HTML and CSS

Dr. Öğr. Üyesi Ali Nizam

HTML and CSS

- HTML (Hypertext Markup Language) uses a markup system composed of elements which represent specific content.
- Markup means that with HTML you declare what is presented to a viewer, not how it is presented.
- Visual representations are defined by Cascading Style Sheets (CSS) and realized by browsers.
- Still existing elements that allow for such, like e.g. font, "are entirely obsolete, and must not be used by authors".

HTML

- HTML is sometimes called a programming language but it has no logic, so is a markup language.
- HTML tags provide semantic meaning and machine-readability to the content in the page.

The doctype

- A ***doctype*** (sometimes capitalized as “DOCTYPE”) is an special instruction which, for legacy reasons that have to do with processing modes in browsers, is a required part of any document in the HTML syntax
- <!DOCTYPE html>

Tags

- Start tags consist of the following parts, in exactly the following order:
  - A "<" character.
  - The element’s tag name.
  - Optionally, one or more attributes, each of which must be preceded by one or more space characters.
  - Optionally, one or more space characters.
  - Optionally, a "/" character, which may be present only if the element is a void element.
  - A ">" character.
- End tags consist of the following parts, in exactly the following order:
  - A "<" character.
  - A "/" character
  - The element’s tag name.
  - Optionally, one or more space characters.
  - A ">" character.

https://www.w3.org/

Elements

- An element usually consists of an opening tag (<element_name>), a closing tag (</element_name>), which contain the element's name surrounded by angle brackets, and the content in between:
- <element_name>...content...</element_name>
- There are some HTML elements that don't have a closing tag or any contents. These are called void elements. Void elements include <img>, <meta>, <link> and <input>.
- Element names can be thought of as descriptive keywords for the content they contain, such as video, audio, table, footer.
- A HTML page may consist of potentially hundreds of elements which are then read by a web browser, interpreted and rendered into human readable or audible content on the screen.
- For this document it is important to note the difference between elements and tags:
- Elements: video, audio, table, footer
- Tags: <video>, <audio>, <table>, <footer>, </html>, </body>

Tags

- Void elements only have a start tag; end tags must not be specified for void elements.

Attributes

- Attributes for an element are expressed inside the element’s start tag. Attributes have a name and a value.

- There must never be two or more attributes on the same start tag whose names are a case-insensitive match for each other.

- The following list describes syntax rules for attributes in documents in the HTML syntax. Syntax rules for attributes in documents in the XML syntax. are defined in the XML specification [XML].

Comments

- the comment start delimiter "<!--"
- text
- the comment end delimiter "-->«
- Example

- <!-- main content starts here -->

Page Structure

- <!DOCTYPE html>
- <html>
- <head>
- <title>Page Title</title>
- </head>
- <body>

- <h1>This is a Heading</h1>
- <p>This is a paragraph.</p>

- </body>
- </html>

Meaning

- **<html>**

  Opens the page. No markup should come after the closing tag (<**/html**>). The lang attribute declares the primary language of the page using the ISO language codes (en for English).

  See the Content Language topic for more information.

- **<head>**

  Opens the head section, which does not appear in the main browser window but mainly contains information *about* the HTML document, called *metadata*. It can also contain imports from external stylesheets and scripts. The closing tag is **</head>**.

- **<meta>**

  Gives the browser some metadata about the document. The charset attribute declares the character encoding. Modern HTML documents should always use UTF-8, even though it is not a requirement. In HTML, the <**meta**> tag does not require a closing tag.

  See the Meta topic for more information.

- **<title>**

  The title of the page. Text written between this opening and the closing tag (<**/title**>) will be displayed on the tab of the page or in the title bar of the browser.

- **<body>**

  Opens the part of the document displayed to users, i.e. all the visible or audible content of a page. No content should be added after the closing tag **</body>**.

Headings

- HTML provides not only plain paragraph tags, but six separate header tags to indicate headings of various sizes and thicknesses. Enumerated as heading 1 through heading 6, heading 1 has the largest and thickest text while heading 6 is the smallest and thinnest, down to the paragraph level.
- <**h1**>Heading 1</**h1**>
- <**h2**>Heading 2</**h2**>
- <**h3**>Heading 3</**h3**>
- <**h4**>Heading 4</**h4**>
- <**h5**>Heading 5</**h5**>
- <**h6**>Heading 6</**h6**>

Paragraphs

- **Column Column**
  - <**p**> Defines a paragraph
  - <**br**> Inserts a single line break
  - <**pre**> Defines pre-formatted text
- Example
  - <**p**>This is a paragraph.</**p**>
  - <**p**>This is another paragraph.</**p**>

Text Formatting

- The <mark> element is new in HTML5 and is used to mark or highlight text in a document "due to its relevance in another context
- Example
  - <**p**>Here is some content from an article that contains the <**mark**>searched query</**mark**> that we are looking for. Highlighting the text will make it easier for the user to find what they are looking for.</**p**>
- **Bold Text**
  - To bold text, use the <**strong**> or <**b**> tags:
- **Italic Text**
- To italicize text, use the <**em**> or <**i**> tags:

Anchors and Hyperlinks

- Specifies the destination address. It can be an absolute or relative URL, or the name of an anchor. An absolute URL is the complete URL of a website like http://example.com/. A relative URL points to another directory and/or document inside the same website, e.g. /about-us/ points to the directory "about-us" inside the root directory (/). When pointing to another directory without explicitly specifying the document, web servers typically return the document "index.html" inside that directory.
- **hreflang** Specifies the language of the resource linked by the href attribute
- **rel** Specifies the relationship between the current document and the linked document
- **target** Specifies where to open the link, e.g. in a new tab or window. Possible values are _blank, _self, _parent, _top, and framename (deprecated). Forcing such behaviour is not recommended since it violates the control of the user over a website.
- **Title:** Specifies extra information about a link. The information is most often shown as a tooltip text when the cursor moves over the link. This attribute is not restricted to links, it can be used on almost all HTML tags.
- **download**: Specifies that the target will be downloaded when a user clicks on the hyperlink. The value of the attribute will be the name of the downloaded file. There are no restrictions on allowed values, and the browser will automatically detect the correct file extension and add it to the file (.img, .pdf, etc.). If the value is omitted, the original filename is used.

Link to another site

- This is the basic use of the <a> (anchor element) element:
- <a href="http://example.com/">Link to example.com</a>
- It creates a hyperlink, to the URL http://example.com/ as specified by the href (hypertext reference) attribute, with the anchor text "Link to example.com". It would look something like the following:
  - Link to example.com

Link to an anchor

- Anchors can be used to jump to specific tags on an HTML page. The <a> tag can point to any element that has an id attribute.
- <**h2** id="Topic1">First topic**</h2**>
- <**p**>Content about the first topic**</p**>
- <**h2** id="Topic2">Second topic**</h2**>
- <**p**>Content about the second topic**</p**>
- Now you can use the anchor in your table of contents:
- <**h1**>Table of Contents**</h1**>
- <**a** href='#Topic1'>Click to jump to the First Topic**</a**>
- <**a** href='#Topic2'>Click to jump to the Second Topic**</a**>

Link to a page on the same site

- You can use a relative path to link to pages on the same website.
- <a href="/example">Text Here</a>
- The above example would go to the file example at the root directory (/) of the server.

Link that dials a number

- If the value of the href-attribute begins with tel:, your device will dial the number when you click it. This works on mobile devices or on computers/tablets running software – like Skype or FaceTime – that can make phone calls.
- <a href="tel:11234567890">Call us</a>

Open link in new tab/window

- <a href="example.com" target="_blank">Text Here</a>

Link that runs JavaScript

- Simply use the javascript: protocol to run the text as JavaScript instead of opening it as a normal link:
    - <a href="javascript:myFunction();">Run Code</a>
- You can also achieve the same thing using the onclick attribute:
    - <a href="#" onclick="myFunction(); return false;">Run Code</a>

Link that runs email client

- If the value of the href-attribute begins with mailto: it will try to open an email client on click:
  - <a href="mailto:example@example.com">Send email</a>

Lists

- HTML offers three ways for specifying lists: ordered lists, unordered lists, and description lists.

Ordered List

- An ordered list can be created with the <ol> tag and each list item can be created with the <li> tag as in the
- example below:
  - <ol>
  - <li>Item</li>
  - <li>Another Item</li>
  - <li>Yet Another Item</li>
  - </ol>
- This will produce a numbered list (which is the default style):

  1. Item
  2. Another Item
  3. Yet Another Item

Ordered list - Manually changing the numbers

- The first way is to set a starting number, using the start attribute. The list will start at this defined number, and continue
- incrementing by one as usual.
  - <ol start="3">
  - <li>Item</li>
  - <li>Some Other Item</li>
  - <li

Unordered List

- An unordered list can be created with the <ul> tag and each list item can be created with the <li> tag as shown by the example below:
  - <ul>
  - <li>Item</li>
  - <li>Another Item</li>
  - <li>Yet Another Item</li>
  - </ul>
- This will produce a bulleted list (which is the default style):
  - Item
  - Another Item
  - Yet Another Item

Nested lists

- You can nest lists to represent sub-items of a list item.
  - <ul>
  - <li>item 1</li>
  - <li>item 2
  - <ul>
  - <li>sub-item 2.1</li>
  - <li>sub-item 2.2</li>
  - </ul>
  - </li>
  - <li>item 3</li>
  - </ul>
- Output
  - item 1
  - item 2
    - sub-item 2.1
    - sub-item 2.2
  - item 3

Description List

- A description list (or definition list, as it was called before HTML5) can be created with the dl element. It consists of name-value groups, where the name is given in the dt element, and the value is given in the dd element.
- <dl>
- <dt>name 1</dt>
- <dd>value for 1</dd>
- <dt>name 2</dt>
- <dd>value for 2</dd>
- </dl>

GoalKicker.com – HTML5 Notes for Professionals 18

Tables

- The HTML <table> element allows web authors to display tabular data (such as text, images, links, other tables, etc.) in a two dimensional table with rows and columns of cells.

```
<table>
<tr>
<th>Heading 1/Column 1</th>
<th>Heading 2/Column 2</th>
</tr>
<tr>
<td>Row 1 Data Column 1</td>
<td>Row 1 Data Column 2</td>
</tr>
</table>
```

Classes and IDs

- **class** Indicates the Class of the element (non-unique)
- **id** Indicates the ID of the element (unique in the same context)
- Classes and IDs make referencing HTML elements from scripts and stylesheets easier.
- The class attribute can be used on one or more tags and is used by CSS for styling.
- IDs however are intended to refer to a single element, meaning the same ID should never be used twice. IDs are generally used with JavaScript and internal document links, and are discouraged in CSS.

Giving an element a class

- Classes are identifiers for the elements that they are assigned to. Use the class attribute to assign a class to an element.
- <div class="example-class"></div>
- To assign multiple classes to an element, separate the class names with spaces.
- <div class="class1 class2"></div>
- Using classes in CSS
- Classes can be used for styling certain elements without changing all elements of that kind. For example, these two
- span elements can have completely different stylings:
- <span></span>
- <span class="special"></span>

- If our CSS is as below, then the color green will be applied to the text within both elements:
- .highlight { color: green; }

Data Attributes

- Data Attributes were introduced in HTML5 which is supported by all modern browsers, but older browsers before HTML5 don't recognize the data attributes.
- <**img** src="sample.jpg" value="test" />

Data Attribute Use

- HTML5 data-* attributes provide a convenient way to store data in HTML elements. The stored data can be read or
- modified using JavaScript
- <div data-submitted="yes" class="user_profile">
- … some content …
- </div>
- Data attribute structure is data-*, i.e. the name of the data attribute comes after the data- part. Using this name, the attribute can be accessed.
- Data in string format (including json) can be stored using data-* attribute.

Giving an element an ID

- The ID attribute of an element is an identifier which must be unique in the whole document. Its purpose is to uniquely identify the element when linking (using an anchor), scripting, or styling (with CSS).
- <div id="example-id"></div>
- You should not have two elements with the same ID in the same document, even if the attributes are attached to two different kinds of elements. For example, the following code is incorrect:
- <div id="example-id"></div>
- <span id="example-id"></span>
- Browsers will do their best to render this code, but unexpected behavior may occur when styling with CSS or adding functionality with JavaScript.
- To reference elements by their ID in CSS, prefix the ID with #.
- #example-id { color: green; }

CSS

- **External Stylesheet Use**
- Use the link attribute in the document's head:
  - <**head**>
  - <**link** rel="stylesheet" type="text/css" href="stylesheet.css">
  - </**head**>

- <**head**>
- <**link** rel="stylesheet"
- href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-
- BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
- </**head**>

CSS

- CSS is a language that describes the style of an HTML document.
- CSS describes how HTML elements should be displayed.

https://www.w3schools.com/css/

CSS Selectors

- Element_Type {property:value} : for elements
- Element_Type, Element_Type {property:value} : for group of elements
- .class {property:value} : for classes
- Element.class{property:value} : for elements’ classes
- #id {property:value} : for elements’ id, : id selector
  - #black {
  -     color: #000000;
  - }
- Element_Type [attr=value] {property:value} styles to HTML elements with particular attributes
- * {property:value}: universal selector, for every HTML element

CSS

- **Internal Stylesheet**
- You can also include CSS elements internally by using the **<style>** tag:
- **<head>**
- **<style** type="text/css">
- body {
- background-color: gray;
- }
- **</style>**
- **</head>**

Inline Style

- You can style a specific element by using the style attribute:
- <span style="color: red">This text will appear in red.</span>

Creating an image

- To add an image to a page, use the image tag.
- Image tags (img) do not have closing tags. The two main attributes you give to the img tag are src, the image source and alt, which is alternative text describing the image.
- <**img** src="images/hello.png" alt="Hello World">

Image alt text

- Alt-text is used by screen readers for visually impaired users and by search engines. It's therefore important to
- write good alt-text for your images
  - <img src="anonymous.png" alt=""/> An anonymous user wrote:
  - <blockquote>Lorem ipsum dolor sed.</blockquote>
  - <a href="https://google.com/"><img src="edit.png" alt="Edit"/></a> /
  - <a href="https://google.com/"><img src="delete.png" alt="Delete"/></a>

Input Control Elements

- **Text**
- The most basic input type and the default input if no type is specified.

- <**input** type="text" size="50">

Checkbox and Radio Buttons

- **Overview**
- Checkboxes and radio buttons are written with the HTML tag **<input>**, and their behavior is defined in the HTML
- specification.
- The simplest checkbox or radio button is an **<input>** element with a type attribute of checkbox or radio,
- respectively:
- **<input** type="checkbox">
- **<input** type="radio">

- **<input** type="radio" name="color" id="red" value="#F00">
- **<input** type="radio" name="color" id="green" value="#0F0">
- **<input** type="radio" name="color" id="blue" value="#00F">

Checkbox and Radio Buttons

- checkboxes and radio buttons have a number of attributes to control their behavior:
- **value**
- Like any other input element, the value attribute specifies the string value to associate with the button in the event of form submission
- **checked**
  - <input checked>
  - <input checked="">
  - <input checked="checked">
  - <input checked="ChEcKeD">

Button Groups

- Since each radio button affects the others in the group, it is common to provide a label or context for the entire group of radio buttons.
- To provide a label for the entire group, the radio buttons should be included in a <**fieldset**> element with a
- <**legend**> element within it.
- Example:
- <**fieldset**>
- <**legend**>Theme color:</**legend**>
- <**p**>
- <**input** type="radio" name="color" id="red" value="#F00">
- <**label** for="red">Red</**label**>
- </**p**>
- <**p**>
- <**input** type="radio" name="color" id="green" value="#0F0">
- <**label** for="green">Green</**label**>
- </**p**>

Input Validation

- HTML input validation is done automatically by the browser based on special attributes on the input element.
- It could partially or completely replace JavaScript input validation. This kind of validation can be circumvented by the user via specially crafted HTTP requests, so it does not replace server-side input validation. The validation only
- occurs when attempting to submit the form, so all restricted inputs must be inside a form in order for validation to occur (unless you're using JavaScript).
- Keep in mind that inputs which are disabled or read-only will not trigger validation.

Input Validation (Version ≥ HTML5)

- **Required**
- Use the required attribute to indicate that a field must be completed in order to pass validation.
- <**input** required>
- **Minimum / Maximum Length**
- Use the minlength and maxlength attributes to indicate length requirements. Most browsers will prevent the user from typing more than *max* characters into the box, preventing them from making their entry invalid even before they attempt submission.
- <**input** minlength="3">
- <**input** maxlength="15">
- <**input** minlength="3" maxlength="15">
- **Specifying a range** Use min and max attributes to restrict the range of numbers a user can input into an input of type number or range
- Marks: <**input** type="number" size="6" name="marks" min="0" max="100" />
- Subject Feedback: <**input** type="range" size="2" name="feedback" min="1" max="5" />

File

- <**input** type="file" name="fileSubmission">
- File inputs allow users to select a file from their local filesystem for use with the current page. If used in conjunction with a form element, they can be used to allow users to upload files to a server
- The following example allows users to use the file input to select a file from their filesystem and upload that file to
- a script on the server named upload_file.php.

<**form** action="upload_file.php" method="post" enctype="multipart/form-data">

Select file to upload:

<**input** type="file" name="fileSubmission" id="fileSubmission">

<**input** type="submit" value="Upload your file" name="submit">

</**form**>

Button

- <**input** type="button" value="Button Text">
- Buttons can be used for triggering actions to occur on the page, without submitting the form. You can also use the
- <**button**> element if you require a button that can be more easily styled or contain other elements:
- <**button** type="button">Button Text</**button**>
- Buttons are typically used with an "onclick" event:
- <**input** type="button" onclick="alert('hello world!')" value="Click Me">
- or
- <**button** type="button" onclick="alert('hello world!')">Click Me</**button**>

Button types

- **submit** : The button submits the form data to the server. This is the default if the attribute is not specified, or if the attribute is dynamically changed to an empty or invalid value.
- **reset** : The button resets all the controls to their initial values.
- **button**: The button has no default behavior. It can have client-side scripts associated with the element's events, which are triggered when the events occur.
- **menu** : The button opens a popup menu defined via its designated element.

Forms

- In order to group input elements and submit data, HTML uses a form element to encapsulate input and submission elements.
- These forms handle sending the data in the specified method to a page handled by a server or handler.
- This topic explains and demonstrates the usage of HTML forms in collecting and submitting input data.

Submitting

- **The Action Attribute**
- The action attribute defines the action to be performed when the form is submitted, which usually leads to a script that collects the information submitted and works with it.
- if you leave it blank, it will send it to the same file

<**form** action="action.php">

Form

- The Method Attribute
- The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data:
- <form action="/action_page.php" **method="get"**>

Form

- When to Use GET?
- The default method when submitting form data is GET.
- However, when GET is used, the submitted form data will be **visible in the page address field**:
- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

Forms

- When to Use POST?
- Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.
- **Notes on POST:**
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

Article

- Definition and Usage
- The <article> tag specifies independent, self-contained content.
- An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.

-
  ```
  article {
    display: block;
  }
  ```

HTML 5 Semantik elemanlar

Tag	Description
<article>	Defines an article in a document
<aside>	Defines content aside from the page content
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content
<footer>	Defines a footer for a document or section
<header>	Defines a header for a document or section
<main>	Defines the main content of a document
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<progress>	Represents the progress of a task
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

Div element

- The <div> tag defines a division or a section in an HTML document.
- The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

```
<div style="background-color:lightblue">
  <h3>This is a heading</h3>
  <p>This is a paragraph.</p>
</div>
//horizontal blocking
.item img {
  display: block;
}
```

Questions