

HTML

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</
body>
</html>
```

- The `<!DOCTYPE html>` declaration defines this document to be HTML5
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The `<body>` element contains the visible page content
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

-
- HTML tags normally come **in pairs** like `<p>` and `</p>`
 - The first tag in a pair is the **start tag**, the second tag is the **end tag**
 - The end tag is written like the start tag, but with a **forward slash** inserted before the tag name
 - **HTML Headings**
 - HTML headings are defined with the `<h1>` to `<h6>` tags.
 - `<h1>` defines the most important heading. `<h6>` defines the least important heading:

- **Example**
- `<h1>`This is heading 1`</h1>`
- `<h2>`This is heading 2`</h2>`
- `<h3>`This is heading 3`</h3>`
- ---

HTML Lists

HTML List Example

An Unordered List:

- Item
- Item
- Item
- Item

An Ordered List:

1. First item
 2. Second item
 3. Third item
 4. Fourth item
-

Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

Unordered HTML List - Choose List Item Marker

The CSS **list-style-type** property is used to define the style of the list item marker:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

```
<ul style="list-style-type:disc">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

Example - Circle

```
<ul style="list-style-type:circle">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

Example - Square

```
<ul style="list-style-type:square">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

Example - None

```
<ul style="list-style-type:none">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

Example

```
<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

Ordered HTML List - The Type Attribute

The **type** attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

Numbers:

```
<ol type="1">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

Uppercase Letters:

```
<ol type="A">
<li>Coffee</li>
<li>Tea</li>
```

```
<li>Milk</li>
</ol>
```

Lowercase Letters:

```
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Uppercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Lowercase Roman Numbers:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The **<dl>** tag defines the description list, the **<dt>** tag defines the term (name), and the **<dd>** tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Nested HTML Lists

List can be nested (lists inside lists):

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

Note: List items can contain new list, and other HTML elements, like images and links, etc.

Horizontal Lists

HTML lists can be styled in many different ways with CSS.

One popular way is to style a list horizontally, to create a menu:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}
```

```
li a {  
  display: block;  
  color: white;  
  text-align: center;  
  padding: 16px;  
  text-decoration: none;  
}  
  
li a:hover {  
  background-color: #111111;  
}  
</style>  
</head>  
<body>  
  
<ul>  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>  
  
</body>  
</html>
```

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.

Note: A link does not have to be text. It can be an image or any other HTML element.

HTML Links - Syntax

In HTML, links are defined with the `<a>` tag:

```
<a href="url">link text</a>
```

Example

`<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>`

The **href** attribute specifies the destination address (https://www.w3schools.com/html/) of the link.

The **link text** is the visible part (Visit our HTML tutorial).

Clicking on the link text will send you to the specified address.

Note: Without a forward slash on subfolder addresses, you might generate two requests to the server. Many servers will automatically add a forward slash to the address, and then create a new request.

Local Links

The example above used an absolute URL (A full web address).

A local link (link to the same web site) is specified with a relative URL (without http://www....).

Example

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

HTML Table Example

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Defining an HTML Table

An HTML table is defined with the `<table>` tag.

Each table row is defined with the `<tr>` tag. A table header is defined with the `<th>` tag. By default, table headings are bold and centered. A table data/cell is defined with the `<td>` tag.

Example


```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Note: The <td> elements are the data containers of the table.
They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

HTML Table - Adding a Border

If you do not specify a border for the table, it will be displayed without borders.

A border is set using the CSS **border** property:

Example

```
table, th, td {
  border: 1px solid black;
}
```

Remember to define borders for both the table and the table cells.

HTML Table - Collapsed Borders

If you want the borders to collapse into one border, add the CSS **border-collapse** property:

Example

```
table, th, td {
  border: 1px solid black;
```

```
border-collapse: collapse;
}
```

HTML Table - Adding Cell Padding

Cell padding specifies the space between the cell content and its borders.

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, use the CSS **padding** property:

Example

```
th, td {
padding: 15px;
}
```

HTML Table - Left-align Headings

By default, table headings are bold and centered.

To left-align the table headings, use the CSS **text-align** property:

Example

```
th {
text-align: left;
}
```

HTML Table - Adding Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, use the CSS **border-spacing** property:

Example

```
table {
border-spacing: 5px;
}
```

Note: If the table has collapsed borders, border-spacing has no effect.

HTML Table - Cells that Span Many Columns

To make a cell span more than one column, use the **colspan** attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Cells that Span Many Rows

To make a cell span more than one row, use the **rowspan** attribute:

Example

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

HTML Table - Adding a Caption

To add a caption to a table, use the **<caption>** tag:

Example

```
<table style="width:100%">
  <caption>Monthly savings</caption>
```

```
<tr>
  <th>Month</th>
  <th>Savings</th>
</tr>
<tr>
  <td>January</td>
  <td>$100</td>
</tr>
<tr>
  <td>February</td>
  <td>$50</td>
</tr>
</table>
```

Note: The <caption> tag must be inserted immediately after the <table> tag.

A Special Style for One Table

To define a special style for a special table, add an **id** attribute to the table:

Example

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Now you can define a special style for this table:

```
table#t01 {
  width: 100%;
  background-color: #f1f1c1;
}
```

And add more styles:

```
table#t01 tr:nth-child(even) {
  background-color: #eee;
}
```

```
table#t01 tr:nth-child(odd) {  
    background-color: #fff;  
}  
table#t01 th {  
    color: white;  
    background-color: black;  
}
```

Summary

- Use the HTML **<table>** element to define a table
 - Use the HTML **<tr>** element to define a table row
 - Use the HTML **<td>** element to define a table data
 - Use the HTML **<th>** element to define a table heading
 - Use the HTML **<caption>** element to define a table caption
 - Use the CSS **border** property to define a border
 - Use the CSS **border-collapse** property to collapse cell borders
 - Use the CSS **padding** property to add padding to cells
 - Use the CSS **text-align** property to align cell text
 - Use the CSS **border-spacing** property to set the spacing between cells
 - Use the **colspan** attribute to make a cell span many columns
 - Use the **rowspan** attribute to make a cell span many rows
 - Use the **id** attribute to uniquely define one table
-

HTML Table Tags

Tag	Description
<u><table></u>	Defines a table
<u><th></u>	Defines a header cell in a table
<u><tr></u>	Defines a row in a table
<u><td></u>	Defines a cell in a table
<u><caption></u>	Defines a table caption
<u><colgroup></u>	Specifies a group of one or more columns in a table for formatting
<u><col></u>	Specifies column properties for each column within a <colgroup> element
<u><thead></u>	Groups the header content in a table
<u><tbody></u>	Groups the body content in a table
<u><tfoot></u>	Groups the footer content in a table

The <form> Element

The HTML **<form>** element defines a form that is used to collect user input:

`<form>`

.

form elements

.

`</form>`

An HTML form contains **form elements**.

Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

The `<input>` Element

The `<input>` element is the most important form element.

The `<input>` element can be displayed in several ways, depending on the **type** attribute.

Here are some examples:

Type	Description
<code><input type="text"></code>	Defines a one-line text input field
<code><input type="radio"></code>	Defines a radio button (for selecting one of many choices)
<code><input type="submit"></code>	Defines a submit button (for submitting the form)

Text Input

`<input type="text">` defines a one-line input field for **text input**:

Example

`<form>`

First name:`<br>`

`<input type="text" name="firstname"><br>`

Last name:`<br>`

`<input type="text" name="lastname">`

`</form>`

Input Type Text

`<input type="text">` defines a **one-line text input field**:

Example

```
<form>
First name:<br>
<input type="text" name="firstname"><br>
Last name:<br>
<input type="text" name="lastname">
</form>
```

Input Type Password

`<input type="password">` defines a **password field**:

Example

```
<form>
User name:<br>
<input type="text" name="username"><br>
User password:<br>
<input type="password" name="psw">
</form>
```

Characters in a password field are masked (shown as asterisks or circles).

Input Type Submit

`<input type="submit">` defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

Example

```
<form action="/action_page.php">
First name:<br>
<input type="text" name="firstname" value="Mickey"><br>
Last name:<br>
<input type="text" name="lastname" value="Mouse"><br><br>
<input type="submit" value="Submit">
</form>
```

If you omit the submit button's value attribute, the button will get a default text:

Example

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit">
</form>
```

The value Attribute

The **value** attribute specifies the initial value for an input field:

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John">
</form>
```

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages –

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back* button might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames

To use frames on a page we use `<frameset>` tag instead of `<body>` tag. The `<frameset>` tag defines, how to divide the window into frames. The **rows** attribute of `<frameset>` tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by `<frame>` tag and it defines which HTML document shall open into the frame.

Note – The <frame> tag deprecated in HTML5. Do not use this element.

Example

Following is the example to create three horizontal frames –

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <frameset rows = "10%,80%,10%">
    <frame name = "top" src = "/html/top_frame.htm" />
    <frame name = "main" src = "/html/main_frame.htm" />
    <frame name = "bottom" src = "/html/bottom_frame.htm" />

    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>

  </frameset>

</html>
```

Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically –

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <frameset cols = "25%,50%,25%">
    <frame name = "left" src = "/html/top_frame.htm" />
    <frame name = "center" src = "/html/main_frame.htm" />
    <frame name = "right" src = "/html/bottom_frame.htm" />

    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>

  </frameset>

</html>
```

Division:–

A section in a document that will be displayed in blue:

```
<div style="color:#0000FF">
  <h3>This is a heading</h3>
```

`<p>This is a paragraph.</p>`
`</div>`

Definition and Usage

The `<div>` tag defines a division or a section in an HTML document.

The `<div>` tag is used to group block-elements to format them with CSS.

Global Attributes

The `<div>` tag also supports the [Global Attributes in HTML](#).

Event Attributes

The `<div>` tag also supports the [Event Attributes in HTML](#).

What is CSS

CSS is an acronym stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

What does CSS do

- You can add new looks to your old HTML documents.
 - You can completely change the look of your website with only a few changes in CSS code.
-

Why use CSS

These are the three major benefits of CSS:

1) Solves a big problem

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will be become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

2) Saves a lot of time

CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

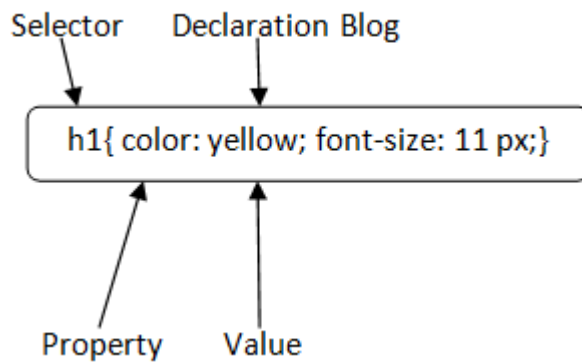
3) Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of

the website.

CSS Syntax

A CSS rule set contains a selector and a declaration block.



Selector: Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

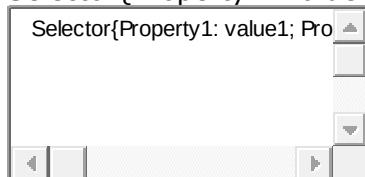
1. `color: yellow;`
2. `font-size: 11 px;`

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

Property: A Property is a type of attribute of HTML element. It could be color, border etc.

Value: Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

1. `Selector{Property1: value1; Property2: value2; .....;}`



CSS Selector

CSS selectors are used *to select the content you want to style*. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

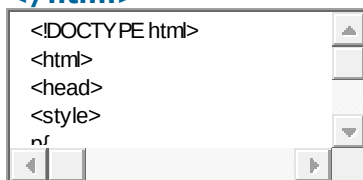
There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

1) CSS Element Selector

The element selector selects the HTML element by name.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `p{`
6. `text-align: center;`
7. `color: blue;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<p>`This style will be applied on every paragraph.`</p>`
13. `<p id="para1">`Me too!`</p>`
14. `<p>`And me!`</p>`
15. `</body>`
16. `</html>`



Test it Now

Output:

This style will be applied on every paragraph.

Me too!

And me!

2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

Let's take an example with the id "para1".

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `#para1 {`
6. `text-align: center;`
7. `color: blue;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<p id="para1">Hello Javatpoint.com</p>`
13. `<p>This paragraph will not be affected.</p>`
14. `</body>`
15. `</html>`



Test it Now

Output:

This paragraph will not be affected.

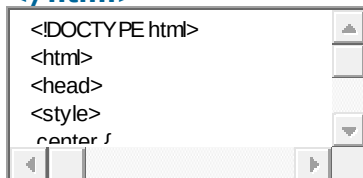
3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

Note: A class name should not be started with a number.

Let's take an example with a class "center".

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `.center {`
6. `text-align: center;`
7. `color: blue;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<h1 class="center">`This heading is blue and center-aligned.`</h1>`
13. `<p class="center">`This paragraph is blue and center-aligned.`</p>`
14. `</body>`
15. `</html>`



Test it Now

Output:

This heading is blue and center-aligned.

This paragraph is blue and center-aligned.

CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

Let's see an example.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `p.center {`
6. `text-align: center;`
7. `color: blue;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<h1 class="center">`This heading is not affected`</h1>`
13. `<p class="center">`This paragraph is blue and center-aligned.`</p>`
14. `</body>`
15. `</html>`



Test it Now

Output:

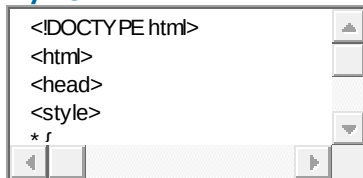
This heading is not affected

This paragraph is blue and center-aligned.

4) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `* {`
6. `color: green;`
7. `font-size: 20px;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<h2>`This is heading`</h2>`
13. `<p>`This style will be applied on every paragraph.`</p>`
14. `<p id="para1">`Me too!`</p>`
15. `<p>`And me!`</p>`
16. `</body>`
17. `</html>`



Test it Now

Output:

This is heading

This style will be applied on every paragraph.

Me too!

And me!

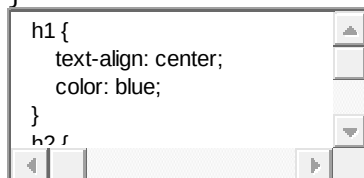
5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

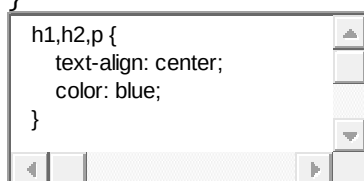
Let's see the CSS code without group selector.

1. h1 {
2. text-align: center;
3. color: blue;
4. }
5. h2 {
6. text-align: center;
7. color: blue;
8. }
9. p {
10. text-align: center;
11. color: blue;
12. }



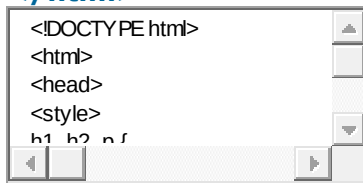
As you can see, you need to define CSS properties for all the elements. It can be grouped in following ways:

1. h1,h2,p {
2. text-align: center;
3. color: blue;
4. }



Let's see the full example of CSS group selector.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `h1, h2, p {`
5. `text-align: center;`
7. `color: blue;`
3. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<h1>Hello Javatpoint.com</h1>`
13. `<h2>Hello Javatpoint.com (In smaller font)</h2>`
14. `<p>This is a paragraph.</p>`
15. `</body>`
16. `</html>`



Test it Now

Output:

Hello Javatpoint.com

Hello Javatpoint.com (In smaller font)

This is a paragraph.

How to add CSS

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

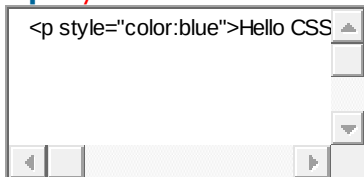
1. Inline CSS
2. Internal CSS
3. External CSS

1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

For example:

1. `<p style="color:blue">Hello CSS</p>`



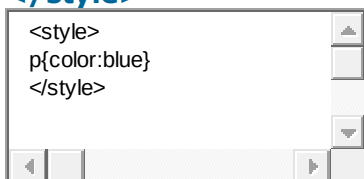
For more visit here: [Inline CSS](#)

2) Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

For example:

1. `<style>`
2. `p{color:blue}`
3. `</style>`



For more visit here: [Internal CSS](#)

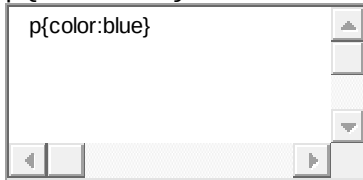
3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write

all the CSS code in a css file. Its extension must be .css for example style.css.

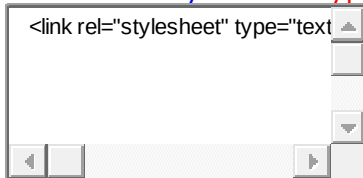
For example:

1. `p{color:blue}`



You need to link this style.css file to your html pages like this:

1. `<link rel="stylesheet" type="text/css" href="style.css">`



The link tag must be used inside head section of html.

Inline CSS

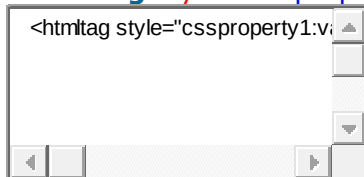
We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

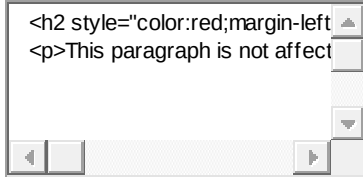
Syntax:

1. `<htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>`



Example:

1. `<h2 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h2>`
2. `<p>This paragraph is not affected.</p>`



Test it Now

Output:

Inline CSS is applied on this heading.

This paragraph is not affected.

Disadvantages of Inline CSS

- You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
- These styles cannot be reused anywhere else.
- These styles are tough to be edited because they are not stored at a single place.
- It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
- Inline CSS does not provide browser cache advantages.

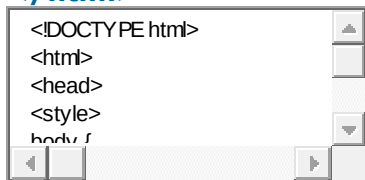
Internal CSS

The internal style sheet is used to add a unique style for a single document. It is defined in `<head>` section of the HTML page inside the `<style>` tag.

Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`

5. body {
5. background-color: linen;
7. }
3. h1 {
9. color: red;
10. margin-left: 80px;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1>**The internal style sheet is applied on this heading.**</h1>**
16. **<p>**This paragraph will not be affected.**</p>**
17. **</body>**
18. **</html>**



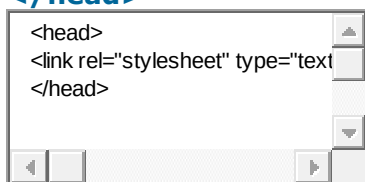
External CSS

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the `<link>` tag on every pages and the `<link>` tag should be put inside the head section.

Example:

1. **<head>**
2. **<link rel="stylesheet" type="text/css" href="mystyle.css">**
3. **</head>**

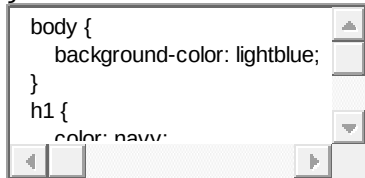


The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

Let's take an example of a style sheet file named "mystyle.css".

File: mystyle.css

1. body {
2. background-color: lightblue;
3. }
4. h1 {
5. color: navy;
5. margin-left: 20px;
7. }



Note: You should not use a space between the property value and the unit. For example: It should be margin-left:20px not margin-left:20 px.

CSS Font

CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more. You have already studied how to make text bold or underlined. Here, you will also know how to resize your font using percentage.

These are some important font attributes:

1. **CSS Font color:** This property is used to change the color of the text. (standalone attribute)
2. **CSS Font family:** This property is used to change the face of the font.
3. **CSS Font size:** This property is used to increase or decrease the size of the font.
4. **CSS Font style:** This property is used to make the font bold, italic or oblique.

5. **CSS Font variant:** This property creates a small-caps effect.
 6. **CSS Font weight:** This property is used to increase or decrease the boldness and lightness of the font.
-

1) CSS Font Color

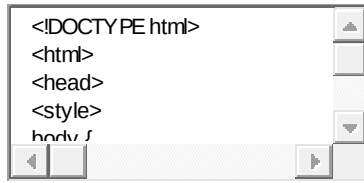
CSS font color is a standalone attribute in CSS although it seems that it is a part of CSS fonts. It is used to change the color of the text.

There are three different formats to define a color:

- By a color name
- By hexadecimal value
- By RGB

In the above example, we have defined all these formats.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.     font-size: 100%;
7. }
8. h1 { color: red; }
9. h2 { color: #9000A1; }
10. p { color:rgb(0, 220, 98); }
11. }
12. </style>
13. </head>
14. <body>
15. <h1>This is heading 1</h1>
16. <h2>This is heading 2</h2>
17. <p>This is a paragraph.</p>
18. </body>
19. </html>
```



Test it Now

Output:

This is heading 1

This is heading 2

This is a paragraph.

2) CSS Font Family

CSS font family can be divided in two types:

- Generic family: It includes Serif, Sans-serif, and Monospace.
- Font family: It specifies the font family name like Arial, New Times Roman etc.

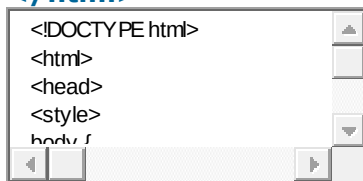
Serif: Serif fonts include small lines at the end of characters. Example of serif: Times new roman, Georgia etc.

Sans-serif: A sans-serif font doesn't include the small lines at the end of characters. Example of Sans-serif: Arial, Verdana etc.



1. `<!DOCTYPE html>`
2. `<html>`

3. **<head>**
4. **<style>**
5. body {
5. font-size: 100%;
7. }
3. h1 { font-family: sans-serif; }
9. h2 { font-family: serif; }
10. p { font-family: monospace; }
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1>**This heading is shown in sans-serif.**</h1>**
16. **<h2>**This heading is shown in serif.**</h2>**
17. **<p>**This paragraph is written in monospace.**</p>**
18. **</body>**
19. **</html>**



Test it Now

Output:

This heading is shown in sans-serif.

This heading is shown in serif.

This paragraph is written in monospace.

3) CSS Font Size

CSS font size property is used to change the size of the font.

These are the possible values that can be used to set the font size:

Font Size Value	Description
xx-small	used to display the extremely small text size.
x-small	used to display the extra small text size.
small	used to display small text size.
medium	used to display medium text size.
large	used to display large text size.
x-large	used to display extra large text size.
xx-large	used to display extremely large text size.
smaller	used to display comparatively smaller text size.
larger	used to display comparatively larger text size.
size in pixels or %	used to set value in percentage or in pixels.

1. `<html>`
2. `<head>`
3. `<title>`Practice CSS font-size property`</title>`
4. `</head>`
5. `<body>`
5. `<p style="font-size:xx-small;">` This font size is extremely small.`</p>`
7. `<p style="font-size:x-small;">` This font size is extra small`</p>`
3. `<p style="font-size:small;">` This font size is small`</p>`
9. `<p style="font-size:medium;">` This font size is medium. `</p>`
10. `<p style="font-size:large;">` This font size is large. `</p>`
11. `<p style="font-size:x-large;">` This font size is extra large. `</p>`
12. `<p style="font-size:xx-large;">` This font size is extremely large. `</p>`
13. `<p style="font-size:smaller;">` This font size is smaller. `</p>`
14. `<p style="font-size:larger;">` This font size is larger. `</p>`

15. `<p style="font-size:200%;">` This font size is set on 200%. `</p>`

16. `<p style="font-size:20px;">` This font size is 20 pixels. `</p>`

17. `</body>`

18. `</html>`



Test it Now

Output:

This font size is extremely small.

This font size is extra small

This font size is small

This font size is medium.

This font size is large.

This font size is extra large.

This font size is
extremely large.

This font size is smaller.

This font size is larger.

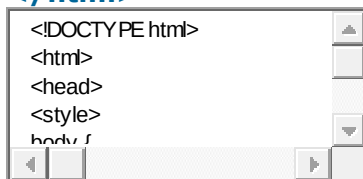
This font size is set on 200%.

This font size is 20 pixels.

4) CSS Font Style

CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
5. font-size: 100%;
7. }
3. h2 { font-style: italic; }
9. h3 { font-style: oblique; }
10. h4 { font-style: normal; }
11. }
12. </style>
13. </head>
14. <body>
15. <h2>This heading is shown in italic font.</h2>
16. <h3>This heading is shown in oblique font.</h3>
17. <h4>This heading is shown in normal font.</h4>
18. </body>
19. </html>



Test it Now

Output:

This heading is shown in italic font.

This heading is shown in oblique font.

This heading is shown in normal font.

5) CSS Font Variant

CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `p { font-variant: small-caps; }`
5. `h3 { font-variant: normal; }`
7. `</style>`
3. `</head>`
9. `<body>`
10. `<h3>`This heading is shown in normal font.`</h3>`
11. `<p>`This paragraph is shown in small font.`</p>`
12. `</body>`
13. `</html>`



Test it Now

Output:

This heading is shown in normal font.

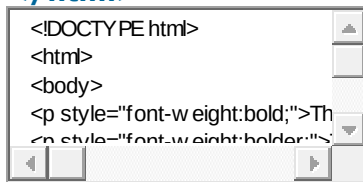
THIS PARAGRAPH IS SHOWN IN SMALL FONT.

6) CSS Font Weight

CSS font weight property defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200..... upto 900).

1. `<!DOCTYPE html>`

2. `<html>`
3. `<body>`
4. `<p style="font-weight:bold;">This font is bold.</p>`
5. `<p style="font-weight:bolder;">This font is bolder.</p>`
5. `<p style="font-weight:lighter;">This font is lighter.</p>`
7. `<p style="font-weight:100;">This font is 100 weight.</p>`
3. `<p style="font-weight:200;">This font is 200 weight.</p>`
9. `<p style="font-weight:300;">This font is 300 weight.</p>`
10. `<p style="font-weight:400;">This font is 400 weight.</p>`
11. `<p style="font-weight:500;">This font is 500 weight.</p>`
12. `<p style="font-weight:600;">This font is 600 weight.</p>`
13. `<p style="font-weight:700;">This font is 700 weight.</p>`
14. `<p style="font-weight:800;">This font is 800 weight.</p>`
15. `<p style="font-weight:900;">This font is 900 weight.</p>`
16. `</body>`
17. `</html>`



Test it Now

Output:

This font is bold.

This font is bolder.

This font is lighter.

This font is 100 weight.

This font is 200 weight.

This font is 300 weight.

This font is 400 weight.

This font is 500 weight.

This font is 600 weight.

This font is 700 weight.

This font is 800 weight.

This font is 900 weight.

Java script

JavaScript is *an object-based scripting language* that is lightweight and cross-platform.

JavaScript is not compiled but translated. The JavaScript Translator (embedded in browser) is responsible to translate the JavaScript code.

Where JavaScript is used

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation
- Dynamic drop-down menus
- Displaying data and time