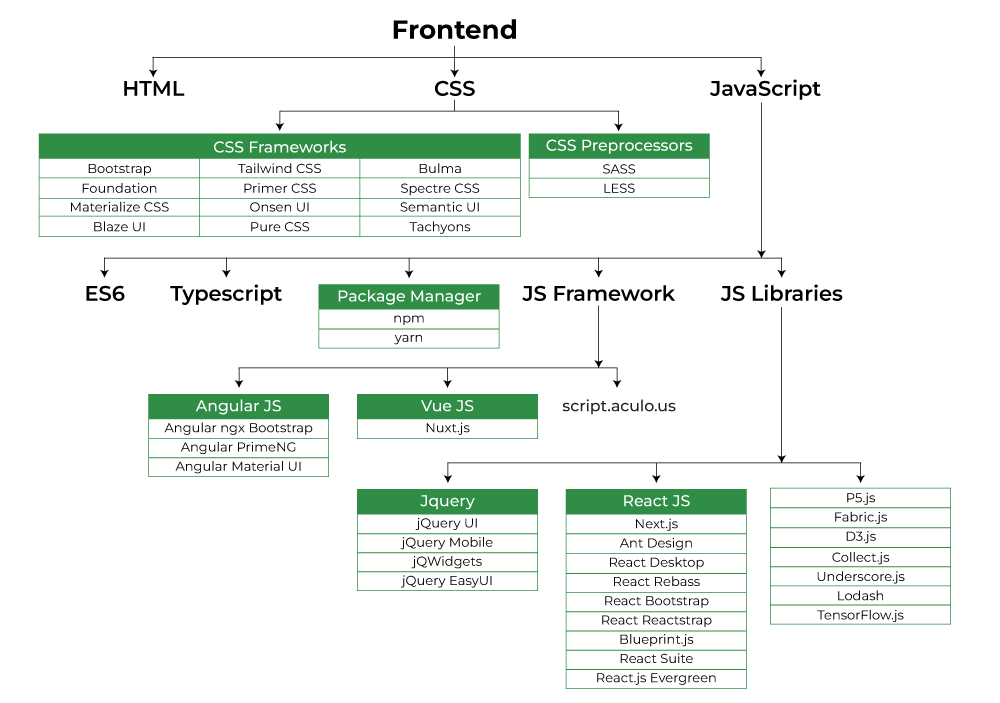
**HTML Syllabus content**

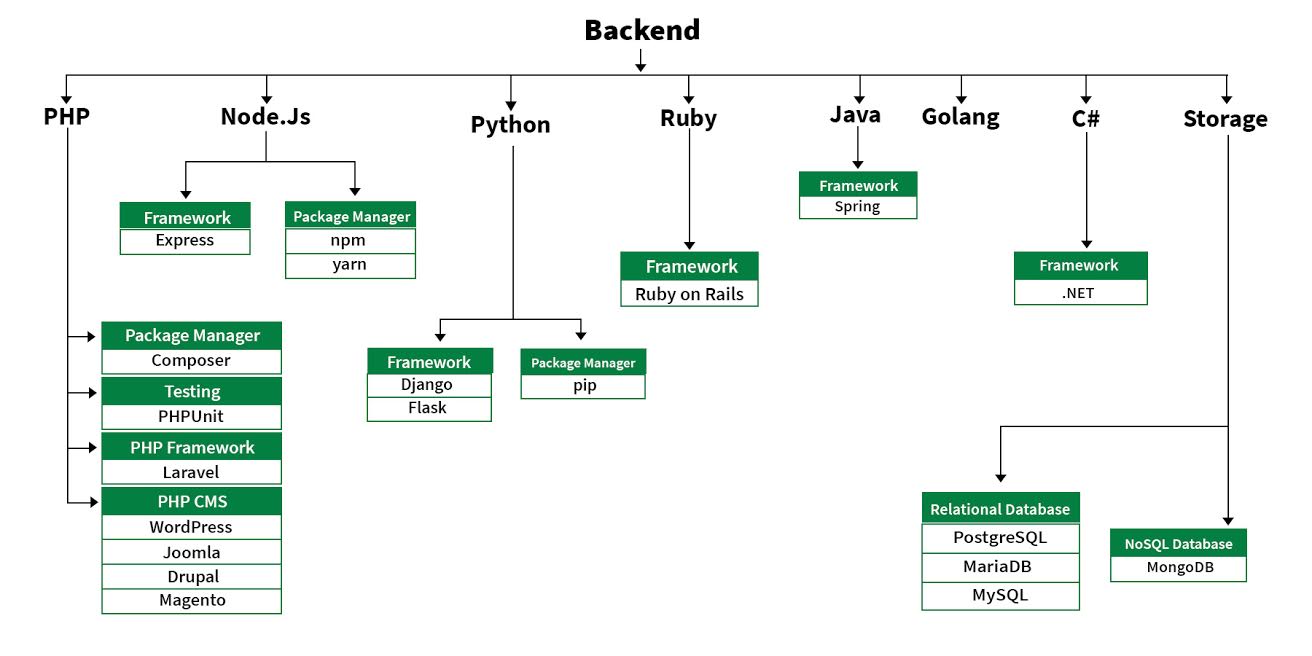
**Introduction of different Web Technology**

Web Development can be classified into two ways:

* **Frontend Development**
* **Backend Development**
* [**Frontend Development:**](https://www.geeksforgeeks.org/how-to-become-a-front-end-developer/) The part of a website that the user interacts directly is termed as front end. It is also referred to as the ‘client side’ of the application.



[**Backend Development:**](https://www.geeksforgeeks.org/what-is-the-difference-between-front-end-and-back-end-web-development/) Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

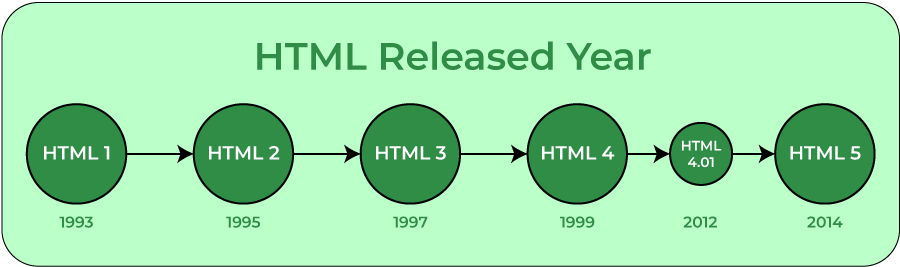


**HTML Introduction**

**HTML** stands for HyperText Markup Language.

**Hypertext** defines the link between web pages.   
A **markup language** is used to define the text document within the tag which defines the structure of web pages.

HTML was designed by the British scientist Sir **Tim Berners-Lee** at the CERN nuclear physics laboratory **in Switzerland** during the 1980s.



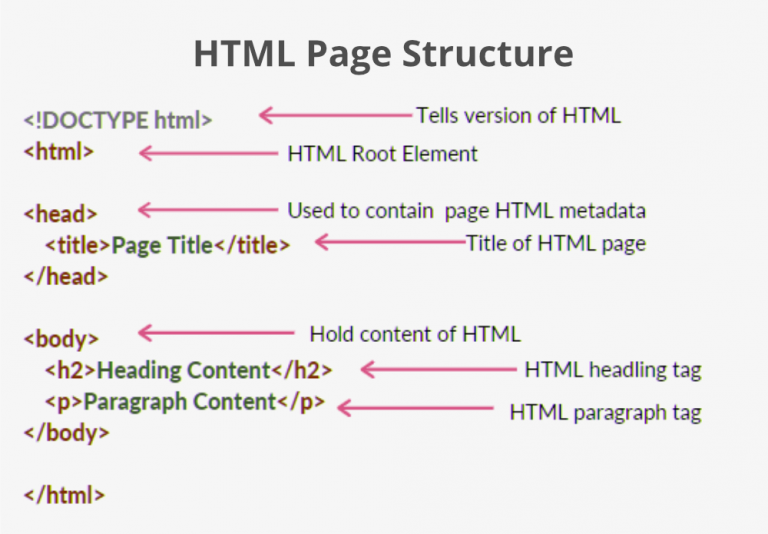
This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly.

The language uses tags to define what manipulation has to be done on the text.

**Giving them different meaning** in a document like (**Is it a paragraph? Is it a bulleted list? Is it part of a table?**), structure a document into **logical sections** (**Does it have a header? Three columns of content? A navigation menu?**), and embed content such as images and videos into a page. This module will introduce the first two of these and introduce fundamental concepts and syntax you need to know to understand HTML.

**HTML Elements**

**HTML page structure:**The basic structure of an HTML page contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.



[**<!DOCTYPE html>**](https://www.geeksforgeeks.org/html-doctypes/)**:** This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive.

[**<html>**](https://www.geeksforgeeks.org/html-html-tag/)**:** This is called the HTML root element. All other elements are contained within it.

[**<head>**](https://www.geeksforgeeks.org/html-head-tag/#:~:text=The%20tag%20in%20HTML,head%3E%20element%20can%20be%20omitted.)**:** The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage.

[**<title>**](https://www.geeksforgeeks.org/html-title-tag/)-The title is what is displayed on the top of your browser when you visit a website and contains the title of the webpage that you are viewing.

[**<body>**](https://www.geeksforgeeks.org/html-body-tag/#:~:text=The%20tag%20in%20HTML,well%20as%20an%20ending%20tag.)**:** The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end.

An HTML document can be created using any text editor. Save the text file using **.html** or **.htm**. Once saved as an HTML document, the file can be opened as a webpage in the browser.

**HTML Attributes**

Elements in HTML have **attributes**; these are additional values that configure the elements

Reference: <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>

**Syntax**

<element **attribute\_name="attribute\_value"**>

|  |
| --- |
| C:\Users\KARTHIK\Documents\Frontend class\Syllabus covered\images\attribute.png |

**Multiple attributes are separated a by space**

|  |
| --- |
| C:\Users\KARTHIK\Documents\Frontend class\Syllabus covered\images\attributes.png |

**HTML Headings**

An HTML heading tag is used to define the headings of a page. There are six levels of headings defined by HTML. These 6 heading elements are h1, h2, h3, h4, h5, and h6

* **<h1>** is used for the main heading. (Biggest in size)
* **<h2>** is used for subheadings, if there are further sections under the subheadings then the**<h3>** elements are used.
* **<h6>** for the small heading (smallest one).

**<p> Paragraph Element**

The **<p>** [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) element represents a paragraph.

HTML paragraphs can be any structural grouping of related content, such as images or form fields.

Paragraphs are [block-level elements](https://developer.mozilla.org/en-US/docs/Glossary/Block-level_content), and notably will automatically close if another block-level element is parsed before the closing </p> tag. See "Tag omission" below.

**Eg:  
<p>**Separating paragraphs with blank lines is easiest for readers to scan, but they can also be separated by indenting their first lines. This is often used to take up less space, such as to save paper in print**.</p>**

**<p>**Writing that is intended to be edited, such as school papers and rough drafts, uses both blank lines and indentation for separation. In finished works, combining both is considered redundant and amateurish**.</p>**

**<p>**How hard to read? See for yourself: <button data-toggle-text="Oh no! Switch back!">Use pilcrow for paragraphs</button> **</p>**

**HTML Formating**

Reference: <https://www.w3schools.com/html/html_formatting.asp>

|  |  |
| --- | --- |
| The HTML <b> element defines bold text, without any extra importance. | |
| <p><b>This text is bold</b></p> | **This text is bold** |
| The HTML <strong> element defines text with strong importance. The content inside is typically displayed in bold. | |
| <p><strong>This text is important!</strong></p> | **This text is important!** |
| The HTML <i> element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.  **Tip:** The <i> tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc. | |
| <p><i>This text is italic</i></p> | *This text is italic* |
| The HTML <em> element defines emphasized text. The content inside is typically displayed in italic.  **Tip:** A screen reader will pronounce the words in <em> with an emphasis, using verbal stress. | |
| <p><em>This text is emphasized.</em></p> | This text is emphasized. |
| <p>This is<sub> subscript</sub> and <sup>superscript</sup></p> | This issubscript and superscrip |
| The HTML <small> element defines smaller text: | |
| <p>This is some normal text.</p>  <p><small>This is some smaller text.</small></p> | This is some normal text.  This is some smaller text. |
| The HTML <mark> element defines text that should be marked or highlighted: | |
| <p>Do not forget to buy <mark>milk</mark> today.</p> | Do not forget to buy milk today. |
| The HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text: | |
| <p>My favorite color is <del>blue</del> red.</p> | My favorite color is  red. |
| The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text: | |
| <p>My favorite color is <del>blue</del> <ins>red</ins>.</p> | My favorite color is  red. |
| The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H2O: | |
| <p>This is <sub>subscripted</sub> text.</p> | This is subscripted text. |
| The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1]: | |
| <p>This is <sup>superscripted</sup> text.</p> | This is superscripted text. |

**HTML Fonts** – Deprecated in HTML5 (Use CSS instead)

**HTML Styles**

## Three Ways to Insert CSS:

* HTML **style attribute** (Inline CSS)
* HTML **style element** (Internal CSS)
* External CSS *(Best)*

HTML **style attribute**

The HTML style attribute is used to add styles to an element, such as color, font, size, and more.

This way of adding the style is called **Inline CSS**

**Syntax:**

|  |
| --- |
| <tagname style="property:value;"> |

**Eg:**

|  |  |
| --- | --- |
| HTML | Output |
| <p>I am normal</p>  <p **style="color:red;"**>I am red</p>  <p **style="color:blue;"**>I am blue</p>  <p **style="font-size:50px;"**>I am big</p> | I am normal  I am red  I am blue  I am big |

HTML **style element**

The **<style>** [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) element contains CSS, which is applied to the contents of that particular document.

The <style> element must be included inside the [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head) of the document.

**Syntax:**

|  |
| --- |
| <style>  </style> |

**Eg:**

|  |  |
| --- | --- |
| HTML | Output |
| **<style>**  **h1 {**  **color: red;**  **}**  **p {**  **color: #26b72b;**  **}**  **</style>**  <h1>Main Heading</h1>  <p>This text will be green. Inline styles take precedence over CSS included externally.</p> | Main Heading This text will be green. Inline styles take precedence over CSS included externally. |

In general, it is better to put your styles in external stylesheets and apply them using [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link) elements.

If you include multiple <style> and <link> elements in your document, they will be applied to the DOM in the order they are included in the document — make sure you include them in the correct order, to avoid unexpected cascade issues.

External CSS

External styles are defined within the <link> element, inside the <head> section of an HTML page:

1. Create a css file and save the file by giving a .css file type extension.   
   Eg: **style.css**
2. Then include the css file using the <link> tag, inside your HTML document, inside the <head> tag.  
     
   Example:

|  |
| --- |
| <!DOCTYPE html> <html> <head>  **<link rel="stylesheet" href="style.css">** </head> <body>   <h1>This is a heading</h1>  <p>This is a paragraph.</p>  </body> </html> |

Syntax:

|  |
| --- |
| <link rel="stylesheet" href="fileName.css"> |

**HTML Links**

* **HTML <link> tag** - (used to link external stylesheets and favicon)
* **HTML <a> anchor link tag** – (used to create an hyper text link)
* **HTML <script> tag** (used for including external Javascript)

## HTML Link tag Definition and Usage

The <link> tag defines the relationship between the current document and an external resource.

The <link> tag is most often used to link to external style sheets or to add a [favicon](https://www.w3schools.com/html/html_favicon.asp) to your website.

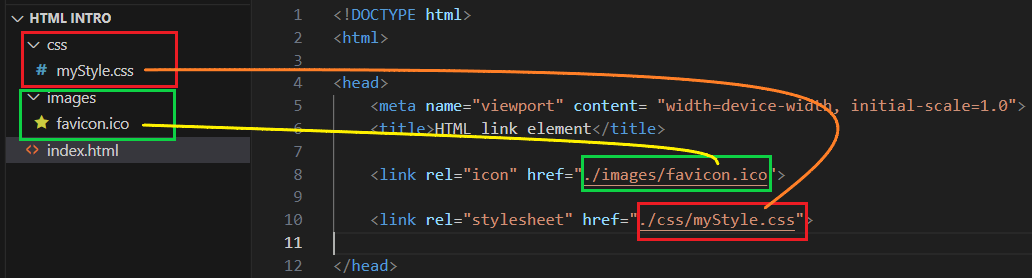
The <link> element is an empty element, it contains attributes only.

Link element attribute reference:   
<https://www.w3schools.com/tags/tag_link.asp>

**Example:**

|  |  |
| --- | --- |
| Element | Output |
| <link rel="icon" href="./images/favicon.ico"> | C:\Users\KARTHIK\Documents\Frontend class\html tasks\favicon_screenshot.PNG |
| <link rel="stylesheet" href="./css/myStyle.css"> |  |

Screenshot



## HTML anchor tag (used for Hypertext / Hyperlink)

The **<a>** [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) element (or anchor element), with [its href attribute](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a#href), creates **a hyperlink to web pages**, **files**, **email addresses, locations in the same page**, or **anything else a URL can address**.

If the **href** attribute is present, pressing the enter key while focused on the <a> element will activate it.

Example:

|  |  |
| --- | --- |
| Element | Output |
| <p>You can reach me at:</p>  <ul>  <li>**<a href="https://example.com">Website</a>**<**/**li>  <li>**<a href="mailto:karthi@example.com">Email</a></**li>  <li>**<a href="tel:+918428882222">Phone</a></**li>  </ul> | You can reach me at:   * [Website](https://example.com/) * [Email](mailto:karthi@example.com) * [Phone](tel:+918428882222) |

Attributes in <a> tag

Reference for global attributes: <https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes>

Reference for <a> local attributes: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a>

|  |  |
| --- | --- |
| <a> Attribute name | Description |
| download | Causes the browser to treat the linked URL as a download.  **Note:** download only works for same-origin URLs, or the blob: and data: schemes. |
| href |  |
| hreflang |  |

**HTML Script**

The **<script>** [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) element is used to embed executable code or data; this is typically used to embed or refer to JavaScript code. The <script> element can also be used with other languages, such as [WebGL](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API)'s GLSL shader programming language and [JSON](https://developer.mozilla.org/en-US/docs/Glossary/JSON).

Reference:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script>

# How to import External Script using the <script> element.

|  |
| --- |
| <script src="javascript.js"></script> |

If you have to use a single script functionality among many HTML pages then it is a good idea to keep that function in a single script file and then include this file in all the HTML pages.

The HTML <noscript> Tag

The HTML <noscript> tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support scripts:

|  |
| --- |
| **JavaScript Example:** |
| <script **type="text/javascript"**>  document.getElementById("demo").innerHTML = "Hello JavaScript!"; </script>  <noscript>Sorry, your browser does not support JavaScript!</noscript> |
| **VBScript Example:** |
| <script **type="text/vbscript">**  document.getElementById("demo").innerHTML = "Hello JavaScript!"; </script>  <noscript>Sorry, your browser does not support VBScript!</noscript> |

Default Scripting Language

You can specify a default scripting language for all your *script* tags to use. This saves you from having to specify the language everytime you use a script tag within the page.

Example:

|  |
| --- |
| <meta http-equiv="Content-Script-Type" content="text/JavaScript" /> |

Note: You can still override the default by specifying a language within the script tag.

Importing modules with importmap

|  |
| --- |
| Script.js |
| **import** { name as squareName, draw } from "./shapes/square.js";  **import** { name as circleName } from "https://example.com/shapes/circle.js"; |

|  |
| --- |
| Index.html |
| <script type="importmap">  {  "imports": {  "square": "./shapes/square.js",  "circle": "https://example.com/shapes/circle.js"  }  }  </script> |

# How to import Internal Script using the <script> element.

|  |
| --- |
| <head>  <title>Internal Script</title>  </head>  <body>  **<script type="text/javascript">**  **document.write("Hello Javascript!")**  **</script>**  </body> |

You can write your script code directly into your HTML document. Usually we keep script code in header of the document using <script> tag, otherwise there is no restriction and you can put your source code anywhere in the document.

* 1. Inline JavaScript

|  |
| --- |
| <button type = "button" **onclick = "(function(){**  alert();  prompt();  **})()"**> Click Me </button> |

**HTML Media**

HTML media Attribute

Used for specific views such as print view, mobile views, tab views.

## Applies to

The media attribute can be used on the following elements:

|  |  |
| --- | --- |
| **Elements** | **Attribute** |
| [<a>](https://www.w3schools.com/tags/tag_a.asp) | [media](https://www.w3schools.com/tags/att_a_media.asp) |
| [<area>](https://www.w3schools.com/tags/tag_area.asp) | [media](https://www.w3schools.com/tags/att_area_media.asp) |
| [<link>](https://www.w3schools.com/tags/tag_link.asp) | [media](https://www.w3schools.com/tags/att_link_media.asp) |
| [<source>](https://www.w3schools.com/tags/tag_source.asp) | [media](https://www.w3schools.com/tags/att_source_media.asp) |
| [<style>](https://www.w3schools.com/tags/tag_style.asp) | [media](https://www.w3schools.com/tags/att_style_media.asp) |

Example

This CSS works only for print view

|  |
| --- |
| <head> <link rel="stylesheet" type="text/css" href="theme.css"> <link rel="stylesheet" type="text/css" href="print.css" **media="print"**> </head> |

This video file works only for screen size starting from 320px

|  |
| --- |
| <source src="movie.ogg" type="video/ogg" **media="screen and (min-width:320px)"**> |

The first image works only for screen size starting from min-width 650px

The second image works only for screen size starting from min-width 465px

|  |
| --- |
| <picture>  <source **media="(min-width: 650px)"** srcset="img\_pink\_flowers.jpg">  <source **media="(min-width: 465px)"** srcset="img\_white\_flower.jpg">  <img src="img\_orange\_flowers.jpg" alt="Flowers" style="width:auto;">  </picture> |

Reference:

<https://www.w3schools.com/tags/att_media.asp>

**HTML Multimedia**

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Reference: <https://www.w3schools.com/html/html_media.asp>

**HTML Audio**

## The HTML <audio> Element

The HTML <audio> element is used to play an audio file on a web page.

Example:

|  |
| --- |
| <audio controls>   <source src="horse.ogg" type="audio/ogg">   <source src="horse.mp3" type="audio/mpeg"> Your browser does not support the audio element. </audio> |

* The controls attribute adds audio controls, like play, pause, and volume.
* The <source> element allows you to specify alternative audio files which the browser may choose from.
* The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.
* Autoplay: To start an audio file automatically, use the autoplay attribute
* Add muted after autoplay to let your audio file start playing automatically (but muted)

Reference: <https://www.w3schools.com/html/html5_audio.asp>

**HTML Object**

The <object> tag defines a container for an external resource.

The external resource can be a web page, a picture, a media player, or a plug-in application.

Example:

|  |
| --- |
| An embedded image: |
| <object data="pic\_trulli.jpg" width="300" height="200"></object> |
| An embedded HTML page: |
| <object data="snippet.html" width="500" height="200"></object> |
| An embedded video: |
| <object data="video.mp4" width="400" height="300"></object> |

Reference: <https://www.w3schools.com/tags/tag_object.asp>

Note:

To embed a picture, it is better to use the <img> tag.

To embed HTML, it is better to use the <iframe> tag.

To embed video or audio, it is better to use the <video> and <audio> tags.

**HTML Video**

The HTML <video> element is used to show a video on a web page.

Example:

|  |
| --- |
| <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> |

**Youtube Video**

To play your video on a web page, do the following:

* Upload the video to YouTube
* Share -> Embed
* Copy the <iframe> code
* Paste it in your html page
* Use the width and height attributes to specify the dimension of the player

Example

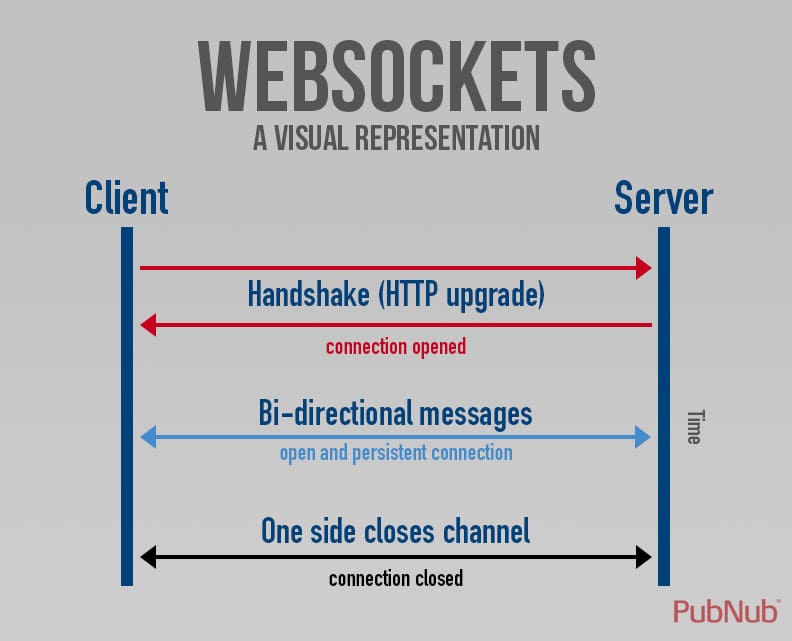
|  |
| --- |
| <iframe width="420" height="315" src="https://www.youtube.com/embed/tgbNymZ7vqY"> </iframe> |

**History, Vision & Future of HTML5**

**What is HTML5**

HTML 5 is the fifth and current version of HTML. It has improved the markup available for documents and has introduced application programming interfaces (API) and Document Object Model (DOM).

There are **three organizations** that are currently **in charge of the specification of HTML5**:

1. **Web Hypertext Application Technology Working Group** **(WHATWG)** created the HTML5 specification and is in charge of the HTML5 development that provides open collaboration of browser vendors and other involved parties.  
     
   Resource: <https://html.spec.whatwg.org/>
2. **World Wide Web Consortium (W3C)** is in charge with delivering the HTML5 specification.  
     
   Resource: <https://dev.w3.org/html5/spec-LC/>
3. **Internet Engineering Task Force (IETF)** is in charge of the development of HTML5 WebSocket API. (an advanced technology that makes it possible to open a two-way interactive communication session between the user's browser and a server)  
     
   Resource: <https://datatracker.ietf.org/doc/html/rfc6455>   
     
   

HTML5 is simply just an umbrella term for the next generation of web apps an how functionality will be expanded with better markup (HTML), better style (CSS), and better interactivity (JavaScript).

## New Features:

* **New Semantic Elements** − These are like <header>, <footer>, and <section>.
* **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
* **Persistent Web Storage** − To achieve without resorting to third-party plugins.
* **WebSocket** − A next-generation bidirectional communication technology for web applications.
* **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
* **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
* **Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.
* **Geolocation** − Now visitors can choose to share their physical location with your web application.
* **Microdata** − This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.  
  Tutorial Reference: <https://www.tutorialspoint.com/html5/html5_microdata.htm>
* **Drag and drop** − Drag and drop the items from one location to another location on the same webpage.

**A Little Retrospective**

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

**Vision And Philosophy Behind HTML5**

HTML5 was designed with major objectives, including:

* Making code easier to read for users and screen readers
* Reducing the overlap between HTML, CSS, and JavaScript
* Promoting design responsiveness and consistency across browsers
* Supporting multimedia without the need for Flash or other plugins

**Compatibility**

HTML5 is now compatible with all popular browsers (Chrome, Firefox, Safari, IE9, and Opera) and with the introduction of DOCTYPE, it is even possible to have a few HTML features in older versions of Internet Explorer too.

**Some Compatibility check reference:**Browser compatibility check tool: <https://caniuse.com/>   
Website test tool for HTML 5 standard: <https://html5test.com/>

**Utilities**

CSS Utilities:  
Utilities are simple HTML classes typically scoped to a single CSS property, like border-style or background-color . Utilities can be used additively to style an object from scratch or to override a style defined in component CSS.

HTML Utilities:

Reference: <http://www.a-test.fr/Web_Technologies/HTML5/html5_tags_utility.php>

**Web interoperability**

**The ability of different systems, devices, applications or products to connect and communicate in a coordinated way, without effort from the end user.** Functions of interoperable components include data access, data transmission and cross-organizational collaboration regardless of its developer or origin.  Similar to [compatibility](https://www.techtarget.com/whatis/definition/compatibility), interoperability helps organizations achieve higher efficiency and a more [holistic](https://www.techtarget.com/searchcio/definition/holistic-holistic-technology) view of information.

**Universal Access**

The practice of ensuring that your websites and web applications are **accessible by people with disabilities of many kinds.**

**Future of HTML5**

HTML5 is the future of web development. It provides a more powerful and flexible way to develop web applications and has empowered business owners and entrepreneurs to provide better customer experiences.

Reference: <https://www.namecheap.com/blog/5-reasons-why-html5-is-the-future/#:~:text=Into%20the%20future,to%20provide%20better%20customer%20experiences>.

**Timeline**

The combined timelines for the W3C recommendations of HTML5, HTML 5.1, HTML 5.2 and HTML 5.3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **First draft** | **Candidate recommendation** | **Recommendation** | **Retired** |
| HTML5 | 1 May 2007[[41]](https://en.wikipedia.org/wiki/HTML5#cite_note-41) | 17 December 2012 | 28 October 2014 | 27 March 2018[[42]](https://en.wikipedia.org/wiki/HTML5#cite_note-42) |
| HTML 5.1 | 17 December 2012 | 21 June 2016 | 1 November 2016 | 28 January 2021[[43]](https://en.wikipedia.org/wiki/HTML5#cite_note-43) |
| HTML 5.1 2nd Edition | — | 20 June 2017 | 3 October 2017 |
| HTML 5.2 | 18 August 2016 | 8 August 2017 | 14 December 2017 | 28 January 2021[[44]](https://en.wikipedia.org/wiki/HTML5#cite_note-44) |
| HTML 5.3 | 14 December 2017[[45]](https://en.wikipedia.org/wiki/HTML5#cite_note-5.3FWD-45) | — | — | 28 January 2021[[46]](https://en.wikipedia.org/wiki/HTML5#cite_note-46) |

**Gracefull Degradation**

Polyfill

A polyfill is a piece of code (usually JavaScript on the Web) used to provide modern functionality on older browsers that do not natively support it.

// before running the transpiler

**height = height ?? 100;**

// after running the transpiler

**height = (height !== undefined && height !== null) ? height : 100;  
  
Reference:** <https://blog.logrocket.com/use-polyfills-react-app/>

Handling common HTML & CSS problems

**Reference:** <https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Cross_browser_testing/HTML_and_CSS>

**Browser support table for css :has() pseudo class:** <https://developer.mozilla.org/en-US/docs/Web/CSS/:has#browser_compatibility>

**Developer Tools**

Every modern web browser includes a powerful suite of developer tools. These tools do a range of things, from inspecting currently-loaded HTML, CSS and JavaScript to showing which assets the page has requested and how long they took to load.

Reference: <https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Tools_and_setup/What_are_browser_developer_tools>



**Page Encoding**

HTML encoding ensures that text will be correctly displayed in the browser, not interpreted by the browser as HTML. For example, if a text string contains a less than sign (<) or greater than sign (>), the browser would interpret these characters as an opening or closing bracket of an HTML tag.

The default **character encoding** in HTML5 is **UTF - 8**

Reference: <https://www.w3schools.com/charsets/>

**HTML 5 Semantic elements**

A semantic element clearly describes its meaning to both the browser and the developer.

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

Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<article>](https://www.w3schools.com/tags/tag_article.asp) | Defines independent, self-contained content |
| [<aside>](https://www.w3schools.com/tags/tag_aside.asp) | Defines content aside from the page content |
| [<details>](https://www.w3schools.com/tags/tag_details.asp) | Defines additional details that the user can view or hide |
| [<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp) | Defines a caption for a <figure> element |
| [<figure>](https://www.w3schools.com/tags/tag_figure.asp) | Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. |
| [<footer>](https://www.w3schools.com/tags/tag_footer.asp) | Defines a footer for a document or section |
| [<header>](https://www.w3schools.com/tags/tag_header.asp) | Specifies a header for a document or section |
| [<main>](https://www.w3schools.com/tags/tag_main.asp) | Specifies the main content of a document |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |
| [<nav>](https://www.w3schools.com/tags/tag_nav.asp) | Defines navigation links |
| [<section>](https://www.w3schools.com/tags/tag_section.asp) | Defines a section in a document |
| [<summary>](https://www.w3schools.com/tags/tag_summary.asp) | Defines a visible heading for a <details> element |
| [<time>](https://www.w3schools.com/tags/tag_time.asp) | Defines a date/time |

**HTML5 New and updated elements**

**Reference:** <https://www.w3schools.blog/html5-new-elements>

**Reference:** <https://www.w3.org/wiki/HTML/New_HTML5_Elements>

**Reference:** <https://www.tutorialspoint.com/html5/html5_new_tags.htm>

**HTML5 Structural Elements**

There are a group of HTML elements that are created to broadly give our documents more structure.

**Reference:** <http://web.simmons.edu/~grovesd/comm244/notes/week3/structural-elements>

**New Attributes in HTML5**