* Attributes are properties of HTML Elements
  + Used to set size, color, border, etc…
  + Put directly in the tags
  + Has value surrounded by " " or ' '
  + The value is always a string
  + There are some attributes that are common for every HTML element
    - Id, class, name, style
  + And some attributes are specific
    - For example the attribute *src* (specifies the URL of the image) of the *img* element
    - Shows the path to the image to be shown
* The html element
  + Used to mark the beginning and ending of a HTML document
  + All the content of the web page is inside this tag
* The head tag contains markup that is not visible to the user (i.e. the person using the browser)
  + But helps the browser to render correctly the HTML document
  + What is in there?
    - Styles, scripts
    - Declare encodings
    - Etc..
  + The title tag - the text in the tab of a browser
* body element contains all the visible to the user markup
  + Headings, text, hyperlinks, images, etc…
  + Textboxes, sliders, buttons…
* Doctype is kind of the validator of the page
  + Tells the browser in which version of HTML the page is written
  + HTML 5 Doctype
* **Text formatting tags**

    <b></b> bold

    <i></i> italicized

    <u></u> underlined

    <sup></sup> Samplesuperscript

    <sub></sub> Samplesubscript

    <strong></strong> strong

    <em></em> emphasized

    <pre></pre> Preformatted text

* Hyperlink Tags

<a href=http://www.telerik.com/ title="Telerik">Link to Telerik Web site</a>

* + inline element
* Image Tags

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

* Text formatting tags

This text is <em>emphasized.</em>

<br />new line<br />

This one is <strong>more emphasized.</strong>

* **Heading Tags (h1 – h6)**
  + <h1>Heading 1</h1>
  + <h2>Sub heading 2</h2>
  + <h3>Sub heading 3</h3>
* Paragraph Tags: Paragraphs are block-level elements
  + <p>This is my first paragraph</p>
  + <p>This is my second paragraph</p>
* Block-level elements
  + HTML (Hypertext Markup Language) elements are usually "block-level" elements or "inline" elements. A block-level element occupies the entire space of its parent element (container), thereby creating a "block."
  + Browsers typically display the block-level element with a newline both before and after the element.
  + Block-level elements may appear only within a <body> element.
* Sections: div and span
  + The HTML <div> element (or HTML Document Division Element)
    - is the generic container for flow content, which does not inherently represent anything.
    - It can be used to group elements for styling purposes (using the class or id attributes), or because they share attribute values, such as lang.
    - It should be used only when no other semantic element (such as <article> or <nav>) is appropriate.
    - Example: <div style="background: skyblue;">This is a div</div>
  + The HTML <span> element
    - generic inline container for phrasing content, which does not inherently represent anything.
    - It can be used to group elements for styling purposes (using the class or id attributes), or because they share attribute values, such as lang.
    - It should be used only when no other semantic element is appropriate.
    - <span> is very much like a <div> element, but <div> is a block-level element whereas a <span> is an inline element.
    - Example: <p><span>Some text</span></p>
* Create an Ordered List using <ol></ol>:

<ol type="1">

<li>Apple</li>

<li>Orange</li>

<li>Grapefruit</li>

</ol>

* + Attribute values for type are 1, A, a, I, or i
* Create an Unordered List using <ul></ul>:

<ul type="disc">

<li>Apple</li>

<li>Orange</li>

<li>Grapefruit</li>

</ul>

* + Attribute values for type are: disc, circle or square
* 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>

* Semantic Structural Tags
  + A sample layout structure of a Web Page
  + The "HTML 4 and Before" Way

<html>

<head> … </head>

<body>

<div id="header"> … </div>

<div id="navigation"> … </div>

<div id="sidebar"> … </div>

<div id="content"> … </div>

<div id="footer"> … </div>

</body>

</html>

* + The HTML 5 Way
    - In HTML 5 there are semantic tags for layout
    - <nav>, <header>, <footer>, <section>

    <html>

    <head>… </head>

    <body>

        <header>… </header>

        <nav>… </nav>

        <aside>… </aside>

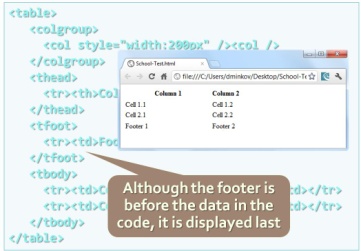
        <section>… </section>

        <footer>… </footer>

    </body>

    </html>

* The *W3C HTML Validator* is a way to validate your HTML
* Tables are comprised of several core tags:
  + <table></table>: begin/end table definition
  + <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
* Table rows split into three semantic sections: header, body and footer
  + <thead> denotes table header and contains <th> elements, instead of <td> elements
  + <tbody> denotes collection of table rows that contain the very data
  + <tfoot> denotes table footer but comes BEFORE the <tbody> tag
  + <colgroup> and <col> define columns (used to set column widths)



* Table "cells" (<td>) can contain nested tables (tables within tables):
* Tables have two attributes related to space
  + cellspacing - Defines the empty space between cells
  + cellpadding - Defines the empty space around the cell content
* Cells have two attributes related to merging
  + colspan - Defines how many columns the cell occupies
  + rowspan - Defines how many rows the cell occupies
* <noframes>
  + HTML element which is used to supporting browsers which are not able to support <frame> elements or configured to do so.
  + **Note:** Because of the fact that all mainstream browsers support frames, usage of this element is not necessary in general cases. It is also entirely obsolete in HTML5 and should be avoided to conform to the standard.
  + **Example:**

<frameset cols="50%,50%">

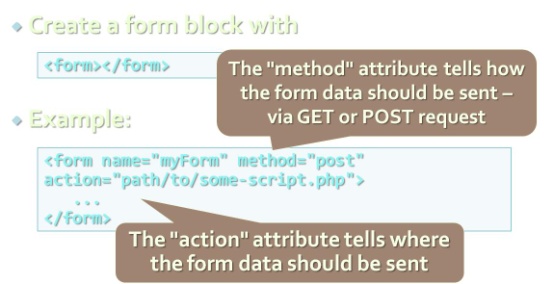
<frame src="https://developer.mozilla.org/en/HTML/Element/frameset" />

<frame src="https://developer.mozilla.org/en/HTML/Element/frame" />

<noframes><p>It seems your browser does not support frames or configured to does not so.</p></noframes>

</frameset>

* The HTML Inline Frame Element (<iframe>)
  + represents a nested browsing context, effectively embedding another HTML page into the current page.
  + In HTML 4.01, a document may contain a head and a body or a head and a frame-set, but not both a body and a frame-set. However, an <iframe> can be used within a normal document body.
  + Each browsing context has its own session history and active document.
* HTML Forms
  + **primary method for gathering data from site visitors**
  + HTML Forms can contain: Text fields, Buttons, Menus, Sliders, etc…



* Single-line text input fields:

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

* Multi-line text input fields (textarea):

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

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

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

* Reset button – brings the form to its initial state

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

* Submit button:

<input type="submit" value="Apply Now" />

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

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

* Ordinary button – no default action, used with JS

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

* 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="Lom" />

<input type="radio" name="city" value="Ruse" />

* Dropdown menus:

    <select name="gender">

        <option value="Value 1"

            selected="selected">Male</option>

        <option value="Value 2">Female</option>

        <option value="Value 3">Other</option>

    </select>

* Multiple-choice menus

<select name="products" multiple="multiple">

      <option value="Value 1" selected="selected">keyboard</option>

      <option value="Value 2">mouse</option>

</select>

* Hidden fields contain invisible data
  + Not shown to the user
  + Used by JavaScript and server-side code ViewState, SessionState in ASP.NET

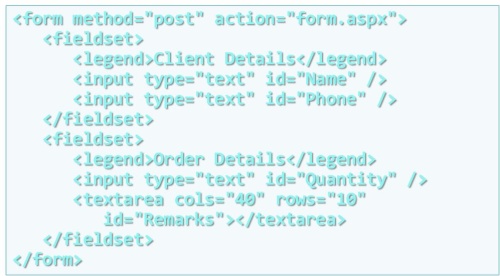
<input type="hidden" name="Account" value="This is a hidden text field" />

* Labels are used to associate an explanatory text to a form field using the field's ID

<label for="fn">First Name</label>

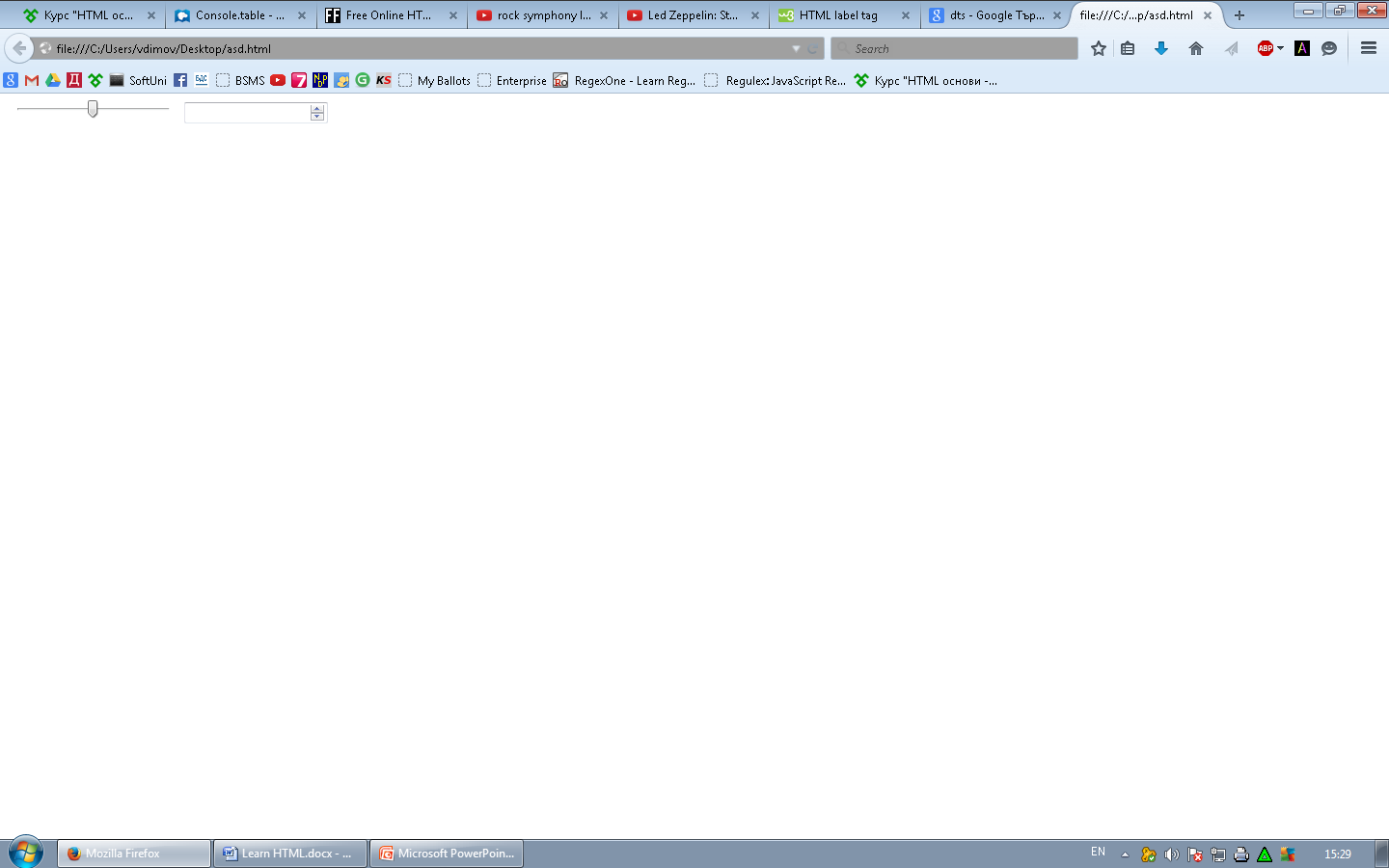
<input type="text" id="fn" />

* + Clicking on a label focuses its associated field
  + Checkboxes are toggled, Radio buttons are checked
  + Labels are Both a usability and accessibility feature
  + Required in to pass accessibility validation
* Fieldsets are used to enclose a group of related form fields:
  + The <legend> is the fieldset's title



* Sliders and Spinboxes
  + Restricts users to enter only numbers
  + Additional attributes min, max and step and value
  + Can become Spinbox or Slider, depending on the input type
  + Have some differences on different browsers
  + Spinboxes do not work on Firefox
    - Shown as regular textboxes

<input type="range" min="0" max="100" />

<input type="number" min="0" max="100" /> 

* Field Attributes from HTML 5
  + Autocomplete
    - The browser stores the previously typed values
    - Brings them back on a later visit
  + Autofocus
    - The field becomes on focus on page load
  + Required
    - The field is required to be filled/selected

<form autocomplete="on">

First name:<input type="text" name="fname" required autofocus><br>

Last name: <input type="text" name="lname" required><br>

E-mail: <input type="email" name="email" autocomplete="off"><br>

<input type="submit">

</form>

* Input Fields with Validation
  + Email – provides a simple validation for email
    - Can be passed a pattern for validation
    - In a mobile device brings the email keyboard

**<input type="email" required="true" pattern="[^ @]\*@[^ @].[^ @]"/>**

* + URL – has validation for url
    - In a mobile device brings the url keyboard

<input type="url" required="true" />

* + Telephone
    - Brings the numeric keyboard

**<input type="tel" required="true" />**

* 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" name="second" tabindex="10" />

<input type="text" name="first" tabindex="5" />

* 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
  + Frames are not supported by all user agents (browsers, search engines, etc.)
  + A <noframes> element is used to provide content for non-compatible agents.

<html>

<head><title>Frames Example</title></head>

<frameset cols="180px,\*,150px">

<frame src="left.html" />

<frame src="middle.html" />

<frame src="right.html" />

</frameset>

</html>

* HTML5 semantic tags
  + <header>
    - Site header or section header or article header
  + <footer>
    - Site footer (sometime can be a section footer)
  + <nav>
    - Site navigation (usually in the header)
  + <section>
    - Site section (e.g. news, comments, links, …)
  + <article>
    - Article in a section (e.g. news item)
  + <aside>
    - Sidebar (usually on the left or on the right)
  + <figure>
    - Figure (a figure, e.g. inside an article)
  + <figcaption>
    - A caption of a figure (inside the <figure> tag)
  + <audio> / <video>
    - Audio / video element (uses the built-in player)
  + <details> + <summary>
    - Accordion-like widget (can be open / closed)
  + <hgroup>
    - Group article header + subheader (<h1> + <h2>)
  + <time>
    - Specifies date / time (for a post / article / news)
* Strong <strong> vs. Bold <b>
  + <b> does not mean anything
    - It just makes the text bolder
  + <strong> marks the text is "stronger" than the other, surrounding text
* Emphasis <em> vs. Italic <i>
  + Emphasis does not always mean, that the code should be italic
  + It could be bolder, italic and underlined
  + The styles for the emphasis text should be set with CSS, not by HTML
  + An example for <em> could be: "Just *do* it already!", or: "We *had* to do something about it". A person or software reading the text would pronounce the words in italics with an emphasis.
* Web 1.0

Web 1.0 refers to the first stage in the World Wide Web, which was entirely made up of Web pages connected by hyperlinks. Although the exact definition of Web 1.0 is a source of debate, it is generally believed to refer to the Web when it was a set of static websites that were not yet providing interactive content. In Web 1.0, applications were also generally proprietary.

* Web 2.0

Web 2.0 is the name used to the describe the second generation of the World Wide Web, where it moved static HTML pages to a more interactive and dynamic Web experience. Web 2.0 is focused on the ability for people to collaborate and share information online via social media, blogging and Web-based communities.

* web 3.0

Upcoming

* XHTML

Езикът е създаден, за да направи [HTML](http://bg.wikipedia.org/wiki/HTML) по-гъвкав език с по-големи възможности. Друга причина за създаването на XHTML е оперативната съвместимост на различните приложения и формати от данни. Докато [HTML4](http://bg.wikipedia.org/w/index.php?title=HTML4&action=edit&redlink=1) е базиран изцяло на [SGML](http://bg.wikipedia.org/wiki/SGML), XHTML е написан изцяло като [XML](http://bg.wikipedia.org/wiki/XML)-базиран език.

**Различия с HTML**

* За разлика от [HTML](http://bg.wikipedia.org/wiki/HTML), XHTML не позволява пропускането на затварящите тагове на някои елементи като paragraph, heading и други.
* В [HTML](http://bg.wikipedia.org/wiki/HTML) целия код е чувствителен към регистъра на малките и главни букви (*case-insensitive*). В [XML](http://bg.wikipedia.org/wiki/XML) само имената на таговете и атрибутите са case-insensitive, но не е позволено тагове да се наименуват или да започват с главни букви.
* HTML позволява скриването на някои от атрибутите и писането само на техните стойности (Пример: <option selected>). В XML-базираните езици този код би изглеждал така: <option selected="selected">..
* За разлика от XML, в HTML някои елементи като <html>, <head>, или <body> могат да бъдат да бъдат пропуснати, а след това автоматично добавени от парсера.
* В старите версии на HTML е позволено стойностите да бъдат задавани без кавички. Пример: <body lang=en>
* Web Page
  + Document or information resource that is suitable for the World Wide Web
  + Can be accessed through a web browser and displayed on a monitor or mobile device
  + This information is usually in HTML or XHTML format, and may provide navigation to other web pages via hypertext links
  + Web pages frequently refer to other resources such as style sheets (CSS), scripts (JavaScript) and images into their final presentation
* **Web Site**
  + Collection of related web pages containing web resources (web pages, images, videos, CSS files, JS files or other digital assets)
  + Common navigation between web pages
  + A website is hosted on at least one web server
  + Accessible via a network (such as the Internet)
  + All publicly accessible websites collectively constitute the World Wide Web
* Web Application
  + Next level web sites
  + High interactivity
  + High accessibility (Cloud)
  + AJAX, Silverlight, Flash, Flex, etc.
  + Applications are usually broken into logical chunks called "tiers", where every tier is assigned a role
  + Desktop-like application in the web browser
  + Web applications on desktop (Windows 8)
* Web Browsers
  + Main responsibilities:
  + Bring information resources to the user (issuing requests to the web server and handling any results generated by the request)
  + Presenting web content (render HTML, CSS, JS)
  + Capable of executing applications within the same context as the document on view (Flash)
* Layout engine
  + A software component that combines content and formatting information for electronic or printed display. Layout engines can be found embedded in web browsers and reporting software, and included in application frameworks that support a graphical user interface.
  + Some applications combine layout engines with a parser that interprets markup to get their content and formatting information, such as web browsers.
* Layout Engines and Web Browsers
  + Trident-based
    - Internet Explorer, Netscape, Maxthon, etc.
  + Gecko-based
    - Firefox, Netscape, SeaMonkey, etc.
  + Blink-based
    - Chrome, Opera
  + WebKit-based
    - Safari, iOS, Maxthon, Chrome (up to v27), etc.
  + EdgeHTML (fork of Trident)
    - Spartan (the new IE)
    - Windows 10 and Windows 10 (Mobile)
* Hardware Servers
  + Physical computer (a hardware system) dedicated to running one or more such services
* ASP.NET
  + an open source[2] server-side Web application framework designed for Web development to produce dynamic Web pages.
  + It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.
* IIS based Web server
  + Internet Information Services (IIS, formerly Internet Information Server) is an extensible web server created by Microsoft for use with Windows NT family.[2] IIS supports HTTP, HTTPS, FTP, FTPS, SMTP and NNTP. It has been an integral part of the Windows NT family since Windows NT 4.0, though it may be absent from some editions (e.g. Windows XP Home edition). IIS is not turned on by default when Windows is installed. The IIS Manager is accessed through the Microsoft Management Console or Administrative Tools in the Control Panel.
  + In what scenarios must WCF be used
    - A secure service to process business transactions.
    - A service that supplies current data to others, such as a traffic report or other monitoring service.
    - A chat service that allows two people to communicate or exchange data in real time.
    - A dashboard application that polls one or more services for data and presents it in a logical presentation.
    - Exposing a workflow implemented using Windows Workflow Foundation as a WCF service.
    - A Silverlight application to poll a service for the latest data feeds.
* The 3-tier architecture consists of the following tiers (layers):
  + Front-end (client layer)
    - Client software – provides the UI of the system
  + Middle tier (business layer)
    - Server software – provides the core system logic
    - Implements the business processes / services
  + Back-end (data layer)
    - Manages the data of the system (database / cloud)

