Web Application Development
(LTAT.05.004)

HTML
HYPERTEXT MARKUP LANGUAGE

MOHAMAD GHARIB
UNIVERSITY OF TARTU
Three-tier arch. in Wep App - Recap

The Front-end is where end user interacts with the application. The Back-end is where information is processed. The Database is where the information is stored and managed.
Representational state transfer (REST) is a software architectural style designed for distributed hypermedia, which defines a set of constraints to be used for creating Web services.
HTML
Hypertext Markup Language
Before we start ...

What do you need to know before starting?

Absolutely nothing.

What we will cover:

• History of HTML
• What HTML is.
• HTML Elements, Tags, attributes, etc.
Before we start ...

What do you need to know before starting?
Absolutely nothing.

What we will cover:
• History of HTML
• What HTML is.
• HTML Elements, Tags, attributes, etc.

How this lecture contributes to the course

HTML
The structure of the webpage

CSS
The styling of the webpage

JavaScript
Makes a webpage interactive
What to expect ...

What do you need to know before starting?
Absolutely nothing.

What we will cover:
• History of HTML
• What HTML is.
• HTML Elements, Tags, attributes, etc.

What you are expected to learn:
• Core aspects of HTML;
• How you can use them to create webpage(s).
Before the World Wide Web (WWW) the Internet only provided screens full of text that is, usually, in one font and font size.

In 1989-1990, Tim Berners-Lee invents the WWW with HTML as its publishing language.

CERN launched the web in 1991 (HTML + HTTP).

HTML was strongly based on SGML (Standard Generalized Mark-up Language), an internationally agreed upon method for marking up text into structural units such as paragraphs, headings, list items, etc. Additionally, SGML could be implemented on any machine.

There are several different version of HTML – HTML 1.0 (1993), 2.0 (1995), 3.2 (1997), 4.01 (1999), and 5 (2008), which is the current version.

Why did HTML have to change?
HTML short history

Before the World Wide Web (WWW) the Internet only provided screens full of text that is, usually, in one font and font size.

In 1989-1990, Tim Berners-Lee invents the WWW with HTML as its publishing language.

CERN launched the web in 1991 (HTML + HTTP).

HTML was strongly based on SGML (Standard Generalized Mark-up Language), an internationally agreed upon method for marking up text into structural units such as paragraphs, headings, list items, etc. Additionally, SGML could be implemented on any machine.

There are several different version of HTML – HTML 1.0 (1993), 2.0 (1995), 3.2 (1997), 4.01 (1999), and 5 (2008), which is the current version.

*Why did HTML have to change? Because the web has changed*
HTML uses tags (<X>..</X>) to represent elements, which can be paired or Unpaired. Elements can contain attributes <P class = “P1”> .. </P>.
Elements can be nested <div> <span> Nested tags </span> </div>.

the syntax of a language is the set of rules that defines the combinations of symbols that are considered to be correctly structured statements or expressions in that language.
Every HTML document starts with a DOCTYPE, followed by an HTML section that contains a head and body.

The <!DOCTYPE>³ declaration helps browsers to display web pages correctly.

³A list of DOCTYPEs is available at: https://www.w3.org/QA/2002/04/valid-dtd-list.html
Structure of an HTML document

The **head** section/element contains the **title** that appears in a Browser's title bar.

The **head** may also include **metadata** that specify the document title, character set, styles, scripts, and other meta information.

**Metadata** is not **displayed**.
The **body** contains the document itself, i.e., what the user sees. Any text within the **body** must be contained within some tags. **Contents inside body tag are displayed.**
Structure of an HTML document

How it looks in the browser
Tags can provide:

- **Meaning of text**: (e.g., `<h1>` means top-level heading, `<p>` means paragraph).
- **Formatting of text** (`<i>` for italic).
- **Additional information** to be displayed (e.g., `<img>` for including an image).
Common tags - Headings

<h1> Heading 1 </h1>
<h2> Heading 2 </h2>
<h3> Heading 3 </h3>
<h4> Heading 4 </h4>
<h5> Heading 5 </h5>
<h6> Heading 6 </h6>

Headings are marked with <h1>, <h2>, ..., <h6> tags, where <h6> defines the most important heading.

Headings are used by search engines to index the structure and content of webpages.
Common tags – Paragraphs/divisions

The `<p>` tag encloses a paragraph of **text**.

The `<div>` tag defines a division or a section in an HTML document.

```html
<h2> Course Description: </h2>
<p> This course will introduce students to key concepts .... </p>
```

```html
<h2> Course Description: </h2>
<div> This course will introduce students to key concepts .... </div>
```

**Course Description:**
This course will introduce students to key concepts ....
Common tags – Comments

Comment <!-- --> tags are used to insert comments in the HTML source code. Comments are not displayed in the browsers, and can be used to explain your code.

Course Description:
This course ....
Common tags – Formatting

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;b&gt;</code></td>
<td>This is a bold text</td>
</tr>
<tr>
<td><code>&lt;strong&gt;</code></td>
<td>This is a strong text</td>
</tr>
<tr>
<td><code>&lt;i&gt;</code></td>
<td>This is an italic text</td>
</tr>
<tr>
<td><code>&lt;em&gt;</code></td>
<td>This is an emphasized text</td>
</tr>
<tr>
<td><code>&lt;sub&gt;</code></td>
<td>This is a subscript</td>
</tr>
<tr>
<td><code>&lt;sup&gt;</code></td>
<td>This is a superscript</td>
</tr>
</tbody>
</table>

Formatting elements are designed to change the display/appearance of text.
Common tags – Formatting

This is a `<small>` Small `</small>` formatting. This is a Small formatting.
This is a `<mark>` Marked `</mark>` formatting. This is a Marked formatting.
This is a `<del>` blue `</del>` red color. This is a blue red color.
This `<ins>` text `</ins>` is inserted. This text is inserted.

**Formatting** elements are designed to change the **display/appearance** of text.
Common tags - Lists

An unordered list (<ul> tag), where each list item starts with the <li> tag.

An ordered list (<ol> tag), where each list item starts with the <li> tag.

A definition list (<dl> tag). The element encloses a list of groups of terms (specified using the <dt> element) and descriptions (by <dd> element).
A **table** is defined with the `<table>` tag.

Each **table row** is defined with the `<tr>` tag.

A **table header** is defined with the `<th>` tag. A table **header** is bold and centred by default.

A **table data** (cell) is defined with the `<td>` tag.

More details can be found at the W3C
https://www.w3.org/WAI/tutorials/tables/

```html
<table>
  <tr>
    <th>First name</th>
    <th>Last name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Sarah</td>
    <td>Jones</td>
    <td>21</td>
  </tr>
</table>
```
The `<a>` tag defines a hyperlink, which is used to link from one page to another. The `href` attribute specifies the destination address of the link. The **content** of the `<a>` tag (link text) is the only **visible part**. To open the link in a **new page**, you can set the target att. to blank (`target="_blank"`).
The `<link>` tag defines the relationship between the current document and an external resource, it is used within the `<head>` section and it has:

**rel (required):** specifies the relationship between the current and linked documents.

**href:** specifies the location of the linked document.
The `<img>` tag is used to **embed an image** into a page, and it has two attributes:

- **src**: specifies the path/location to the image; and
- **alt**: specifies an **alternate text** to display if the image cannot be displayed.

The `<img>` tag contains **attributes** only, and does not have a **closing tag**.
Other tags – Audio

The `<audio>` tag is used to embed an audio into a webpage, and it has:

- controls attribute that adds audio controls, like play, pause, and volume.
- `<source>` allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

Before HTML5, audio files could only be played in a browser with a plug-in (like flash).
Other tags – Video

```html
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

The `<video>` tag is used to **embed a video** into a webpage, and it has:

- `controls` attribute that adds **video controls**, like play, pause, and volume.

`<source>` allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
The `<iframe>` tag specifies an inline frame that is used to embed another document within the current HTML document. A very common example of the use of `iframe` is embedding Google maps within a webpage.
User input and computer code

The `<kbd>` tag is used to define **keyboard input**.
The `<samp>` tag is used to define **sample output** from a computer program.
The `<code>` tag is used to define a piece of computer **code**.

```
<p>press `<kbd>Ctrl + S</kbd>` to save</p>

<p>press `<samp>F1 to continue</samp>`</p>

```code```
x = 5;
y = 6;
z = x + y;
```

press Ctrl + S to save

press F1 to continue

x = 5; y = 6; z = x + y;
User input and computer code

The `<kbd>` tag is used to define keyboard input.
The `<samp>` tag is used to define sample output from a computer program.
The `<code>` tag is used to define a piece of computer code.
The `<pre>` tag defines preformatted text.
Forms

The `<form>` element is used to create an HTML form for user input. It is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

A description of the HTML form elements can be found at: [https://www.w3schools.com/html/html_form_elements.asp](https://www.w3schools.com/html/html_form_elements.asp)
Forms - Attributes

```html
<form action="/page.php" method="get">
  <label for="fname">First name: </label><br>
  <input type="text" name="fname"><br>
  <label for="lname">Last name: </label><br>
  <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
</form>
```

**action** defines the action to be performed when the form is submitted.

**method** specifies the HTTP method to be used when submitting the form data.

The **method** attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data.
Forms – Input element

An `<input>` element can be displayed in several ways, depending on the `type` attribute.

A description of the `input types` can be found at: https://www.w3schools.com/html/html_form_input_types.asp
Forms – Input element - Attributes

```html
<input type="email" name="email" required
placeholder="Enter a valid email"
value="o@ut.ee">
```

**name** specifies a name for an HTML element.

**required** specifies that a field must be filled before submitting the form.

**placeholder** specifies a text that describes the expected value of an input field.

**value** specifies an initial value for an input field:
**Forms – Input element - Attributes**

```html
<input type="email" name="email" required
    placeholder="Enter a valid email" value="o@ut.ee"
    minlength="8" maxlength="20" autofocus>
```

`minlength/maxlength` specifies the minimum/maximum number of characters the user can enter into an `<input>` or `<textarea>`.

`autofocus` an element should be focused on page load.

`readonly` when used, a read-only input field cannot be modified.

`disabled` when used, a disabled input element is unusable and un-clickable.

Please lengthen this text to 8 characters or more (you are currently using 5 characters).
Forms – other input types

```html
<input type="color" name="color">
<input type="range" name="points" min="0" max="10">
<input type="number" name="number" min="0" max="10" step="2">
```

**color** provide a user interface element that lets a user specify a color.

**range** allows specifying a numeric value which must be no less than a given value, and no more than another given value. It is represented using a slider.

**number** allows user enter a number, and **step** specifies the number intervals.
There are two types of display values for HTML elements, **block-level** and **inline**:  

**A block-level element** always starts on a new line and takes up the full width available.  

**An inline element** does not start on a new line and only takes necessary width.
### Block-level vs Inline Elements

<table>
<thead>
<tr>
<th>Block tags</th>
<th>Inline tags</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;address&gt;</code> <code>&lt;article&gt;</code> <code>&lt;aside&gt;</code> <code>&lt;blockquote&gt;</code> <code>&lt;canvas&gt;</code> <code>&lt;dd&gt;</code> <code>&lt;div&gt;</code> <code>&lt;dl&gt;</code> <code>&lt;dt&gt;</code> <code>&lt;fieldset&gt;</code> <code>&lt;figcaption&gt;</code> <code>&lt;figure&gt;</code> <code>&lt;footer&gt;</code> <code>&lt;form&gt;</code> <code>&lt;h1&gt;</code> <code>&lt;hr&gt;</code> <code>&lt;li&gt;</code> <code>&lt;main&gt;</code> <code>&lt;nav&gt;</code> <code>&lt;noscript&gt;</code> <code>&lt;ol&gt;</code> <code>&lt;p&gt;</code> <code>&lt;pre&gt;</code> <code>&lt;section&gt;</code> <code>&lt;table&gt;</code> <code>&lt;tfoot&gt;</code> <code>&lt;ul&gt;</code> <code>&lt;video&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;a&gt;</code> <code>&lt;abbr&gt;</code> <code>&lt;acronym&gt;</code> <code>&lt;b&gt;</code> <code>&lt;bdo&gt;</code> <code>&lt;big&gt;</code> <code>&lt;br&gt;</code> <code>&lt;button&gt;</code> <code>&lt;cite&gt;</code> <code>&lt;code&gt;</code> <code>&lt;dfn&gt;</code> <code>&lt;em&gt;</code> <code>&lt;i&gt;</code> <code>&lt;img&gt;</code> <code>&lt;input&gt;</code> <code>&lt;kbd&gt;</code> <code>&lt;label&gt;</code> <code>&lt;map&gt;</code> <code>&lt;object&gt;</code> <code>&lt;output&gt;</code> <code>&lt;q&gt;</code> <code>&lt;samp&gt;</code> <code>&lt;script&gt;</code> <code>&lt;select&gt;</code> <code>&lt;small&gt;</code> <code>&lt;span&gt;</code> <code>&lt;strong&gt;</code> <code>&lt;sub&gt;</code> <code>&lt;sup&gt;</code> <code>&lt;textarea&gt;</code> <code>&lt;time&gt;</code> <code>&lt;tt&gt;</code> <code>&lt;var&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>

There are two types of display values for HTML elements, **block-level** and **inline**:

**A block-level element** always starts on a new line and takes up the full width available.

**An inline element** does not start on a new line and only takes necessary width.
HTML5 was developed with the following aims:

Encouraging semantic (meaningful) markups: means a markup should has meaning, rather than just simply looking a certain way (e.g., `<h1>-<h6>` tags).
HTML5 was developed with the following aims:

**Encouraging semantic (meaningful) markups:** means a markup should has meaning, rather than just simply looking a certain way (e.g., `<h1-6>` tags). This is clear with the new added tags (`<header>`, `<nav>`, `<main>`, `<article>`, `<aside>`, `<section>`, `<footer>`).
HTML5

HTML5 was developed with the following aims:

**Separating design from content:** non-meaningful markups that only tell the browser how to display things should be deprecated. That is why most of these features were completely deprecated from HTML5, or are still officially supported but are not recommended practices.

**Supporting rich media:** HTML was originally created for (hyper-)text documents, with perhaps a few images, no audio and video. HTML5 provides more rich support for *media* with elements like `<video>` and `<audio>`, and `<canvas>`.
HTML5

HTML5 was developed with the following aims:

Promoting responsive web design that aims at using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones).

To create a responsive website, add the following `<meta>` tag to all your web pages:

```html
<meta name="viewport" content="width=device-width, initial-scale=1.0">`
How to use HTML5

To properly use HTML5, follow these simple rules:

Avoid using deprecated tags/attributes⁴: familiarize your self with which HTML features are no longer supported in HTML5, and avoid using them.

Learn to appropriately use the new tags/attributes: knowing the new HTML5 new tags/attributes (e.g., `<header>`, `<nav>`, `<footer>`) is not enough, you should know how to use them.

Use the `<!DOCTYPE html>` Declaration: many practitioners and some IDE do not include the `<!DOCTYPE html>` declaration, which is considered an error in HTML5.

Validate Your Pages: The W3C provides an Markup Validation Service⁵, which allows checking a page against the HTML5 specification.

---

⁴ Check the list of deprecated tags/attributes, and try to understand why they have been deprecated.
⁵ https://validator.w3.org/
HTML Advantages

HTML Disadvantages
HTML Advantages

• Easy to learn and code/use.
• Loose syntax.
• Supported on every browser.
• Widely used; established on almost every website, if not all websites.
• You need not buy/use any specialized software.
• Very similar to XML syntax, which is increasingly used for data storage.

HTML Disadvantages

• It is a static language, i.e., it cannot produce dynamic content.
• The structuring of HTML documents is hard to grasp.
• You have to keep up with deprecated/new tags.
• Security features offered by HTML are limited.
HTML

Try it yourself …
Chrome Developer Tools (HTML)

This is a heading (h1)

This is Paragraph 1

Nested tags

You can open the Developer Tools by right clicking and inspect.

In the Element tab, right click on any element you want, you can edit the text, the whole element, add or modify an attribute, duplicate or even delete the element.

Note: changes are made in your browser, you can refresh the page to return it to its original status.
Always write HTML code with accessibility in mind!

Make your HTML code as **semantic** as possible.

Ex. of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

Ex. of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.
Check the **Accessibility** of a webpage

**Right clicking, inspect**, then, select the **Lighthouse tab**.

Select type of device (desktop/laptop), and select only **Accessibility** from the categories. Then, run **Analyze page load**. What percentage of **accessibility** did you receive? Try to read the report to better understand the result.
Extra reading/exercises

• W3Schools is a free educational website for learning to code online. With their "Try it Yourself" editor, you can edit the HTML code and view the result immediately: https://www.w3schools.com/html/

• HTML resources and tutorials https://developer.mozilla.org/en-US/docs/Web/HTML
Thank You
for your attention

Mohamad Gharib
mohamad.gharib@ut.ee