New tags

# New Tags

- `<article>`
  - For external content, like text from a news-article, blog, forum, or any other external source
- `<aside>`
  - For content aside from (but related to) the content it is placed in
- `<details>`
  - For describing details about a document, or parts of a document
- `<summary>`
  - A caption, or summary, inside the details element

# New Tags (2)

- `<mark>`
  - For text that should be highlighted
- `<nav>`
  - For a section of navigation
- `<section>`
  - For a section in a document (e.g. chapters, headers, footers)
- `<wbr>`
  - Word break.  For defining an appropriate place to break a long word or sentence
- Other tags: `<command>`, `<datalist>`, `<details>`, `<progress>`, etc.

# New Media Tags

- Media Tags
  - `<audio>`
    - Attributes: `autoplay`, `controls`, `loop`, `src`
  - `<video>`
    - Attributes: `autoplay`, `controls`, `loop`, `height`, `width`, `src`

```
<audio width="360" height="240" controls="controls" >
  <source src="someSong.mp3" type="audio/mp3">
  </source>
  Audio tag is not supported
</audio>
```

# Embed Tag - New Syntax

- `<embed>`
  - Defines embedded content, such as a plug-in
- Attributes
  - `src="url", type="type"`

```
<embed src="helloworld.swf" />
```

# Form validation and new attributes

# New Attributes

| Attribute | Description |
|---|---|
| `Autocomplete` | `on`/`off`. In case of “on”, the browser stores the value, auto fill when the user visits the same form next time |
| `Autofocus` | `Autofocus`. Input field is focused on page load |
| `Required` | `Required`. Mandates input field value for the form submit action |
| `Dragabble` | `True`/`false` indicates if the element is dragabble or not |

# New <input> Types

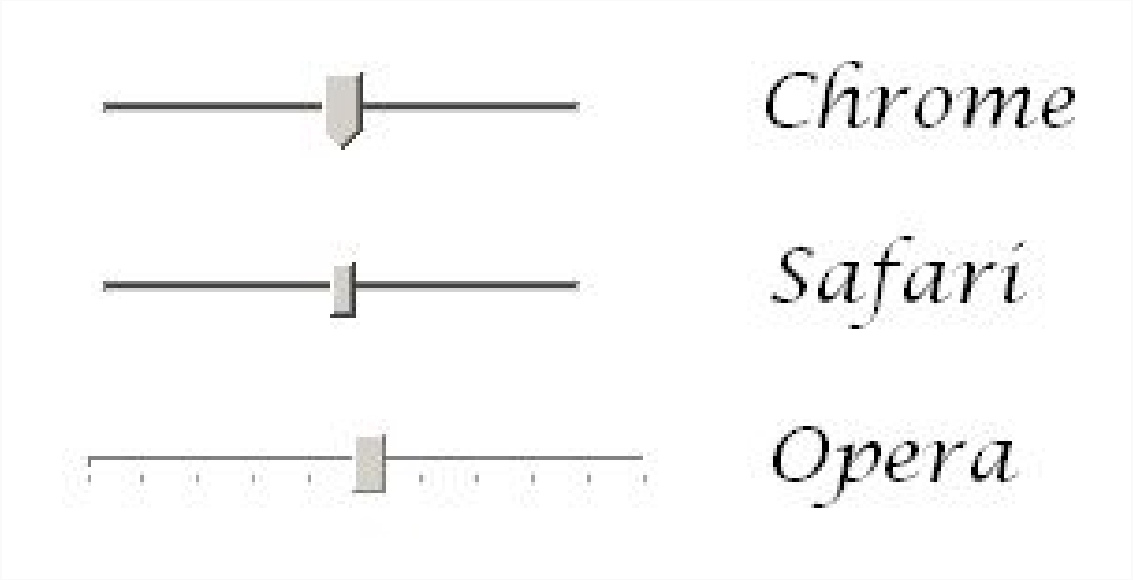
| Attribute | Description |
| --- | --- |
| Number/Range | Restricts users to enter only numbers. Additional attributes min, max and step and value can convert the input to a slider control or a spin box |
| date, month, week, time, datetime, datetime-local | Providers a date picker interface. |
| Email | Input type for Email Addresses |
| URL | Input field for URL address |
| Telephone | Input type for Telephone number |

# New <input> Types

| Attribute | Description |
| --- | --- |
| Color | Specifies that its input element is a color-well control, for setting the element's value to a string representing a simple color |
| File | Specifies that its input element represents a list of file items |
| Search | Has a pattern = pattern<br>Specifies a regular expression against which a UA is meant to check the value of the control represented by its element |

# New Form Tags

- <input type = “range”>



# Canvas vs SVG

# Canvas

- Allows dynamic, scriptable rendering of 2D shapes and bitmap images
- Low level, procedural model
  - Updates a bitmap
  - Does not have a built-in scene graph
- Consists of a drawable region defined in HTML
  - Has height and width attributes
  - Accessible by JavaScript Code
- Used for building graphs, animations, games, and image composition

# Canvas - Example

- In HTML:

```
<canvas id="example" width="200" height="200">
 This is displayed if HTML5 Canvas is not supported.
</canvas>
```

- Access with JavaScript:

```
var example = document.getElementById('example');
var context = example.getContext('2d');
context.fillStyle = "rgb(255,0,0)";
context.fillRect(30, 30, 50, 50);
```

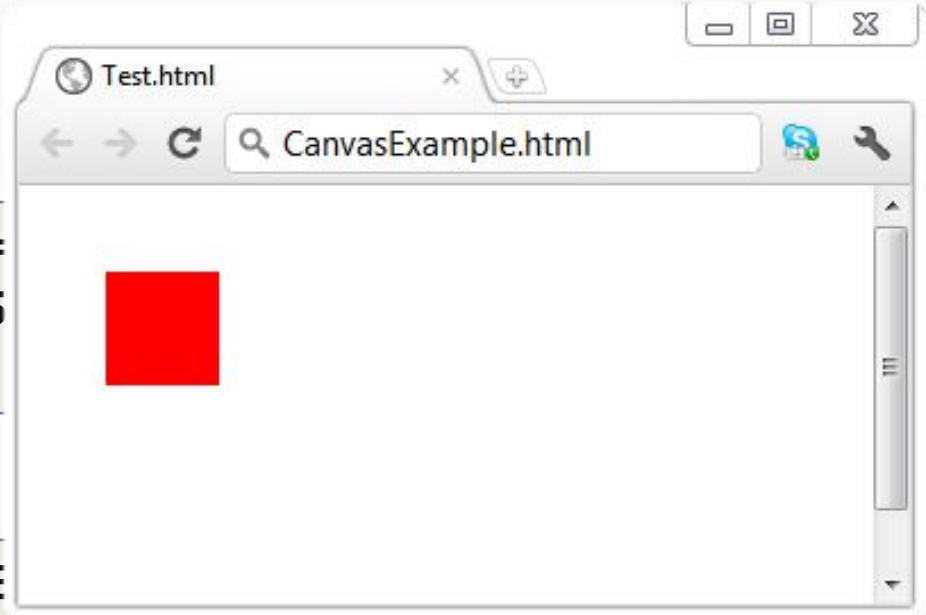
# SVG

- SVG stands for Scalable Vector Graphics
  - A language for describing 2D-graphics
  - Graphical applications
- Most of the web browsers can display SVG just like they can display PNG, GIF, and JPG
- HTML5 allows embeding SVG
  - Directly using `<svg>...</svg>`

# Canvas - Example

- In HTML:

```
<canvas id="example" width=
 This is displayed if HTML5
</canvas>
```

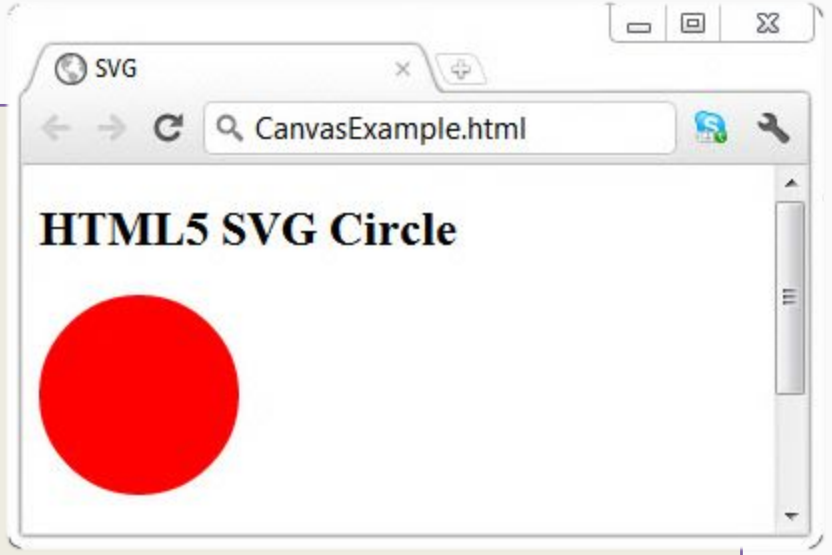



- Access with JavaScript:

```
var example = document.getE
var context = example.getContext('2d');
context.fillStyle = "rgb(255,0,0)";
context.fillRect(30, 30, 50, 50);
```

# SVG - Example

```
<!DOCTYPE html>
<head>
  <title>SVG</title>
  <meta charset="utf-8" />
</head>
<body>
  <h2>HTML5 SVG Circle</h2>
  <svg id="svgelem" height="200"
   xmlns="http://www.w3.org/2000/svg">
    <circle id="redcircle" cx="50" cy="50" r="50"
     fill="red" />
  </svg>
</body>
</html>
```



# SVG Gradient - Example

```
<svg id="svgelem" height="200" xmlns="http://www.w3.org/2000/svg">
  <defs>
    <radialGradient id="gradient" cx="50%" cy="50%"
     r="50%" fx="50%" fy="50%">
      <stop offset="0%"
       style="stop-color:rgb(200,200,200);stop-opacity:0"/>
      <stop offset="100%"
       style="stop-color:rgb(0,0,255);stop-opacity:1"/>
    </radialGradient>
  </defs>
  <ellipse cx="100" cy="50" rx="100" ry="50"
   style="fill:url(#gradient)" />
</svg>
```

# Summary

# HTML 5 - Summary (1)

- Provides a very rich user experience without Plug-ins
- RIA replacement?
- Better Performance
- Leverages GPU for better graphical experience
- HTML5 is not a thing someone can detect
  - It consists of many elements that can be detected
    - `<canvas>`, `<video>`, etc.

# HTML 5 - Summary (2)

- The HTML5 specification defines how tags interact with JavaScript
  - Through the Document Object Model (DOM)
- HTML5 doesn't just define a `<video>` tag
  - There is also a corresponding DOM API for video objects in the DOM
  - You can use this API to detect support for different video formats, etc.

# HTML 5 - Resources

- HTML 5 Rocks – Examples, Demos, Tutorials
  - http://www.html5rocks.com/
- HTML 5 Demos
  - http://html5demos.com/
- Internet Explorer 9 Test Drive for HTML 5
  - http://ie.microsoft.com/testdrive/
- Apple Safari HTML 5 Showcases
  - http://www.apple.com/html5/
- Dive into HTML 5
  - http://diveintohtml5.org/