Week 03 - HTML5

Tomáš Pitner

March 2, 2023

Outline

- Introduction: HTML5 as the foundation of WWW
- Commonly used elements
- WAI ARIA
- Regular DOM, Virtual DOM, Shadow DOM & AOM
- Browser rendering
- Minification, Compression

Introduction to HTML5

- Part of basic web standards, currently on version HTML 5.2
- HTML 5.3 under preparation as living standard, see https://html.spec.whatwg.org/multipage/
- HTML5 is used on websites and mobile and desktop devices/apps
- Previous popular versions: XHTML and HTML 4 still in use
- XHTML was the only HTML being a strict XML markup
- On the other hand, HTML is based on the old Structured Generalized Markup Language (SGML, see e.g. http://webserver.ics.muni.cz/bulletin/articles/106.html)
- HTML5 needs
  - CSS for styling
  - JavaScript (ECMAScript) for scripting - interactivity

What is HTML5

- HyperText Markup Language (HTML) 5
- **HTML5 is not XML** in the strict sense (some syntax rules are omitted)
- HTML5 is similar to original (SGML-based) versions of HTML
- Extends previous versions by many new elements and API
    - multimedia (`<audio>` , `<video>` , `<picture>` , `<embed>` ...)
    - document type, structure, metadata (`<article>` , `<section>` , `<header>` , `<main>`)
    - navigation and controls (`<nav>` , `<menu>` , `<progress>` , `<menuitem>`)
    - semantic markup (`<time>` , `<samp>` , `<kbd>`)
    - support for data in the document (`<data>` , `<datalist>`)

Example

```html
<!DOCTYPE html>
<html lang="en">
  <head>
      <title>A simple HTML document</title>
  </head>
  <body>
      <p>Hello World!<p>
  </body>
</html>
```

- `<!DOCTYPE html>` is the document type declaration (always like this)
- `<head>` - contains metadata of the document, such as `<title>`, or links to css styles
- `<body>` - contains the content, i.e. text, headings, links to js files...

Basic syntax

- File extension `.html` (preferred) or `.htm`
- Element names (tags) are case-insensitive (lower case is best practice)
- Attribute values are usually *case-sensitive*!
- Types of elements:
    - **Block elements** = create block (rectangle), e.g. paragraph
    - **Inline elements** = change formatting of some objects (text) in line
    - **Semantic elements** = intended meaning or purpose (`article` , `main` …)
    - **Non-semantic elements** = other elements without meaning (`div` , `span` …)

Elements

- HTML5 is a concrete markup, so only defined elements may be used
- They also must be used only in *proper context*, i.e. nested only in certain elements
- An element has one of these meanings:
  - *structures* the text
  - places *images, frames, video...*
  - defines *styles*
  - defines *scripts*

One paragraph of text with one line break inside

```
<p>This paragraph contains <br> a line break.</p>
```

- Note: `<br>` is an *empty* element
- In contrast to XML, we do not need to write "well-formed" `<br/>`

Nesting

Elements must be nested within each other properly.

```html
<!-- Correct way of nesting -->
<div><span>These tags are nested properly.</span></div>

<!-- Incorrect way of nesting -->
<span><div>These tags are not nested properly.</span></div>
```

Exact nesting rules (which element can appear in which context) are defined in HTML5 Specification.

Comments

- Same as (general) XML comments
- HTML comment begins with `<!--`, and ends with `-->`

```html
<!--
  Example of multiline comment
-->

<!-- Single line comment -->
```

Element categories

- Basic structural `<html>`, `<head>`, `<body>`
- Section-related elements such as `<main>`, `<article>`, `<nav>` defining semantics
- Common block elements, e.g. `<h1>`, `<p>`
- Common inline elements, e.g. `<em>`

Basic structural elements

- `<html>` - complete enclosure for the entire page including head and body
- `<head>` - mostly metadata, e.g. title, links to styles, fonts, and scripts
- `<body>` - page (textual) content

Section-related elements

They define semantics of parts of the document.

- `<article>` - self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable
- `<section>` - generic standalone section of a document, may be nested
- `<main>` - dominant content of the `<body>` of a document
- `<aside>` - subdominant, less important content, may appear aside of main text
- `<footer>` - footer for its nearest ancestor, e.g. page or section footer
- `<header>` - header for its nearest ancestor
- `<nav>` - major block of navigation links for the page

Common block elements

- `<div>` - division (general block element, usually styled)
- `<p>` - paragraph (similar to `<div>`)
- `<h1>` through `<h6>` - headings
- `<form>` - form for inputs to submit to server
- `<textarea>` - text area (big text fields)
- `<ol>`, `<ul>` - ordered and unordered lists
- `<li>` - list item

Block elements **must not** appear within inline elements!

Common inline elements - styling

- `<a>` - insert HTML link or link anchor
- `<span>` - general inline element, usually styled
- `<code>` - preformatted (non-proportional) font used
- `<input>`, `<button>` - text input fields, buttons
- `<img>` - insert image (behaves like inline-block)

Common inline elements - semantics

The following elements are mainly used for inline **semantics and not styling**:

- `<strong>` - emphasized (typically bold) text
- `<b>` - bold text
- `<em>` - emphasized (typically italics) text
- `<i>`, `<u>` - text in italics, underlined

Attributes

Specify details of an element (e.g. image source, target URL, title, id)

```html
<img src="images/smiley.png" width="30" height="30" alt="Smiley">
<a href="https://www.google.com/" title="Search Engine">Google</a>
<abbr title="Hyper Text Markup Language">HTML</abbr>
<input type="text" value="John Doe">
```

General attributes

Some attributes can be used by most elements:

- `id` - unique ID of the element within the document
- `class` - CSS class for the element
- `style` - in-place style specification for the element (e.g. color, size)
- `title` - for many elements - specifies title (e.g. for accessibility)
- `lang` - language code for the text content, mainly used on `html` tag

Commonly used elements

Headings

- Define *structure* of document - i.e. start of chapter, section
- Not just bigger font
- Important to use for search engines, SEO
- The different levels produce headings of decreasing font size

```html
<h1>Heading level 1</h1> <!-- biggest -->
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6> <!-- smallest -->
```

Text paragraphs

- paragraphs start with `<p>` and may (preferred) or may not end with `</p>` (when used as separator)
- `<br>` means just a line break, such as `Shift` + `Enter` in Word (**not used for positioning/layouts**)

```html
<p>This is a paragraph <br> with line break.</p>
<p>This is <br>another paragraph <br> with line breaks.</p>
```

Paragraphs are usually (but not always) styled by assigning to a class

```html
<p class="my-nice-para">This is a styled paragraph.</p>
```

Horizontal rulers

Horizontal line between block elements, e.g. paragraphs

```html
<p>This is a paragraph.</p>
<hr>
<p>This is another paragraph.</p>
```

Preformatted text

```html
<pre>
  Twinkle, twinkle, little star,
  How I wonder what you are!
  Up above the world so high,
  Like a diamond in the sky.
</pre>
```

Address

```html
<address>
Mozilla Foundation<br>
331 E. Evelyn Avenue<br>
Mountain View, CA 94041, USA
</address>
```

Whitespaces

- you can use either space char (ASCII 32), tab char, newline... everything is a white space
- if you use a fixed whitespace entity `&nbsp;` (**non-breaking space**)

```html
<p>This paragraph has multiple&nbsp;&nbsp;&nbsp;spaces.</p>
```

- then you get
  - exact number of fixed-width spaces
  - connecting the text before and after as non-space characters

Links

- Allow to click and browse from page to another

```
<a href="page.html">link to Page</a>
```

- Can point and go even inside of document to specific anchor/ID:

```
<a href="page.html#topicA">Go to TopicA</a>
```

- To open a new window or tab:

```
<a target="_blank" href="https://website.com">Open in new tab</a>
```

Text formatting

The following example contains elements with a purely semantic purpose

```
<p>This is <b>bold text</b>.</p>
<p>This is <strong>strongly important text</strong>.</p>
<p>This is <i>italic text</i>.</p>
<p>This is <em>emphasized text</em>.</p>
<p>This is <mark>highlighted text</mark>.</p>
<p>This is <code>computer code</code>.</p>
<p>This is <small>smaller text</small>.</p>
<p>This is <sub>subscript</sub> and <sup>superscript</sup> text.</p>
<p>This is <del>deleted text</del>.</p>
<p>This is <ins>inserted text</ins>.</p>
```

- `<strong>` and `<b>` are usually the same but may differ, `<strong>` is more general
- `<em>` and `<i>` are usually the same but may differ, `<em>` is more general

Quotes

Block quote - the block element is usually indented and in a different font

```
<blockquote>
    <p>Learn from yesterday, live for today, hope for tomorrow.</p>
    <cite>— Albert Einstein</cite>
</blockquote>
```

Inline quote

```
<p>According to WHO: <q>Health is a state of complete physical and mental well-being.</q></p>
```

Citations

- `<cite>` usually appears inside `<blockquote>`
- Reference to a real source, e.g. person, movie, book
- Usually appears as italics
- Does not produce any link

```
<p>My favorite movie is <cite>Star Wars</cite>.</p>
<p>My another favorite movie is <cite>Harry Potter</cite>.</p>
```

Abbreviations

- `<abbr>` explains an abbreviation

```html
<p>The <abbr title="World Wide Web Consortium">W3C</abbr> is the main international
standards organization for the <abbr title="World Wide Web">WWW or W3</abbr>.
It was founded by Tim Berners-Lee.</p>
```

Images

Inline element so it can be used also for small images online

```html
<img src="kites.jpg" alt="Flying Kites">
<img src="sky.jpg" alt="Cloudy Sky">
<img src="balloons.jpg" alt="Balloons">
```

- `src` points to URL with the image source (file)
- `alt` provides text alternative (description) for accessibility or slow connection
- Note that `<img>` is an empty element, no need to write `<img src=... />`
- Can be styled (via `class` or `style`)

Picture

Advanced new HTML5 element for multiple versions (e.g. resolutions) of an image

```html
<picture>
    <source media="(min-width: 1000px)" srcset="logo-large.png">
    <source media="(max-width: 500px)" srcset="logo-small.png">
    <img src="logo-default.png" alt="My logo">
</picture>
```

There is also image map `<map>` element for creating clickable map

Video

- Complete container for linking and showing video clips
- `source` allows to specify multiple formats available

```html
<video controls width="250" height="200" muted>
    <source src="/media/cc0-videos/flower.webm"
            type="video/webm">
    <source src="/media/cc0-videos/flower.mp4"
            type="video/mp4">
    Download the
    <a href="/media/cc0-videos/flower.webm">WEBM</a>
    or
    <a href="/media/cc0-videos/flower.mp4">MP4</a>
    video.
</video>
```

Figure

- Even more abstract element than `picture`, may represent ilustration (picture), chart, diagram, video… in a paper or book.
- May have `figcaption`

```
<figure>
    <img src="/media/cc0-images/elephant-660-480.jpg"
         alt="Elephant at sunset">
    <figcaption>An elephant at sunset</figcaption>
</figure>
```

Tables

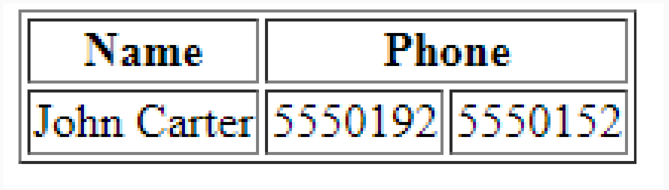
Used for showcasing the data, **not for formatting nor layouts**! (we use `<div>` s and CSS styles for layouts)

```
<table>                   <!-- tag enclosing the entire table -->
    <tr>                  <!-- table rows contain cells -->
        <th>No.</th>      <!-- table heading cell (usually formatted differently) -->
        <th>Name</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>1</td>        <!-- simple table cell -->
        <td>Peter</td>
        <td>16</td>
    </tr>
</table>
```

- Details inside of `<table>` element: `<thead>` , `<tbody>` , and `<tfoot>`

Cells spanning multiple columns

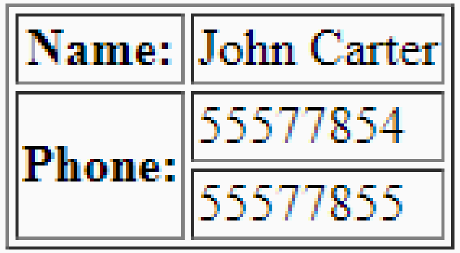
```
<table>
    <tr>
        <th>Name</th>
        <th colspan="2">Phone</th>
    </tr>
    <tr>
        <td>John Carter</td>
        <td>5550192</td>
        <td>5550152</td>
    </tr>
</table>
```



Name	Phone	
John Carter	5550192	5550152

Cells spanning multiple rows

```
<table>
    <tr>
        <th>Name:</th>
        <td>John Carter</td>
    </tr>
    <tr>
        <th rowspan="2">Phone:</th>
        <td>55577854</td>
    </tr>
    <tr>
        <td>55577855</td>
    </tr>
</table>
```



Name:	John Carter
Phone:	55577854
	55577855

Captions

- `<caption>` - allows to specify table caption (e.g. title)
- can be positioned elsewhere by styles' `caption-side` property
- must be the first child element of `<table>`

```
<table>
    <caption>Users Info</caption>
    <tr>
        <th>No.</th>
        <th>Name</th>
        <th>Age</th>
    </tr>
    ...
```

Unordered list

Used to create a **bullet point list** of related items.

```html
<ul>
    <li>Chocolate Cake</li>
    <li>Black Forest Cake</li>
</ul>
```

Ordered list

Used to create a **numbered list** of related items.

```
<ol>
    <li>Fasten your seatbelt</li>
    <li>Starts the car's engine</li>
</ol>
```

Description list

Used to create a **list of terms** and their descriptions.

```html
<dl>
    <dt>Bread</dt>
    <dd>A baked food made of flour.</dd>
    <dt>Coffee</dt>
    <dd>A drink made from roasted coffee beans.</dd>
</dl>
```

Miscelaneous elements

- Often dynamic or interactive
- `details` - disclosable widget, shows more info then
- `datetime` - shows date and time for machine processing
- `progress` - shows progress bar

Element `details`

- `<details>` creates a disclosure widget in which information is visible only when the widget is toggled

```html
<details>
    <summary>Details</summary>
    Something small enough to escape casual notice.
</details>
```

Element `datetime`

- specifies either datetime in 24 hrs representation, a Gregorian calendar datetime, or *duration*

```html
<time datetime="20:00">20:00</time>
<time datetime="PT2H30M">2h 30m duration</time>
```

Element `progress`

- draws a progressbar, may containt percentage value

```html
<label for="file">File progress:</label>
<progress id="file" max="100" value="70"> 70% </progress>
```

Forms

Input elements

- Direct input from user - entering text, selecting option
- `<input>` but also others (`<textarea>` , `<select>` and `<option>`)
- Uniquely identified in form by both `name="username"` and `id="username"`

```
<form>
  <input type="text" name="username" id="username">
  <input type="password" name="password" id="password">
  <input type="hidden" name="browser" value="chrome">
</form>
```

Full list of `input` s (1)

- `<input type="button">` - clickable button (usually activates a script)
- `<input type="checkbox">` - one checkbox
- `<input type="color">` - color selection, e.g. palette
- `<input type="date">` - for input of date
- `<input type="datetime-local">` - for input of local date and/or time
- `<input type="email">` - for entering email (checks validity)

Full list of `input` s (2)

- `<input type="file">` - shows file selection dialogue (usually for upload)
- `<input type="hidden">` - not shown but heavily used for sending inputs to server
- `<input type="image">` - shows image file selection dialogue
- `<input type="month">` - month of year selection
- `<input type="number">` - selects a number (spinner, e.g. 1 to 100 inclusively)
- `<input type="password">` - allows to enter password (shows `*` instead of characters)

Full list of `input` s (3)

- `<input type="radio">` - one radiobutton
- `<input type="range">` - selects number from given range (imprecise)
- `<input type="reset">` - reset button
- `<input type="search">` - text search input field
- `<input type="submit">` - submit button
- `<input type="tel">` - phone number input

Full list of `input` s (4)

- `<input type="text">` - plain text input
- `<input type="time">` - time input (validates, may use time selector)
- `<input type="url">` - URL input (validates)
- `<input type="week">` - entry of a year plus the ISO 8601 week number

Label

- `<label>` provides description to specific input field

```html
<form>
  <label for="username">Username:</label>
  <input type="text" name="username" id="username">
</form>
```

Checkbox and Radio button set

- Radio buttons are selected as *mutually exclusive*
- Radio buttons are usually round symbols paired with a label

```html
<form>
  <input type="radio" name="gender" id="female">
  <label for="female">Female</label>

  <input type="radio" name="gender" id="male">
  <label for="male">Male</label>
</form>
```

Checkboxes

- Checkboxes can be *independently* selected
- Checkboxes are usually square symbols with indication when selected

```
<form>
  <input type="checkbox" name="sports" id="soccer">
  <label for="soccer">Soccer</label>

  <input type="checkbox" name="sports" id="cricket">
  <label for="cricket">Cricket</label>
</form>
```

File select

- used for file upload
- appears as button to select file

```html
<form>
  <label for="file-select">Upload:</label>
  <input type="file" name="upload" id="file-select">
</form>
```

Textarea

- Same as text field but multiline (rectangle area of certain number of rows and columns)

```html
<form>
  <label for="address">Address:</label>
  <textarea rows="3" cols="30" name="address" id="address"></textarea>
</form>
```

Select from list

- Select from dropdown list
- Returns the selected option `value`

```html
<label for="city">City:</label>
<select name="city" id="city">
  <option value="sydney">Sydney</option>
  <option value="melbourne">Melbourne</option>
  <option value="cromwell">Cromwell</option>
</select>
```

Buttons

Either generic button `<button>` or specific `reset` and `submit` (*default*) buttons

```html
<form action="action.php" method="post">
  <label for="first-name">First Name:</label>
  <input type="text" name="first-name" id="first-name">

  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>
```

Grouping inputs

- logically (and visually) group related inputs
- may contain general description - `<legend>`

```html
<fieldset>
  <legend>Contact Details</legend>
  <label>Email Address: <input type="email" name="email"></label>
  <label>Phone Number: <input type="text" name="phone"></label>
</fieldset>
```

Form element

- Encloses the entire form to be submitted
- Usually contains at least submit button
- The input data is sent via HTTP `POST` (typically) or `GET` (by default) methods
- **GET sends the data in URL** while **POST sends data in request body**
- Controlled by form attributes:
  - `name` of the form (most useful for scripting)
  - `action` - target for processing the form
  - `method` - either `get` or `post`
  - `target` - where to display the response (default: `_self`)
  - `enctype` - character encoding for POST data

```
<form action="action.php" method="post"></form>
```

Accessibility

Web Accessibility Initiative’s Accessible Rich Internet Applications (WAI ARIA)

- Describe roles, states and properties
- Provides recognition and usability for users with assistive technology (Screen Reader, Keyboard navigation)
- Structured access to the website
- Types of Aria attributes:
    - Indication of widgets (`role="alert"` , `role="modal"` , `aria-label="Choose action"`)
    - Indication of structure (`role="main"` , `role="article"`)
    - Indication of state (`aria-expanded="false"`)
    - Indication of properties (`aria-haspopup="true"`)
- Also known as **a11y** (accessibility)

Example for image

```html
<p id="dimg">
A long description of our paragliding trip ...
</p>

<div>
<img src="takeoff.png"
         alt="Getting ready to take off"
         aria-labelledby="limg"
         aria-describedby="dimg">
</div>
```

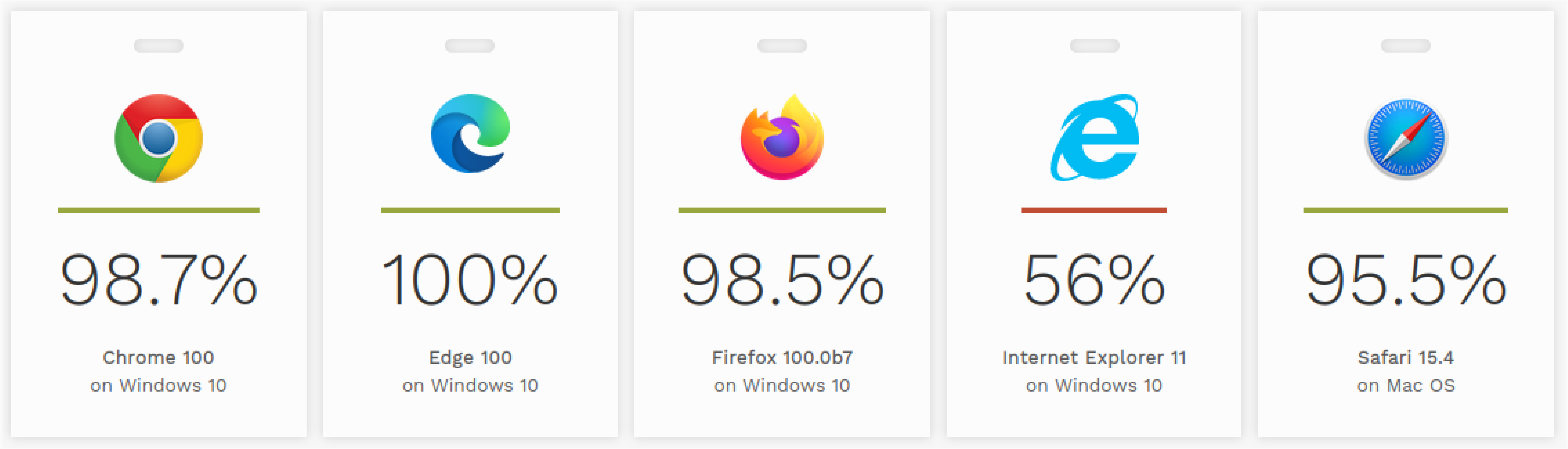
Example with dropdown widget

```html
<div class="dropdown">
  <button type="button" aria-haspopup="true" aria-expanded="false" id="dropdown-1">
    Choose action
  </button>
  <div aria-labelledby="dropdown-1">
    <a href="#">First action</a>
    <a href="#">Second action</a>
  </div>
</div>
```

HTML5 and Accessibility

`<nav>` vs `<div role="navigation">`

- Not fully supported, thus recommended to use `aria*` attributes



98.7%
Chrome 100
on Windows 10

100%
Edge 100
on Windows 10

98.5%
Firefox 100.0b7
on Windows 10

56%
Internet Explorer 11
on Windows 10

95.5%
Safari 15.4
on Mac OS

- See more at https://stevefaulkner.github.io/HTML5accessibility/

Resources

- HTML5 Tutorial on TutorialRepublic.com
- https://dev.opera.com/articles/introduction-to-wai-aria/
- https://wicg.github.io/aom/spec/
- https://developer.mozilla.org/en-US/docs/Web/Performance/Critical_rendering_path
- https://medium.com/jspoint/how-the-browser-renders-a-web-page-dom-cssom-and-rendering-df10531c9969
- https://shortdiv.com/posts/aom-at-me-bro/
- https://reactjs.org/docs/reconciliation.html
- https://developers.google.com/web/fundamentals/performance/critical-rendering-path/render-tree-construction
- https://developer.mozilla.org/en-US/docs/Web/Performance/Critical_rendering_path