

HTML

(Hyper Text Markup Language)

Introduction To HTML

- HTML enables documents to be displayed as **web pages** on the internet
- HTML elements
 - Building blocks of an HTML page
 - Represented by tags </>
 - Examples: Paragraphs, lists, tables, etc..
- HTML5 Overview
 - Defines a single language which can be written in HTML or XML syntax
 - Allows interoperability with earlier HTML versions
 - Improves markups for documents
 - HTML5 introduces new elements that provide clearer and more meaningful structure to web documents such as
 - <header>, <footer>, <nav>, <article>, <section>, and <aside>
 - By using these semantic tags, developers can create more organized and accessible web content, making it easier for both humans and search engines to understand the structure and purpose of different parts of a webpage.
 - Includes markups and APIs for web storage, video and audio content
 - HTML5 includes native support for embedding audio and video content directly into webpages without the need for third-party plugins. This is achieved through the <audio> and <video> elements, which allow developers to specify audio and video files using standard HTML attributes.
 - Additionally, HTML5 provides APIs for web storage, such as the localStorage and sessionStorage APIs.
 - These allow developers to store data locally in the user's browser.
 - These storage options provide a way for web applications to store user preferences, session data, and other information without relying on server-side storage mechanisms.

HTML Example

- Documents having **.txt / .html** extension are parsed using HTML rules
- Documents transmitted with HTML content type are processed as an HTML document

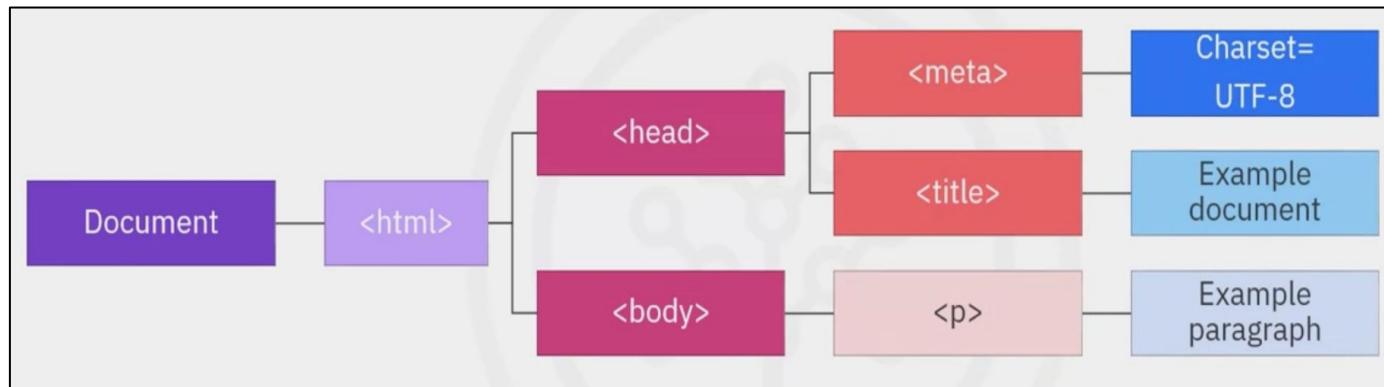
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sample Page</title>
  </head>
  <body>
    <p>Example paragraph</p>
    <!-- this is a comment -->
  </body>
</html>
```

- Declaration tag that represents the document type
- Instructs the web browser about the version of HTML
- Should be the first line of the HTML code

- The **<html>** tag is the root element of this tree.
 - It contains all of the other HTML elements, except the **<!DOCTYPE>** tag.
- The **<head>** element can contain the following tags:
 - Title (**<title>**)
 - Scripts (**<script>**)
 - Style (**<style>**)
 - Style sheet links (**<link>**)
 - Meta information (**<meta>**)
 - Browser support information and other initialization functions (**<base>**)
- The **<body>** tag contains all content that is displayed on the webpage.

HTML DOM Tree

- A DOM tree is an **in-memory representation of a document**.
 - Describes how the web page is structured.
- DOM trees contain several kinds of nodes that define the document type and structure such as
 - Headers and paragraphs
 - Text nodes
 - Comment nodes



XML Example

- Here is an example of a document that conforms to the XML (Extensible Markup Language) syntax of HTML5.
- XML documents look similar to HTML documents, except they have an **XML tag** on the first line.
- In addition, the **content-type** must be specified as an XML media type such as application or xml.

```
<?xml version="1.0" encoding="UTF-8"?> ←  
<html xmlns="https://www.w3.org/1999/xhtml">  
  <head>  
    <meta http-equiv="Content-Type"  
          content="application/xhtml+xml; charset=ISO-8859-1" /> ←  
    <title>Example document</title>  
  </head>  
  <body>  
    <p>Example paragraph</p>  
  </body>  
</html>
```

What To Choose : XHTML OR HTML ?

- XHTML must be well-formed.
 - Every element must have an end tag.
 - All attributes must have a value and double or single quotation marks must surround all attribute values.

XHTML	HTML
<ul style="list-style-type: none">• Tags need to be in lowercase• Codes must be well-formed• XML parser will stop processing if it encounters a situation where the syntax is not well-formed	<ul style="list-style-type: none">• Case used does not matter• Unmatched quotation marks, non-terminated and uncontained elements are allowed• Syntax is less rigorous than XHTML syntax

HTML5 Features

- **Improved search indexes with <meta> tag**
 - <meta> tags are used to provide information to search engines.
 - Search engine optimizations uses keywords from HTML attributes to improve the visibility of a website when search results are displayed.
- **Better page load time**
 - The efficient use of HTML and CSS, while reducing the number of images, can lead to faster load times for rendering web pages.
- **Enhanced user experience**
 - Supports number of APIs that enhance the user experience, such as the advanced animation, drawings, audio, and video elements.

HTML5 Elements

- Structural elements help to logically define the page structure
 - section, article, header, footer, figure, and figcaption
- APIs help with graphics and embedded content
 - canvas, audio, and video
- Input elements can automatically be validated by the browser
 - tel, email, datetime, number, range, and color
- Web storage APIs can store data in the browser
 - localStorage, sessionStorage
- Web workers run processing-intensive tasks without blocking the user interactions to the current page

HTML Scripting

- Scripting is done often through the use of JavaScript
 - Within your **HTML code**
 - Within the `<script>` tag
 - Within a **separate file** which is called in your HTML code
- Scripting provides a **more interactive user experience** when browsing websites.
- It can be used for various tasks, such as
 - Form validation
 - Dynamically changing the content of a website
 - Manipulating images.
- Since scripting can be turned off, the recommendation is to use scripting but not to rely on it.

Enabling Scripting

- Scripting is enabled for a browser context when the following conditions are true.
 - The browser supports scripting
 - The user has scripting enabled
 - The browser does not have sandboxed content flag set
 - Sandboxed: specifies an extra set of content restrictions and can be used when hosting untrusted content.
 - This sandboxed can be set at the page level or specified as an attribute for embedded objects.
 - Running a page that contains an embedded object without the sandbox attribute grants the embedded object the same permissions as the rest of your page. This implicitly gives a third-party vendor permission to run scripts with the same permissions that you have for that page, which is one way that you can inadvertently allow advertisements to occur in your application.
 - So, to prevent granting implicit permissions to embedded objects, use the sandbox attribute on any tag that contains an embedded object.

```
• <meta http-equiv="Content-Type" content="text/html-  
    sandboxed; ...  
• <iframe sandbox src=\"http://maps.google.com/?ll=" +  
    theLat + "," + theLong + "&z=16&output=embed\"></iframe>
```

HTML Document API - DOM Tree Accessors

- Each HTML document that is loaded into a browser page becomes a **Document object**.
 - The Document object provides access to all HTML elements in a page and can be accessed from within a script.
- The DOM tree accessors are HTML document APIs that provide access to all the HTML elements on a page.
- The property to access is prefixed by the word “document”
 - For example: `document.head`, `document.title`

PROPERTY	DESCRIPTION
head	Returns the <i>head</i> element
title	Sets or returns the <i>title</i> of the document
images	Returns an <code>HTMLCollection</code> of the <i>img</i> elements in the document
lastModified	Returns the date of the last modification to the document
scripts	Returns an <code>HTMLCollection</code> of the <i>script</i> elements in the document

PROPERTY	DESCRIPTION
getElementById('id')	Accesses the first element with the specific id
getElementsByName('tag')	Returns a nodelist of all elements with the specified HTML tag name
open()	Opens an output stream to collect the output from document.write()
write()	Writes JavaScript code to the document
close()	Closes the output stream previously opened with document.open()

HTML5 Browser Support



Some browsers do not fully support all HTML5 and CSS3 capabilities

- Older versions of browsers



New browser versions are continually added with support for HTML5 features

- Google Chrome supports most features



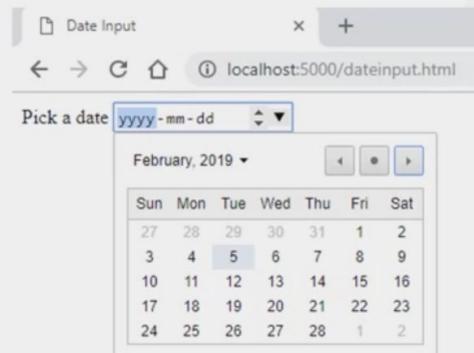
Support tables show features supported by different browsers

- Caniuse.com provides support tables for HTML5 and CSS3

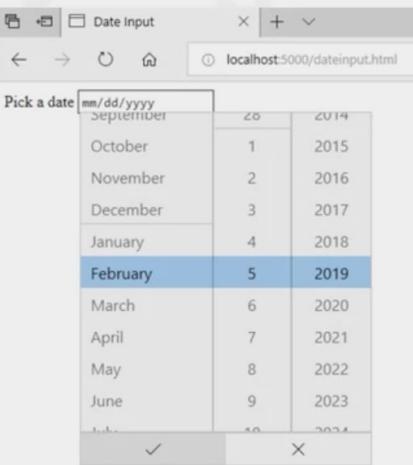
How Different Browsers Display HTML5 Elements

Browsers displaying <input type='date'> element

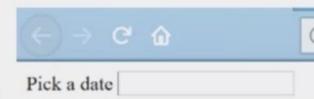
Google Chrome



Microsoft Edge



Firefox



Verifying HTML5 Support Using JavaScript

- Create DOM element using `document.createElement()`
 - This creates the element regardless of whether the browser supports it or not
- Check the DOM object for known property or method
 - The browser displays the supported element
- Set the `type` value for input elements
 - The browser displays the supported element
 - Otherwise, the browser returns `input type="text"` and displays a regular text field by default

Example: Verifying HTML5 Browser Support

```
<body>
  <form id="thisform">
  </form>
  <script>
    var datepicker = document.createElement("input");
    var formelement = document.getElementById("thisform");
    datepicker.setAttribute("type", "date");
    formelement.appendChild(datepicker);
    if (datepicker.type == "text") {
      alert("Date input not supported");
    }
  </script>
</body>
```

HTML Cheat Sheet

https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NilsInR5cCl6IkpxVCJ9eyJtZF9pbnN0cnVjdGlvbnNfdXJsljoiaHR0cHM6Ly9jZi1ib3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsB3Vkl0ICTURldmVsB3BlcINraWxsc05ldHdvcnstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9DaGVhdHNoZWV0cy9DMk0yJTIwY2hIYXRzaGVIdCUyMHYxLjlubWQiLCJ0b29sX3R5cGUoIjpbnN0cnVjdGlvbmFsLWxhYilsImFkbWluljpmYWxzZSwiaWF0ljoxNjc5NDAwNjI4fQ.YY9Y4xYkUq53KvHxr2TwTbCY_2JaCow2N6V1oa_Hu-E

More HTML5 tags

TAG	DESCRIPTION
<article>	Content from an external source - news article, blog, or forum
<aside>	Content aside from the page content
<audio>	Used to embed sound content
<canvas>	Used to draw graphics
<datalist>	Provides a list of predefined options for input controls
<details>	Used to show or hide contents
<embed>	Embeds an external application or interactive content into page
<figcaption>	Caption for the figure tag
<figure>	Specifies self-contained content
<footer>	The footer of a document or section
<header>	Specifies a group of introductory or navigational elements
<keygen>	Specifies a key-pair generator field used in forms

TAG	DESCRIPTION
<mark>	Represents highlighted text
<meter>	Used for measurements with minimum and maximum values
<nav>	Specifies navigation for a document
<output>	Represents the result of a calculation
<progress>	Specifies the state of work in progress
<section>	Defines sections in a document or article
<source>	Used to specify multiple media resources for media elements
<summary>	Defines a visible heading for the details tag
<time>	Used to specify the date or time in a document
<video>	Specifies video, such as a movie clip or video stream
<!-- -->	Comments in source that are not displayed in the browser

- Tags that provide structural elements:
 - **<div>**: Separates areas in a document into divisions, enabling you to apply different styles
 - **<article>, <section>, <header>, <footer>** : More specific than the generic div element.
 - **<aside>, <figure>, <figcaption>**: Enables content grouping
 - **<nav>**: Enables grouping of navigational links

```

<body>
  <nav>
    <div class="menu">
      <a href="#">Home</a> |
      <a href="about.html">About</a> |
      <a href="register.html">Register</a> |
      <a href="#">Sign in</a>
    </div>
  </nav>
  <h1>Pages for your</h1>
</body>

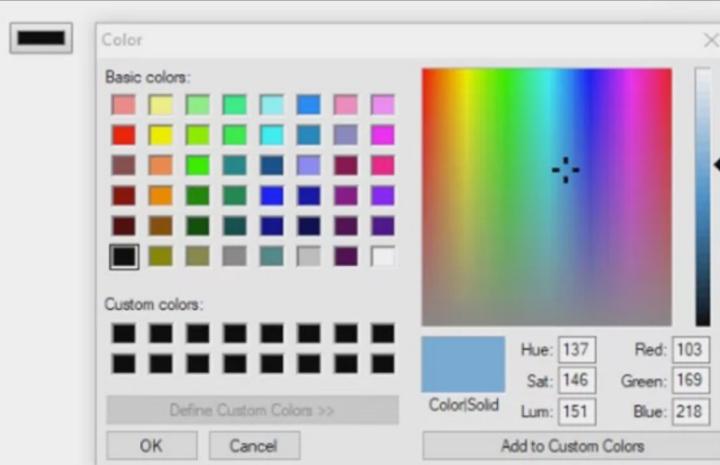
```

- For more information of the same:
<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NilsInR5cCl6IkpxVCJ9eyJtZF9pbnN0cnVjdGlvbnnNfdXjsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3Vkl0ICTURldmVsb3Blc1NraWxsc05ldHdvcnstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9SZWFkaW5ncy9Nb2R1bGUIMjAzL2NvbW1vbI9odG1sX3RhZ3NfYW5kX3N0cnVjdHVyYWxfZWxlbVVudHMubWQiLCJ0b29sX3R5cGUiOiJpbnN0cnVjdGlvbnnFsLWxhYilslmFkbWluljpmYWxzZSwiaWF0ljoxNjcxMTg3NTk5fQ.E55Cm1tBRr3NAYRhcMV4Twrxv5sLgtvpB0E7DIUTyg>

Common Attributes of the <input> tag

1. Color
2. Date
3. Datetime-local
4. Email
5. Number
6. Range
7. Search
8. Tel
9. URL

1. <input type = "color"/>

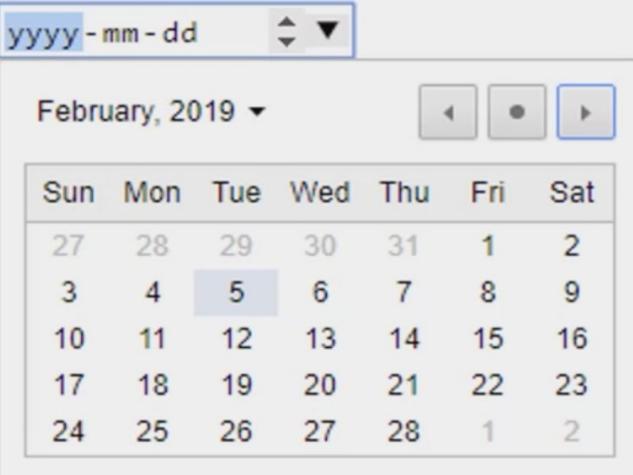


The image shows the color picker dialog box from Google Chrome. It features a grid of basic colors on the left and a larger color wheel on the right. At the bottom, there are fields for Hue, Sat, Lum, Red, Green, and Blue, along with their corresponding numerical values: Hue: 137, Sat: 146, Lum: 151, Red: 103, Green: 169, and Blue: 218. Buttons for 'OK' and 'Cancel' are at the bottom left, and 'Add to Custom Colors' is at the bottom right.

Google Chrome

- Allows a user to select color
- Dialog varies depending on browser
- Not all browsers support such input type
- Some browsers display the input type as regular text

2. <input type = "date"/>



The screenshot shows a date input field in Google Chrome. The input field has a placeholder 'yyyy-mm-dd' and a small calendar icon. Below it is a calendar dialog for February 2019. The days of the week are labeled Sun through Sat. The dates are arranged in a grid:

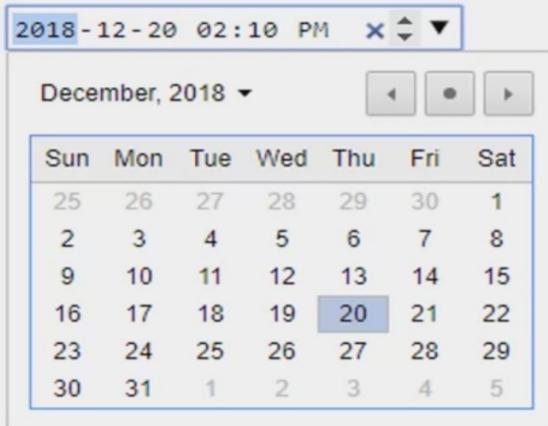
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2

Google Chrome

- A date control (year, month, day) with no time zone
- Input dialog varies depending on the browser

3. <input type = “dateime-local”/>

- Provides input for a date and time (year, month, day, hour, minute, AM/PM) with no time zone



In Google Chrome:

- Input field is displayed as a drop-down calendar
- Time can be typed or entered using the spinner control

4. <input type = "email"/>

- Displays a regular text input field
- Accepts only a valid email format



Please include an '@' in the email address. 'Non-conforming email' is missing an '@'.

5. <input type = “number”/>

- Takes a numeric value as input
- Can be used with minimum and maximum values
- Example: Google Chrome

```
<input type="number" min="5" max="10">
```

A screenshot of a web page in Google Chrome. At the top, there is a code snippet: <input type="number" min="5" max="10">. Below this, there is an input field containing the number 11. To the right of the input field is a 'Submit' button. A validation error message is displayed in a red-bordered box below the input field: 'Value must be less than or equal to 10.' An exclamation mark icon is part of the error message.

6. <input type = “range”/>

- Takes a numeric range as input
- Example: Google Chrome

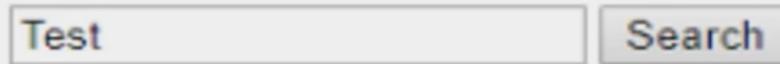
```
<input type="range" min="5" max="10">
```



- Additional JavaScript code is required to display the value of the slider

7. <input type = “search”/>

- Similar to the text field but differs in style



A screenshot of a search interface. It features a rectangular text input field with a thin gray border. Inside the field, the word "Test" is typed in a dark blue font. To the right of the input field is a rectangular button with a thin gray border and a slightly darker gray background. The word "Search" is printed in white, sans-serif capital letters on the button.

- WebKit-based browsers return a history of recently searched text strings
- The search input field on the Safari browser has rounded corners

8. <input type = “tel”/>

- Used for telephone numbers
- Provides a text entry field in the browsers
- Does not enforce numeric only input
 - Accepts characters, such as the plus sign and hyphens

```
<input type="tel" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
```

234-454-45244

Submit



Please match the requested format.

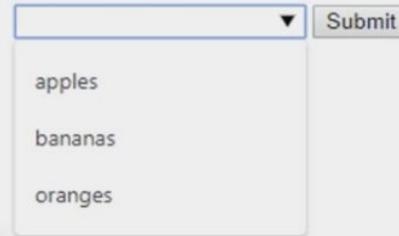
9. <input type = “URL”/>

- Used to validate if the input is a properly formatted URL or web address

Input Attribute : list

- Used with <datalist> tag that include the options list
- Useful for auto-complete functionality

```
<input type="text" list="fruits">
<datalist id="fruits">
  <option value="apples"></option>
  <option value="bananas"></option>
  <option value="oranges"></option>
</datalist>
```



A screenshot of a web browser interface. On the left, there is an input field with a blue border and a dropdown arrow icon. To the right of the input field is a "Submit" button. A dropdown menu is open, listing three items: "apples", "bananas", and "oranges".

Input Attribute : placeholder

- Provides an example of the input text format in a lighter shade of text

```
<input type="email" placeholder="example@email.com">
```

example@email.com

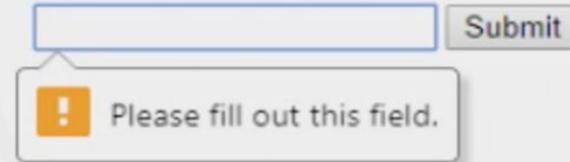
Submit

- Does not send the placeholder text as input during form submission

Input Attribute : required

- Indicates that the field is required

```
<input type="email" required>
```



Validation fallback



What if browser-based validation is not supported?

- Use prebuilt script libraries, such as JavaScript and JQuery libraries
- Assume browsers will support more HTML5 features over time
 - Leave all final validation to server-side
- Include custom client-side validation that is attached to the form submit event handler

<legend> and <fieldset> tag

- https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnnNfdXJsljoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UUuYXBwZG9tYWluLmNsb3Vkl0ICTURIdmVsB3BlclNraWxsc05ldHdvcmsQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9SZWFkaW5ncy9Nb2R1bGUIMjAzL0hUTUxfRm9ybV9MYWJfXzFfLm1kliwidG9vbF90eXBlIjoiaW5zdHJ1Y3Rpb25hbC1sYWliLCJhZG1pbil6ZmFsc2UsImhdCI6MTY3MTE4NzUyMX0.2bbSEe46OXVW0sNfk58pBY4nPppRu_Siz_WICXENk