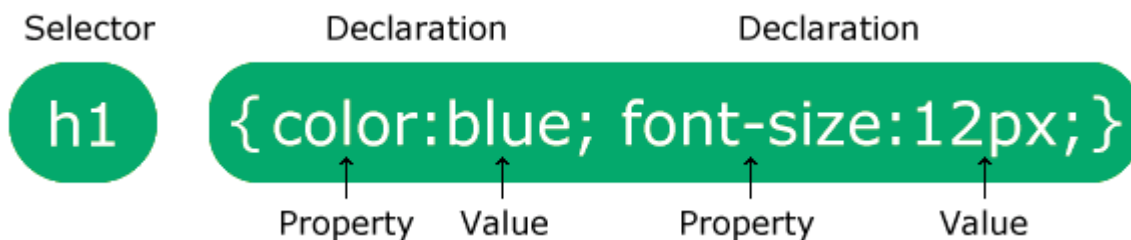


1.1 What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

CSS Syntax



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

1.1.1 Adding CSS (Inline, Internal, External)

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

Example

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

```
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

The external .css file should not contain any HTML tags.

Here is how the "mystyle.css" file looks:

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
```

```
margin-left: 20px;  
}
```

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

Example

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

```
<html>  
<head>  
<style>  
body {  
  background-color: linen;  
}  
  
h1 {  
  color: maroon;  
  margin-left: 40px;  
}  
</style>  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>
```

Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<html>
<body>
<h1 style="color:blue;text-align:center;">
This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

1.1.2 Html Links and Attribute(_self, _blank, _parent, _top)

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

HTML Links - Syntax

The HTML `<a>` tag defines a hyperlink. It has the following syntax

```
<html>
```

```
<body>

<h1>HTML Links</h1>

<p><a href="https://www.w3schools.com/">Visit W3Schools.com!</a></p>

</body>
</html>
```

HTML Links - The target attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The **target** attribute specifies where to open the linked document.

The **target** attribute can have one of the following values:

- **_self** - Default. Opens the document in the same window/tab as it was clicked
- **_blank** - Opens the document in a new window or tab
- **_parent** - Opens the document in the parent frame
- **_top** - Opens the document in the full body of the window

```
<html>
```

```
<body>
```

```
<h2>The target Attribute</h2>
```

```
<a href="https://www.w3schools.com/" target="_blank">Visit
W3Schools!</a>
```

```
<p>If target="_blank", the link will open in a new browser window or
tab.</p>
```

```
</body>
```

```
</html>
```

1.1.3 Absolute URL and Relative URL in <href>

Both examples above are using an **absolute URL** (a full web address) in the **href** attribute.

A local link (a link to a page within the same website) is specified with a **relative URL** (without the "https://www" part):

Example

```
<html>

<body>

<h2>Absolute URLs</h2>

<p><a href="https://www.w3.org/">W3C</a></p>

<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>

<p><a href="html_images.asp">HTML Images</a></p>

<p><a href="/css/default.asp">CSS Tutorial</a></p>

</body>

</html>
```

1.1.4 tag and its attributes(src,alt,style,width,height)

The HTML **** tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages. The **** tag creates a holding space for the referenced image.

The **** tag is empty, it contains attributes only, and does not have a closing tag.

The **** tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image

Syntax

```

```

The src Attribute

The required **src** attribute specifies the path (URL) to the image.

```
<html>
```

```
<body>
```

```
<h2>Alternative text</h2>
```

```
<p>The alt attribute should reflect the image content, so users who cannot see the  
image gets an understanding of what the image contains:</p>
```

```

```

```
</body>
```

```
</html>
```

The alt Attribute

The required **alt** attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the **alt** attribute should describe the image.

Image Size - Width and Height

You can use the **style** attribute to specify the width and height of an image.

```
<html>
```

```
<body>
```

```
<h2>Image Size</h2>
```

```
<p>Here we use the style attribute to specify the width and height of an image :</p>  
  
</body>  
</html>
```

Alternatively, you can use the width and height attributes:

Example:

```

```

1.2 HTML FORMS:

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

1.2.1 The <form> Element and their attributes

The HTML <form> element is used to create an HTML form for user input:

```
<form>  
.  
form elements  
.  
</form>
```

The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

1.2.1.1 Form (action,method,novalidate,autocomplete,target)

1. The Action Attribute

The **action** attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

```
<html>

<body>

<h2>HTML Forms</h2>

<form action="D:\KP\valid.htm">

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

<input type="submit" value="Submit">

</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called
"valid.htm".</p>

</body>

</html>
```

2. The Method Attribute

The **method** attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with **method="get"**) or as HTTP post transaction (with **method="post"**).

The default HTTP method when submitting form data is GET.

GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

`<form action="/action_page.php" method="get">`

POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

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

3. Novalidate

The **novalidate** attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

Example

A form with a novalidate attribute:

`<form action="D:\KP\valid.htm" novalidate>`

4. Autocomplete

The **autocomplete** attribute specifies whether a form should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

Example

A form with autocomplete on:

```
<form action="D:\KP\valid.htm" autocomplete="on">
```

5. The Target Attribute

The **target** attribute specifies where to display the response that is received after submitting the form.

The **target** attribute can have one of the following values:

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window

frameName The response is displayed in a named iframe

The default value is `_self` which means that the response will open in the current window.

Example

Here, the submitted result will open in a new browser tab:

```
<form action=" D:\KP\valid.htm" target="_blank">
```

1.2.1.2 Label, input (text, radio, button, checkboxes, submit/reset button)

1. The <label> Element

Notice the use of the `<label>` element in the example above.

The `<label>` tag defines a label for many form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.

The `for` attribute of the `<label>` tag should be equal to the `id` attribute of the `<input>` element to bind them together.

The <input> Element

The HTML `<input>` element is the most used form element.

An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Here are some examples:

Type	Description
<code><input type="text"></code>	Displays a single-line text input field
<code><input type="radio"></code>	Displays a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Displays a checkbox (for selecting zero or more of many choices)
<code><input type="submit"></code>	Displays a submit button (for submitting the form)
<code><input type="button"></code>	Displays a clickable button

Example:

```
<html>
```

```
<body>
```

```
<p> Registration Form</p>
```

```
<form action="D:\WD\registration_javascript\valid.htm">
```

```
<label for="fname">First name:</label><br>
```

```
<input type="text" id="fname" name="fname"><br>
<label for="lname">Last name:</label><br>
<input type="text" id="lname" name="lname"><br><br><br>
    <input type="radio" id="male" name="gender" value="Male">
    <label for="male">Male</label><br>
    <input type="radio" id="female" name="gender" value="Female">
    <label for="female">Female</label><br><br><br>
<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
<label for="vehicle1"> I have a bike</label><br>
<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
<label for="vehicle2"> I have a car</label><br><br>
<input type="submit" name="Submit">
<input type="reset" name="Reset">
</form>
</body>
</html>
```

1.2.1.3 Select (id, name, and <option>)

The **<select>** element defines a drop-down list:

```
<html>
<body>
<h2>The select Element</h2>
```

```
<p>The select element defines a drop-down list:</p>
<form action="/action_page.php">
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
<input type="submit">
</form>
</body>
</html>
```

1.2.1.4 textarea(name,rows,cols)

The **<textarea>** element defines a multi-line input field (a text area):

```
<html>
<body>
<h2>Textarea</h2>
<p>The textarea element defines a multi-line input field.</p>
```

```
<form action="/action_page.php">
  <textarea name="message" rows="10" cols="30">The cat was playing in the
  garden.</textarea>
<br><br>
<input type="submit">
</form>
</body>
</html>
```

1.2.1.5 Button(type,onclick)

The **<button>** element defines a clickable button:

```
<html>
<body>

<h2>The button Element</h2>

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

</body>
</html>
```

1.2.1.6 datalist

The **<datalist>** element specifies a list of pre-defined options for an **<input>** element.

Users will see a drop-down list of the pre-defined options as they input data.

The **list** attribute of the **<input>** element, must refer to the **id** attribute of the **<datalist>** element.

```
<html>
<body>
<h2>The datalist Element</h2>

<p>The datalist element specifies a list of pre-defined options for an input
element.</p>

<form action="/action_page.php">
<input list="browsers" name="browser">
<datalist id="browsers">
<option value="Internet Explorer">
<option value="Firefox">
<option value="Chrome">
<option value="Opera">
<option value="Safari">
</datalist>
```

```
<input type="submit">
</form>
</body>
</html>
```

1.3 Media: Video, Audio

Video

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

```
<html>
<body>
<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>
</body>
</html>
```

How it Works

The **controls** attribute adds video controls, like play, pause, and volume.

It is a good idea to always include **width** and **height** attributes. If height and width are not set, the page might flicker while the video loads.

The **<source>** element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the **<video>** and **</video>** tags will only be displayed in browsers that do not support the **<video>** element.

HTML <video>Autoplay

To start a video automatically, use the **autoplay** attribute:

```
<video width="320" height="240" autoplay>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">
```

Your browser does not support the video tag.

```
</video>
```

Add **muted** after **autoplay** to let your video start playing automatically (but muted):

```
<video width="320" height="240" autoplay muted>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">
```

Your browser does not support the video tag.

```
</video>
```

Audio

To play an audio file in HTML, use the **<audio>** element:

```
<html>
```

```
<body>
```

```
<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>
```

```
</body>
```

```
</html>
```

HTML Audio - How It Works

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 browser will use the first recognized format.

The text between the **<audio>** and **</audio>** tags will only be displayed in browsers that do not support the **<audio>** element.

HTML **<audio>** Autoplay

To start an audio file automatically, use the **autoplay** attribute:

```
<audio controls autoplay>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

Add **muted** after **autoplay** to let your audio file start playing automatically (but muted):

Example

```
<audio controls autoplay muted>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```