HTML: HyperText Markup Language

Databases and Web Applications Laboratory (LBAW)
Bachelor in Informatics Engineering and Computation (L.EIC)

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Outline

- A Brief History of HTML
- HTML Fundamentals
- Microdata
- Web Components

A Brief History of HTML

Origins of HTML

- Created by Tim Berners-Lee and Robert Cailliau at CERN in the late 1980s.
- Main goal was to facilitate document sharing between researchers.
- CERN released it as royalty free in 1993.
- First official version published by IETF in 1993.
- World Wide Web Consortium (W3C) was created to define common standards for browsers and developers to adhere to.

HTML Proposal

- **Information Management: A Proposal**
  - https://www.w3.org/History/1989/proposal.html
  - "*This proposal concerns the management of general information about experiments at CERN.*"
  - "*It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.*"
  - Some practical requirements: remote access, heterogeneity, non-centralization, text-based, "live links".
- Problems being addressed:
  - Information loss - "Often, the information has been recorded, it just cannot be found."
  - Constantly changing information. Keeping a "book-like" organization of all information at CERN is impractical. Changes are distributed.
  - Tree-like organizations and keyword-based organization are also not feasible. Too strict and inflexible.

HTML Proposal

- During its first years (1990-1995), HTML revisions and extensions where first hosted at CERN and then IETF.
- Development was moved to the W3C after its creation in 1994.
- HTML development stopped in 1998 with the publication of HTML 4.
- W3C decided to migrate to an XML-based equivalent, named XHTML.
- XHTML was not widely adopted by web authors.
- HTML development continued outside W3C, with the WHATWG, whose work is now the basis for HTML5.
  - WHATWG - Web Hypertext Application Technology Working Group

The Early Days (1989 - 1993)

- From proposal (1989) to Mosaic release (1993).
- Initial user base consisted primarily of academic and research institutions
- Limited number of browsers, predominantly text-based
- HTML documents were simple and usually written by hand.

Growth Years (1994 - 2002)

- ➔ Wide adoption of the web to the dot.com bubble (1995-2000).
- ➔ Companies dispute the web browser market (aka “browser wars”).
- ➔ Browser development focused on new features, less on standards support.
- ➔ Wide differences between rendering engines.
  Many web pages “designed for browser version x.x”.
- ➔ Extensive use of tables and sliced graphics to achieve “pixel perfect” layouts - “print-like design”. Resulted in ugly and complex HTML code.

Modern Era (2003 -)

- ➔ Wide adoption of modern web browsers.
- ➔ Separation of content and structure from layout and presentation.
- ➔ HTML controls content and structure.
- ➔ CSS controls layout and presentation.
- ➔ Return to semantic markup principles
- ➔ CSS (2003), AJAX (2005), mobile (2007).
- ➔ A platform for (web) applications.

XHTML

➔ In 1998, the W3C decided to abandon HTML development and focus on a XML-based equivalent, named XHTML.

➔ XHTML 1.0 was completed in 2000.

➔ W3C then moved to XHTML 2.0, introducing several new features and less backward compatibility.

➔ Real world adoption of XHTML was small.

➔ In 2004, a proposal to refocus on HTML was discarded by the W3C, leading to outside development of HTML.

WHATWG

➔ Members of the W3C formed a new group: the Web Hypertext Application Technology Working Group (WHATWG).

➔ WHATWG didn't follow a consensus-based approach, so it was able to move much faster.

➔ In 2006, the W3C acknowledged that XHTML wasn't being adopted and work on HTML was resumed.

➔ Instead of starting from scratch, the W3C decided to use the work from WHATWG.

➔ Work on XHTML 2.0 ended in 2009.

W3C and WHATWG

➔ WHATWG continues working on HTML as a "living standard" (no versions).
https://html.spec.whatwg.org

➔ W3C HTML Working Group
https://www.w3.org/groups/wg/htmlwg

➔ Latest W3C HTML Recommendation (snapshots from WHATWG)
https://www.w3.org/standards/history/html

➔ Memorandum of Understanding Between W3C and WHATWG (2019)
https://www.w3.org/2019/04/WHATWG-W3C-MOU.html

  ➔ More details: https://www.w3.org/html

  ➔ More details: https://wiki.whatwg.org/wiki/W3C

HTML5

➔ HTML5 is a collection of features and technologies.

- ➔ Language / Markup features
- ➔ Document Model Definition (DOM)
- ➔ APIs for supporting JavaScript interaction with the DOM

HTML Fundamentals

Document Structure

- Document Structure defines the basic HTML document organization
  - `DOCTYPE` declares the document type and version
  - `html` element is the root container
  - `head` contains metadata and resources
  - `body` contains the visible content

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document Title</title>
</head>
<body>
    <!-- Content goes here -->
</body>
</html>
```

Elements and Tags

- **Elements** are the building blocks of HTML documents
- **Tags** mark the beginning and end of elements
- Most elements have opening and closing tags
- Some elements are self-closing (empty elements)

```
<!-- Standard element with content -->
<p>This is a paragraph element</p>

<!-- Self-closing elements -->
<img src="image.jpg" alt="An image">
<br>
<input type="text">
```

Attributes

- Attributes provide additional information about elements
- Attribute format: `name="value"`
- Some attributes are required (`src` for `img`)
- Some attributes are optional (`class`, `id`)
- Global attributes work on any element

```
<!-- Required attributes -->
<img src="image.jpg" alt="Description">

<!-- Optional attributes -->
<div id="unique-id" class="style-class">
<a href="https://example.com" target="_blank">

<!-- Boolean attributes -->
<input type="checkbox" checked>
<button disabled>
```

Content and Text Elements

- **Headings** define document structure (`h1-h6`)
- **Paragraphs** contain blocks of text
- **Lists** organize information
- **Inline elements** format text within blocks

```
<!-- Document structure -->
<h1>Main Title</h1>
<h2>Section Title</h2>

<!-- Text content -->
<p>Paragraph with <em>emphasized</em>

<!-- Multiple spaces and line breaks
are rendered as a single space -->

and <strong>important</strong> text.</p>

<!-- Lists -->
<ul>
    <li>Unordered list item</li>
    <li>Another item</li>
</ul>

<ol>
    <li>First ordered item</li>
    <li>Second ordered item</li>
</ol>
```

Links and Navigation

- **Anchors** create hyperlinks between pages
- Hyperlinks support following destinations:
  - Other pages
  - Sections within the same page
  - Files and resources
  - Email addresses

```
<!-- External links -->
<a href="https://example.com">Visit Website</a>

<!-- Internal page links -->
<a href="#section-id">Jump to Section</a>

<!-- Email links -->
<a href="mailto:contact@example.com">Send Email</a>

<!-- File downloads -->
<a href="document.pdf" download>Download PDF</a>
```

Media

- Media elements handle images, audio, and video content
- `img` element displays images
- `audio` and `video` elements handle multimedia content
- `figure` groups media with its caption
- picture adapts image sources based on device capabilities

```html
<!-- Basic image -->
<img src="image.jpg" alt="Description">

<!-- Figure with caption -->
<figure>
    <img src="image.jpg" alt="Description">
    <figcaption>Image caption</figcaption>
</figure>

<!-- Video with controls -->
<video controls>
    <source src="video.mp4" type="video/mp4">
    Your browser does not support video.
</video>
```

Semantic Containers

- Semantic containers organize document structure and meaning
  - `header` defines document or section headers
  - `nav` contains navigation elements
  - `main` holds the primary content
  - `article` represents independent content
  - `section` groups related content
  - `aside` contains supplementary content
  - `footer` defines document or section footers

```
<!-- Semantic containers -->
<header>Site header</header>
<nav>Navigation</nav>
<main>Main content</main>
<article>Article content</article>
<aside>Sidebar</aside>
<footer>Site footer</footer>

<!-- Generic containers -->
<div>Block container</div>
<span>Inline container</span>
```

Forms and User Input

- Form elements handle user input and data submission
  - `form` defines the container for user input
  - `input` creates interactive controls
  - `label` associates text with form controls
  - `button` triggers form submission or actions
  - `fieldset` groups related form controls
  - `select` creates dropdown menus
  - `textarea` enables multiline text input

```
<form action="/submit" method="post">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email">

  <label for="age">Age:</label>
  <input type="number" id="age" name="age" min="0">

  <fieldset>
      <legend>Preferences</legend>
      <input type="checkbox" id="newsletter" name="newsletter">
      <label for="newsletter">Subscribe to newsletter</label>
  </fieldset>

  <button type="submit">Submit</button>
</form>
```

HTML References

➔ **HTML: HyperText Markup Language | MDN**
https://developer.mozilla.org/en-US/docs/Web/HTML

➔ **Latest version of HTML**
https://www.w3.org/TR/html

➔ **WHATWG HTML Specification**
https://html.spec.whatwg.org/multipage

➔ **Dive Into HTML5**
https://diveintohtml5.info

➔ **HTML Dog: HTML, CSS and JavaScript tutorials**
https://htmldog.com

➔ **Chapter 2 - A history of HTML**
https://www.w3.org/People/Raggett/book4/ch02.html

Microdata

HTML Microdata

- Extension to define new attributes and embed simple machine-readable data in HTML documents.
- Goal: annotate content with machine-readable labels.
- Common use case: search engines can better 'understand' and index information that has been annotated using schema.org vocabulary.
- Microdata provides a mechanism to identify items and define their properties.
  - The `itemscope` attribute creates an item.
  - The `itemprop` attribute descends of itemscope and defines an item property.
  - With `itemtype` is possible to associate a vocabulary to an item.
  - An `itemid` can be used to define a global unique identifier for the item.

Microdata Example

➔ Defines an item with two properties.

```
<div itemscope>
  <p>Flavors in my favorite ice cream:</p>
  <ul>
   <li itemprop="flavor">Lemon sorbet</li>
   <li itemprop="flavor">Apricot sorbet</li>
  </ul>
 </div>
```

Schema.org

- Vocabularies define concepts and relationships used to describe and represent areas of concern. Can be very simple (one or two concepts) or very complex (thousands of terms).

- A shared vocabulary makes it possible to have a common understanding of defined concepts and relationships.

- Schema.org is a collaborative, community driven initiative to create, maintain, and promote the use of schemas for structured data on the web. Founded by Google, Microsoft, Yahoo, and Yandex.

- Schema.org defines more than 600 types and >900 properties. Such as CreativeWork, Book, Movie, Event, Organization, Person, Place, Restaurant, etc.

Microdata Example using Vocabulary

- Example using Schema.org vocabulary.
- Defines an item of the type LocalBusiness, as defined by the Schema.org vocabulary, containing three properties, one of which is a item of the type PostalAddress, containing four properties.

```
<div itemscope itemtype="http://schema.org/LocalBusiness">
  <h1 itemprop="name">Beachwalk Beachwear & Giftware</h1>
  <span itemprop="description"> A superb collection [...].</span>
  <div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
    <span itemprop="streetAddress">3102 Highway 98</span>
    <span itemprop="addressLocality">Mexico Beach</span>,
    <span itemprop="addressRegion">FL</span>
  </div>
  Phone: <span itemprop="telephone">850-648-4200</span>
</div>
```

HTML Microdata References

➔ W3C Editor’s Draft - Microdata (April 2021)
https://w3c.github.io/microdata

➔ HTML Standard Microdata Specification
https://html.spec.whatwg.org/#microdata

➔ Schema.org
https://schema.org

➔ Semantic Web (aka Web of Data)
https://www.w3.org/standards/semanticweb

Web Components

Why Web Components?

- ➔ Standard HTML provides limited support for component reusability
- ➔ CSS encapsulation presents challenges in complex applications
- ➔ Code duplication when same UI pattern is needed in multiple places
- ➔ Difficult to share UI elements between different projects

Example: A Simple User Card

- ➔ Repetitive HTML structure
- ➔ CSS classes might conflict with other elements
- ➔ JavaScript needs to find and handle each element separately

```
<div class="user-card">
  <img class="user-avatar" src="user1.jpg">
  <div class="user-name">John Doe</div>
  <div class="user-title">Software Developer</div>
  <button class="follow-button">Follow</button>
</div>

<!-- Same structure, different data -->
<div class="user-card">
  <img class="user-avatar" src="user2.jpg">
  <div class="user-name">Jane Smith</div>
  <div class="user-title">UX Designer</div>
  <button class="follow-button">Follow</button>
</div>
```

Web Components

- ➔ Cleaner, more readable HTML
- ➔ Encapsulated styles and behavior
- ➔ Reusable across different pages or projects
- ➔ Works like built-in HTML elements (like <video> or <select>)

```
<user-card
  avatar="user1.jpg"
  name="John Doe"
  title="Software Developer">
</user-card>

<user-card
  avatar="user2.jpg"
  name="Jane Smith"
  title="UX Designer">
</user-card>
```

Web Components References

➔ MDN Web Components
https://developer.mozilla.org/en-US/docs/Web/Web_Components

➔ MDN Web Components Examples
https://github.com/mdn/web-components-examples

➔ WebDev: Web Components
https://web.dev/articles/web-components

➔ WebComponents.org
https://www.webcomponents.org