Web Technology

HTML5

Dr. Navanath Saharia
Indian Institute of Information Technology Senapati, Manipur
nsaharia@iiitmanipur.ac.in

My First Program

- Open Gedit Text Editor either (terminal `$gedit` or `show applications` menu)
- Type the following lines in the Gedit Text Editor

```
1  <!DOCTYPE html>
2  <html>
3     <head>
4        <title>Homepage</title>
5     </head>
6     <body>
7        <h1> Welcome to my Homepage</h1>
8     </body>
9  </html>
```

- Save and run the `HTML` program
- Open in a Web Browser and see the output

Head element

- The head element represents a collection of metadata for the Document.

```
<!doctype html>
<html>
 <head>
  <title>Introduction to HTML: Week 2</title>
  <base href="https://www.abc.in">

  <meta charset="utf-8" name="application-name" content="Second week discussion">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <link rel="stylesheet" href="wk2-min.css" title="minimal version" type="text/css">
  <link rel="stylesheet alternate" href="wk2-complete.css" title="complete version">

  <script> ..... </script>
  <style> ..... </style>
 </head>
```

Elements of Head

- The `<title>` element defines the document's title or name
- Use: The content of the title is one of the components used by search engine algorithms to decide the order in which to list pages in search results. In general, a longer, descriptive title performs better than short or generic titles.
- Syntax: `<title>Title of the document</title>`

Elements of Head

- The `<title>` element defines the document's title or name
- Use: The content of the title is one of the components used by search engine algorithms to decide the order in which to list pages in search results. In general, a longer, descriptive title performs better than short or generic titles.
- Syntax: `<title>Title of the document</title>`
- The `<base>` element specifies a base URL for all relative URLs within a document
- Syntax: `<base href="URL">`. Example: `<base href="https://www.abc.in">`
- The `href` attribute of `<base>` specifies the base URL `https://www.abc.in` for all relative URLs in the document. For example, file path used by `href` attribute of `<a>` element and `src` attribute of `<img>` element are relative. That relative path will be resolve based on specified base URL in the `<base>` element

Elements of Head

- The `<title>` element defines the document's title or name
- Use: The content of the title is one of the components used by search engine algorithms to decide the order in which to list pages in search results. In general, a longer, descriptive title performs better than short or generic titles.
- Syntax: `<title>Title of the document</title>`
- The `<base>` element specifies a base URL for all relative URLs within a document
- Syntax: `<base href="URL">`. Example: `<base href="https://www.abc.in">`
- The `href` attribute of `<base>` specifies the base URL `https://www.abc.in` for all relative URLs in the document. For example, file path used by `href` attribute of `<a>` element and `src` attribute of `<img>` element are relative. That relative path will be resolve based on specified base URL in the `<base>` element
- The `<meta>` element represents various kinds of metadata that cannot be expressed using other elements belongs to head. Metadata is information about the document that is not displayed on the web page but is used by browsers, search engines, and other web services

http-equiv attribute of <meta> element

- http-equiv is an enumerated attribute that provides information to the browser about how the content should be interpreted.
  - Enumerated attribute → an attribute that can only take one of a predefined set of values.
  - The state for such an attribute is derived by combining the attribute's value, a set of keyword/state mappings given in the specification of each attribute
  - Example: type attribute of <input> element, target attribute of <a> element

Keyword	State	Brief description
content-language	Content language	Sets the pragma-set default language.
content-type	Encoding declaration	An alternative form of setting the charset.
default-style	Default style	Sets the name of the default CSS style sheet set.
refresh	Refresh value	Acts as a timed redirect.
x-ua-compatible	X-UA-Compatible	Encourages Internet Explorer to more closely follow the
content-security-policy	Content security policy	Enforces a Content Security Policy on a Document.

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta http-equiv="refresh" content="5">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta http-equiv="Content-Security-Policy" content="default-src 'self'; script-src 'self' 'uns
```

<link> element

- The <link> specifies relationships between the current document and an external resources, such as, stylesheets, icon files, or other documents

```
<link rel="stylesheet" type="text/css"  href="abc.css">
<link rel="alternate"  type="application/rss+xml" href="rssFeed.xml">
```

<link> element

- The <link> specifies relationships between the current document and an external resources, such as, stylesheets, icon files, or other documents

```
<link rel="stylesheet" type="text/css"  href="abc.css">
<link rel="alternate"  type="application/rss+xml" href="rssFeed.xml">
```

- rel attribute defines the relationship between the current document and the linked resource. Value includes stylesheet, icon, alternate, canonical, prefetch, next, prev, author, help
  - rel="stylesheet" indicates that the linked document is a style-sheet

```
<link rel="stylesheet" href="default.css" />
```

  - rel="alternate" indicates that the linked document is an alternative of the current document

```
<link rel="alternate stylesheet" href="highContrast.css" title="High contrast"
<link rel="alternate" href="/en/index.pdf " hreflang="en" type="application/pdf
```

  - rel="icon" represents the icon for the current document

```
<link rel="icon"  type="image/png"  href="favicon.png">
```

  - rel="preload" indicates that the browser should preload the resource. Content type, such as ArrayBuffer, JSON/BSON, font, image, script, and style can be preloaded.

```
<link rel="preload" href="myFont.woff2" as="font" type="font/woff2">
<link rel="preload" href="style.css" as="style" />
<link rel="preload" href="main.js" as="script" />
```

<link> element

- `rel="author"` indicates about the author of the current document
- `rel="canonical"` indicates that the linked resource is the canonical (preferred) version of the document, particularly useful for specifying the preferred version of a URL to search engines
- `href` attribute specifies the URL and `hreflang` indicates the language of the linked resource.
- `type` attribute specifies the MIME type of the linked resource
- `sizes` attribute specifies the size of the icon
- `as` attribute is used with `rel="preload"` that specifies the type of content being loaded by the `<link>`, which is necessary for request matching, application of correct content security policy, and setting of correct Accept request header
- `media` attribute specifies the media type for which the linked stylesheet is intended. It is typically used to specify different stylesheets for different devices or screen sizes
- The resource will only be loaded if the media condition is true

```
<link href="print.css" rel="stylesheet" media="print"/>
<link href="mobile.css" rel="stylesheet" media="all"/>
<link href="desktop.css" rel="stylesheet" media="(max-width: 600px)" />
<link href="highres.css" rel="stylesheet" media="screen and (min-width: 601px)" />
```

<style> element

- The <style> element is used to define internal/embedded cascading styles.

```
<style>
    a{
        color: blue;
        text-decoration: none;
        }
</style>
```

- Styles defined within the <style> element have a higher priority than external stylesheets linked via the <link> element and lower priority than inline styles applied directly to HTML elements using the style attribute.
- media attribute of <style> element defines, which media the style should be applied to. Its value is a media query, which defaults to all if the attribute is missing.
  - Media queries allows to apply CSS styles depending on a device's
    - Media type, such as, print, screen and all
    - Media feature that describes specific characteristics of the user agent (browser) such as screen resolution, orientation, aspect ratio, browser viewport width/height
    - User preferences, such as, preferring reduced motion, data usage, and transparency

<script> element

- The <script> element is used to
  - Embed JavaScript code

```
<script>alert("Hello Web Tech!"); </script>
```

  - Link external JavaScript source file

```
<script src="abc.js"></script>
```

    src attribute specifies the URL of an external script file to be used. If this attribute is present, the contents of the <script> element are ignored.
  - Embed executable codes. The type attribute indicates the type of script represented. The value includes text/javascript, importmap, module

```
<script id="data" type="application/json">{
      "userId": 1234,
      "userName": "abc bcd"
}
</script>
<script>
  const userInfo = JSON.parse(document.getElementById("data").text);
  console.log("User information: %o", userInfo);
</script>
```

Anchor element

- Anchor or <a> element is used to create hyperlinks, allowing users to navigate to another web-page or resource, which is one of the fundamental building blocks of the web.
- href attribute specifies the URL of the destination where the hyperlink leads. It can be a(n)
  - Absolute URL

```
<a href="https://www.abc.in/index.php">Abc Inc.</a>
```

  - Relative URLs

```
<a href="index.php">Abc Inc.</a>
```

  - Element on the same page

```
<a href="#sectionID">Jump to section/heading/ .....</a>
<h3 id="sectionID">Landing from previous section/heading .... </h2>
```

  - Email address

```
<a href="mailto:info@webtech.in">Contact us</a>
```

  - Telephone numbers

```
<a href="tel:+91.123.456.0123">Call: +91 123 456 0123</a>
```

  - A skip link is a link placed as early as possible in <body> content that points to the beginning of the page's main content.

Anchor element

- `target` attribute specifies, where to open the linked document. Common values include:
  - `target = "_self"` : Opens the linked document in the same frame or window (default)
  - `target = "_blank"`: Opens the linked document in a new window or tab.
  - `target = "_parent"`: Opens the linked document in the parent frame.
  - `target = "_top"`: Opens the linked document in the full body of the window

```
<a href="https://www.abc.in/index.php">Abc Inc.</a>
```

- `download` attribute, when present, suggests that the target resource is downloadable

```
<a href="files/document.pdf" download>Download PDF</a>
```

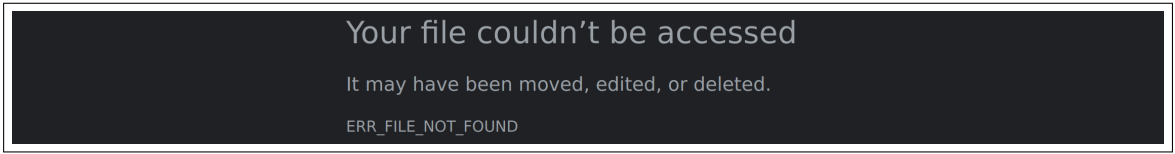
- `title` attribute provides additional information about the linked document, typically displayed as a tool-tip when the user hovers over the link.

```
<a href="https://www.abc.in" title="Visit Example Website">Example Website</a>
```

- `aria-label` attribute provides an accessible name for the link when the visible anchor text alone is not sufficient for screen readers or other assistive technologies.

```
<a href="https://www.abc.in" aria-label="Visit Example Website">Example Website</a>
```

Anchor element



- File not found error
  - Make sure that the specified
    - file is in the same directory
    - if file is in different directory, specify the filename along with the path
    - otherwise you will get `"File` or `Page not found"` error

List

- Structured ways to present information on web page by grouping the related content together
- Represents a list item based on type of parent element `<ol>`, `<ul>` or `<menu>`
  - `Ordered List`: `<ol>` is used to create an ordered list, where the list items are displayed with ordered number or alphabet

```
<ol>
    <li>Web Technology</li>
    <li>Signal and Systems</li>
    <li>Object Oriented Programming</li>
</ol>
```

1. Web Technology
2. Signal and Systems
3. Object Oriented Programming

  - `Unordered List`: `<ul>` is used to create an unordered list, where the list items are displayed with bullet points

```
<ul>
    <li>Web Technology</li>
    <li>Signal and Systems</li>
    <li>Object Oriented Programming</li>
</ul>
```

- Web Technology
- Signal and Systems
- Object Oriented Programming

Nested List

- `Nested list` → A list that contains another list(s) within one or more of its `<li>` elements

```
<ul>
    <li>Module I
        <ul>
            <li>Chapter I</li>
            <li>Chapter II</li>
        </ul>
    </li>
    <li>Module II
        <ul>
            <li>Chapter III</li>
            <li>Chapter IV</li>
        </ul>
    </li>
</ul>
```

Description List

- `<dl>` element provides a semantic way to represent lists of terms and their descriptions making content more structured and accessible that contains one or more pairs of `<dt>` (description term) and `<dd>` (description) elements.

- The `<dt>` element is used to define a term or a name in the list and the `<dd>` element is used to define the description of the term. Typically, `<dd>` comes after `<dt>`

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

nav

- <nav> element is used to create well-structured, accessible, and SEO-friendly navigation systems
- <nav> wraps an unordered list (<ul>) containing navigation links (<li> with <a>), providing semantic meaning to the collection of links as a navigation menu.

```
<nav> <ul>
    <li><a href="home.php">Home</a></li>
    <li><a href="about.html">About</a></li>
    <li><a href="#contact">Contact</a></li>
</ul> </nav>
```

```
<nav><a href="home.php">Home</a>   <a href="about.html">About</a>   <a href="#co
```

- <menu> and <ul> elements both represent an unordered list of items. However, <ul> primarily contains items for display, while <menu> was intended for interactive items

```
<menu>
        <li><a href="home.php">Home</a></li>
        <li><a href="about.html">About</a></li>
        <li><a href="#contact">Contact</a></li>
</menu>
```

head and header

- `<head>` element contains machine-readable metadata information about the document, such as, its title, scripts, and style sheets.

```
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Document title</title>
  </head>
```

- `<header>` element is commonly used to create the header section of a webpage, which usually includes the site's branding, main navigation links, and additional functionality to the user.

```
<header>
  <h1>Main Page Title</h1>
  <img src="banner.jpg" alt="Abc incorporation" />
</header>
```

- `<header>` element represents human-visible information, such as, introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also a logo, a search form, an author name, and other elements. However, the `<head>` element is used to provide information about the document to the browser, search engine optimization and accessibility engine.

Heading elements

- Heading elements, <h1> to <h6> represents six levels of section headings used by user agents. <h1> is the highest section level and <h6> is the lowest. By default, all heading elements create a block-level box in the layout, starting on a new line and taking up the full width available in their containing block.
- Standard: One <h1> per page and nest headings without skipping levels

```
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
```

- <hgroup> element allows the grouping of a heading with any secondary content, such as subheadings, an alternative title, or tagline. Each of these types of content represented as a <p> element within the <hgroup>

```
<hgroup id="document-title">
    <h1>HTML: Living Standard</h1>
    <p>Last Updated 12 July 2022</p>
  </hgroup>
```

Embeding an Image

- `<img>` element is used to embed image(s) in an HTML document.

```
<img src="abc.png" width="300" height="200" alt="Example Image">
```

- Using Data URIs: Data URIs allow you to embed image data directly into the HTML document

```
<img src="data:image/png; base64,bG9nby5wbmcK" alt="Example Image">
```

- `<svg>` element is used to embed Scalable Vector Graphics content

```
<svg width="200" height="100">
    <rect width="100%" height="100%" fill="lightblue"/>
    <circle cx="100" cy="50" r="48" fill="orange" stroke="green" stroke-width="1"/>
    <text x="50%" y="50%" text-anchor="middle" dy=".3em" fill="white">SVG Example</text>
</svg>
```

- Using background-images in CSS

```
<style>
  .imgContainer{background-image: url('abc.png'); width: 300px; height: 200px;}
</style>
<div class="imgContainer"></div>
```

img vs figure vs picture

- `<img>` is used to embed image(s) in an HTML document.

```
<img src="abc.jpeg" width="300" height="200" alt="Example Image">
```

- `<figure>` is used to provide semantic structure by grouping together an image with its caption. The `<figure>` element may nest `<audio>`, `<video>` and `<figcaption>`

```
<figure>
    <img src="abc.jpeg"> <figcaption>Caption for the image</figcaption>
</figure>
```

- `<picture>` is used to ensure responsive design, where an image behaves differently with changing device-type and display conditions

```
<picture>
      <source media="(min-width: 601px)" srcset="abc-large.jpeg">
      <source media="(max-width: 600px)" srcset="abc-small.jpeg">
      <img src="abc.jpeg">
</picture>
```

Image map

- An image map allows geometric areas on an image to be associated with hyperlinks. It may have multiple clickable areas, and each area may be associated with a specific hyperlink.
- The <map> along with <area> element is used to define a clickable image map.
- An image, in the form of an <img> element, may be associated with an image map by specifying a usemap attribute on the <img> element. The usemap attribute, if specified, must be a valid hash-name reference to a <map> element.

```
<img src="abc.png" usemap="#referenceID">
<map name="referenceID">
    <area shape="rect"   coords="0,0,100,100"    href="page1.html"      target="_blank">
    <area shape="circle" coords="200,150,50"     href="page1.html#section2">
</map>
```

The shape attribute is an enumerated attribute with the following keywords:

Keyword	Brief description
circle	Designates a circle, using exactly three integers in the coords attribute.
default	The whole image.
poly	Designates a polygon, using at-least six integers in the coords attribute.
rect	Designates a rectangle, using exactly four integers in the coords attribute.

Canvas

- `<canvas>` is used to draw 2D graphics on a web page with the help of JavaScript.

```
<canvas">An alternative fallback text</canvas>
```

- By default, a canvas has no border and no content.

```
<canvas id="myCanvas" width="300" height="150" style="border:1px solid #009A00;">
An alternative fallback text
</canvas>
```

- Inside `<script>`, add following lines to draw a rectangle

```
<script> const canvas = document.querySelector("canvas");
         const cnvs = canvas.getContext("2d");
         cnvs.fillStyle = "green";
         cnvs.fillRect(10, 10, 280, 130);
```

- Append bellow lines to draw a triangle

```
         const canvas = document.querySelector("canvas");
         const cnvs = canvas.getContext("2d");
         cnvs.fillStyle = "green";
         cnvs.fillRect(10, 10, 280, 130); </script>
```

SVG : Scalable Vector Graphics

- A W3C XML dialect to markup graphics. It defines markup and APIs for creating static or dynamic images, capable of interactivity and animation, including graphical effects. SVG graphics are scalable, and do not lose any quality if they are zoomed or resized
- `<svg>` behaves as container for scalable vector graphics. This block level element falls into the embedded, phrasing, flow, and palpable content categories for the purposes of the content models.

```
<svg width="200" height="100">
    <rect width="100%" height="100%" fill="lightblue"/>
    <circle cx="100" cy="50" r="48" fill="orange" stroke="green" stroke-width="1"/>
    <text x="50%" y="50%" text-anchor="middle" dy=".3em" fill="white">SVG Example</text>
</svg>
```

- `<svg>` vs `<canvas>`
  - `<svg>` renders graphics using XML and retains scalability and quality. `<canvas>` renders using JavaScript and is pixel-based
  - `<svg>` elements are part of the DOM, whereas canvas graphics are not.
  - `<svg>` is declarative (describes what to draw) and is better for static or interactive graphical elements that need to scale. `<canvas>` is imperative (describes how to draw) and is better for dynamic and complex graphics, animations, and real-time rendering.

Embeding Audio

- The `<audio>` element is used to embed sound or audio stream

```
<audio src="/path/abc.mp3" controls></audio>
```

Embeding Audio

- The `<audio>` element is used to embed sound or audio stream

```
<audio src="/path/abc.mp3" controls></audio>
```

- Attributes to control audio
  - `src` : Address of the resource
  - `controls` : Show user agent controls
  - `crossorigin` : How the element handles crossorigin requests
  - `preload` : An enumerated attribute that hints how much buffering the media resource will likely need. Values are: auto, none, metadata
  - `autoplay` : Hint that the media resource can be started automatically when the page is loaded
  - `loop` : Whether to loop the media resource
  - `muted` : Whether to mute the media resource by default

Embeding Audio

- The `<audio>` element is used to embed sound or audio stream

```
<audio src="/path/abc.mp3" controls></audio>
```

- Attributes to control audio
  - `src` : Address of the resource
  - `controls` : Show user agent controls
  - `crossorigin` : How the element handles crossorigin requests
  - `preload` : An enumerated attribute that hints how much buffering the media resource will likely need. Values are: auto, none, metadata
  - `autoplay` : Hint that the media resource can be started automatically when the page is loaded
  - `loop` : Whether to loop the media resource
  - `muted` : Whether to mute the media resource by default
- Industry practice: To ensure compatibility across browsers, provide multiple source files within the `<audio>` element along with fallback message. The browser will use the first format it supports.

```
<audio controls autoplay loop muted>
  <source src="abc.mp3" type="audio/mpeg">
  <source src="abc.ogg" type="audio/ogg">
  Your browser does not support the audio element <!--fallback message-->
</audio>
```

Embeding Video

- The `<video>` element is used to embed videos or movies, and audio files with captions.

```
<video controls src="/path/abc.mp4" poster="abc.jpg"  width="600"></video>
```

Embeding Video

- The `<video>` element is used to embed videos or movies, and audio files with captions.

```
<video controls src="/path/abc.mp4" poster="abc.jpg"  width="600"></video>
```

- Captions can be provided using the `<track>` element

```
<video controls autoplay muted>
    <source src="/path/abc.mp4" type="video/mp4">
    <track src="caption.vtt" kind="subtitles" srclang="hi" label="Hindi">
```

- The caption file (captions.vtt in the above example) should be in WebVTT format. WebVTT (Web Video Text Tracks) is a format used for displaying timed text tracks. It is designed to provide subtitles, captions, descriptions, chapters, or metadata that are time-aligned with the media content.

```
WEBVTT
00:00:00.000 --> 00:00:05.000
Welcome to our video.
```

Embeding Video

- The `<video>` element is used to embed videos or movies, and audio files with captions.

```
<video controls src="/path/abc.mp4" poster="abc.jpg"  width="600"></video>
```

- Captions can be provided using the `<track>` element

```
<video controls autoplay muted>
    <source src="/path/abc.mp4" type="video/mp4">
    <track src="caption.vtt" kind="subtitles" srclang="hi" label="Hindi">
```

- The caption file (captions.vtt in the above example) should be in WebVTT format. WebVTT (Web Video Text Tracks) is a format used for displaying timed text tracks. It is designed to provide subtitles, captions, descriptions, chapters, or metadata that are time-aligned with the media content.

```
WEBVTT
00:00:00.000 --> 00:00:05.000
Welcome to our video.
```

- Apart from the media controlling elements associated with audio, the video specific attributes are:
  - `poster` : Displays representative image of the video while no video data is available.
  - `playsinline` : Indicates that the video ought to be displayed 'inline' in the document
  - `width` : width of the video's display area, in pixel (not in %)

Multimedia elements

- Both audio and video elements can be used for both audio and video. The main difference between the two is simply that the audio element has no playback area for visual content (such as video or captions), whereas the video element does.

Multimedia elements

- Both audio and video elements can be used for both audio and video. The main difference between the two is simply that the audio element has no playback area for visual content (such as video or captions), whereas the video element does.

```
<video controls width="600">
  <source src="/path/abc.mp4" type="video/mpeg" />
  <source src="/path/abc.webm" type="video/webm" />
  Your browser does not support the video element <!--fallback message-->
  Instead, here is a
<a href="/path/abc.mp4" download="/path/abc.mp4">link</a> to download the video
<\video>
```

Table

- `<table>` element is used to present tabular data in the form of rows and columns
- To define a row of cells in a table, the `<tr>` element is used, while `<th>` and `<td>` element is used to define cell data in table header and table body part.

Course name	Course code	Credit	Semester
Web Technology	CS-2015	3	III
Signal and Systems	EC-2014	3	III
Mathematics-III	MA-2014	4	III

```
<style>
td, th {border: 1px solid gray;}
</style>
```

```
<table>
  <tr>
    <th>Course name</th>
    <th>Course code</th>
    <th>Credit</th>
    <th>Semester</th>
  </tr>
  <tr>
    <td>Web Technology</td>
    <td>CS-2015</td>
    <td>3</td>
    <td>III</td>
  </tr>
```

```
  <tr>
    <td>Signal and Systems</t
    <td>EC-2014</td>
    <td>3</td>
    <td>III</td>
  </tr>

  <tr>
    <td>Mathematics-III</td>
    <td>MA-2014</td>
    <td>4</td>
    <td>III</td>
  </tr>
</table>
```

Semantic sections of table

- `<thead>`, `<tbody>`, and `<tfoot>` elements are used to structure a table into semantic sections
  - The `<thead>` element encapsulates a set of table rows (`<tr>` elements), indicating that they comprise the head of a table with information about the table's columns `<thead>` is placed after any `<caption>` and `<colgroup>` elements, but before any `<tbody>`, `<tfoot>`, and `<tr>` elements.
  - `<th>` element defines a cell as the header of a group of table cells and may be used as a child of the `<tr>` element
  - `<tbody>` element encapsulates a set of table rows, indicating that they comprise the body of a table's data.
  - A `<table>` may have multiple `<tbody>` but only one `<thead>`
  - `<tfoot>` element encapsulates a set of table rows, indicating that they comprise the foot of a table with information about the table's columns. This is usually a summary of the columns

Caption of a table

- `<caption>` element specifies the caption of a table, providing the table an accessible description. The `<caption>` must be the first child of its parent `<table>` element.

```
<table>
  <caption>Description of the table</caption>
  <tr>  ..... </tr>
</table>
```

  - An accessible description is a text description associated with an HTML element that provides users of assistive technology with a description for the element beyond what is provided by the element's accessible name.
  - The accessible description of an element is part of the accessibility tree that makes web content available to screen readers and other assistive technologies, which, in turn, make that content available to the users of those technologies
  - The accessible description for a
    - `<table>` is its `<caption>`
    - `<figure>` is its `<figcaption>`
    - `<input>` is the `value attribute's value`, unless the element also has a `aria-describedby` or `aria-description` attribute
    - `title` attribute for other elements

Assistive technology

- Assistive technology refers to a third party application which augments or replaces the existing UI for an application. For example: `screen reader`, which replaces the visual and pointer-based UI with an auditory output
- An accessibility tree is a hierarchical structure generated from the HTML of a webpage or the UI components of an application. It ensures that people who use assistive technologies can access and interact with web content effectively. It includes details about the structure, properties, and relationships of the elements on the page, such as text content, form controls, images, and interactive elements.

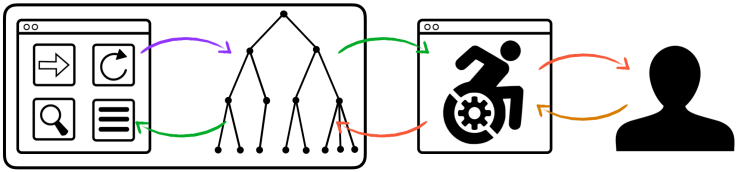


Image source: https://wicg.github.io/aom

Accessibility tree

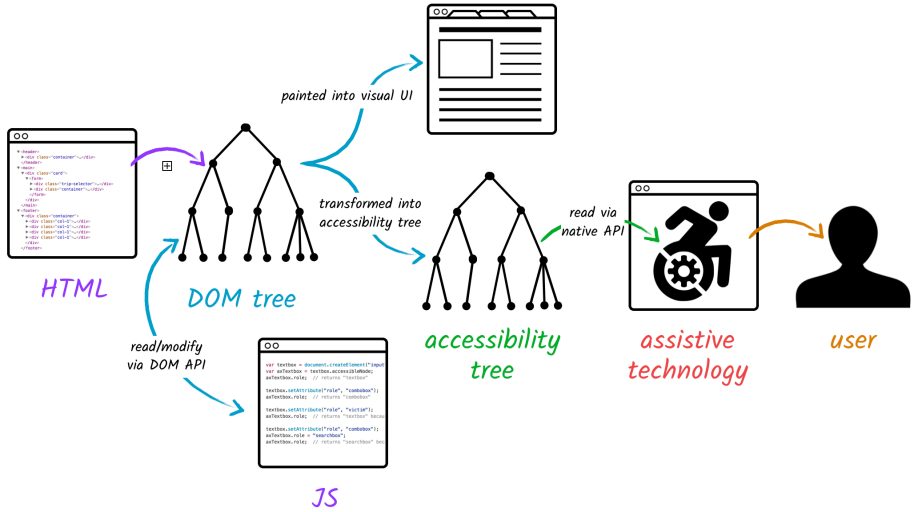


Image source: https://wicg.github.io/aom