
بسم الله الرحمن الرحيم

Web Design & Development

: Introduction to HTML, CSS and JavaScript

Syed Hussain Ali Shah Gillani

Lecturer, KICSIT Campus

Introduction to Web Design

- Web design is about making websites that look good and work well, so people enjoy using them. It includes things like how the website is laid out, the colors used, the fonts, and the images to make sure everything is easy to use and understand.
- The process involves thinking about what the website will look like, how it will be organized, and planning how the content will be shown to visitors. It's all about creating a site that's both attractive and easy for people to navigate.

Web Development Technologies

Introduction To Html

- **What Is HTML?**
- HTML (HyperText Markup Language) is the basic language used to create and structure web pages. It tells the web browser how to display the content on a webpage, HTML uses tags to mark different parts of the content. Each tag describes the role of that part, like headings, paragraphs, images, or links etc. For example:
- **<h1> to <h6>:** Defines headings.
- **<p>:** Defines paragraphs.
- **<img>:** Displays images.
- **<a>:** Creates links.
- HTML is the foundation of every webpage, allowing browsers to read and display content properly.
- ch as text, images, and links.

HTML Editor

- An HTML editor is a software tool used to write and edit HTML code for creating web pages. It helps developers create the structure of a website by providing a space to write HTML, CSS, and sometimes JavaScript.

- Text-based HTML Editors:
- These editors allow you to write HTML code manually. They don't add any extra formatting, so you have complete control over the code.
- Examples include:
- Notepad++
- Sublime Text
- Visual Studio Code

HTML ELEMENTS

- **The HTML element is everything from the start tag to the end tag:**
- <tagname>Content goes here...</tagname>
- Examples of some HTML elements:
- <h1>My First Heading</h1>

- My first paragraph.

Tags

- The essence of HTML programming is tags
- A tag is a keyword enclosed by angle brackets (Example: <I>)
- There are opening and closing tags for many but not all tags; The affected text is
- between the two tags
- The opening and closing tags use the same command except the closing tag
- contains and additional forward slash /
- For example, the expression <B> Warning </B> would cause the word ‘Warning’
- to appear in bold face on a Web page

Nested Tags

- Whenever you have HTML tags within other HTML tags, you must close the nearest tag first
- Example:
- <H1> <I> The Nation </I> </H1>

HTML ATTRIBUTES

- All HTML elements can have attributes
- Attributes provide additional information about elements
- Attributes are always specified in the start tag
- ***Attributes usually come in name/value pairs like: name="value***
- **The href Attribute**

  Visit W3Schools
- The src Attribute

  ******

- The width and height Attributes

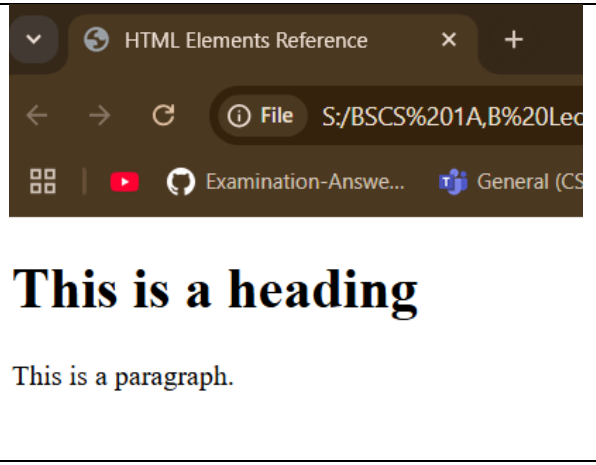
  <img src="img_girl.jpg" width="500" height="600">
- The style Attribute

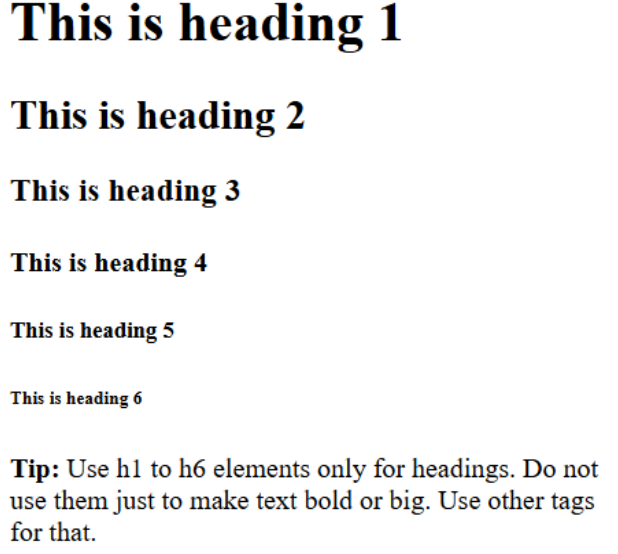
  *** This is a red paragraph. ***

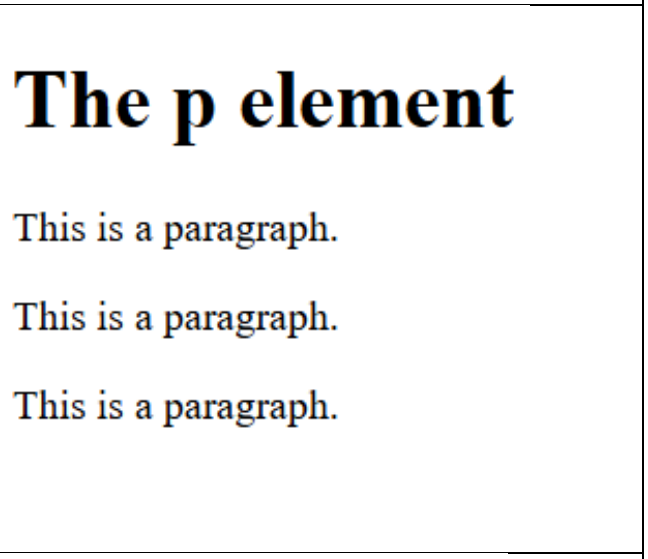
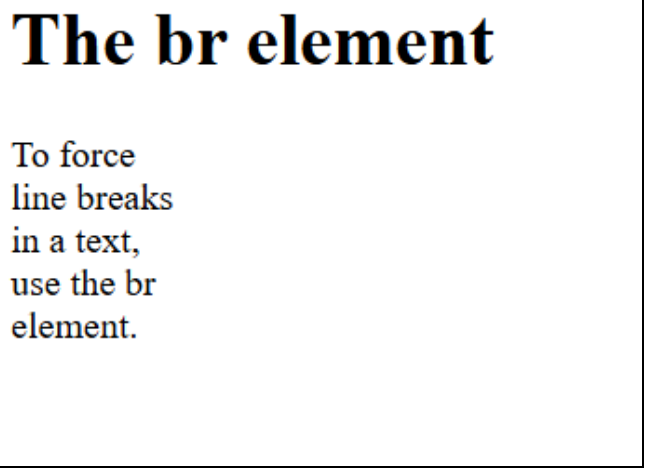
Tag Description

Basic HTML

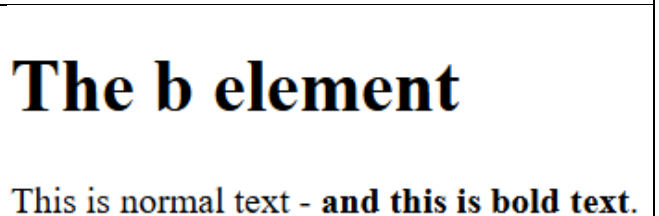
Tag	Def	Code Example	Browser View
<!DOCTYPE>	Defines the document type	<!DOCTYPE html> <html> <head> <title>Title of the document</title> </head> <body> The content of the document…… </body> </html>	The content of the document……
<html>	Defines an HTML document/ tag represents the root of an HTML document.	<!DOCTYPE html> <html lang="en"> <head> <title>Title of the document</title> </head> <body> This is a heading This is a paragraph. </body> </html>	**This is a heading** This is a paragraph.

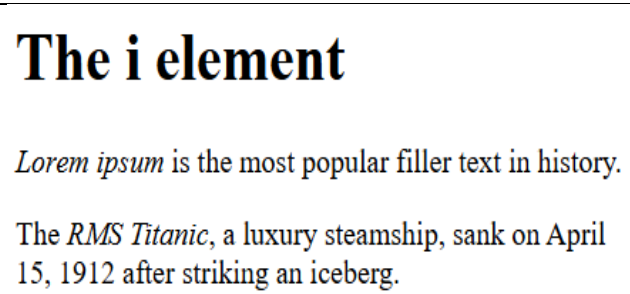
<head>	Contains metadata/information for the document The following elements can go inside the <head> element: <title> (required in every HTML document) <style> <base> <link> <meta> <script> <noscript>	<!DOCTYPE html> <html lang="en"> <head> <title>Title of the document</title> </head> <body> This is a heading of HTML This is a paragraph. </body> </html>	
<title>	Defines a title for the document	<!DOCTYPE html> <html> <head> <title>HTML Elements Reference</title> </head> <body> This is a heading This is a paragraph.	

		</body> </html>	
<body>	Defines the document's body The <body> element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.	<html> <head> <title>Title of the document</title> </head> <body> This is a heading This is a paragraph. </body> </html>	
<h1> to <h6>	Defines HTML headings The <h1> to <h6> tags are used to define HTML headings. <h1> defines the most important heading. <h6> defines the least important heading.	<!DOCTYPE html> <html> <body> This is heading 1 This is heading 2 This is heading 3 This is heading 4 This is heading 5 This is heading 6 Tip: Use h1 to h6 elements only for headings. Do not use them just to make text bold or big. Use other tags for that. </body>	

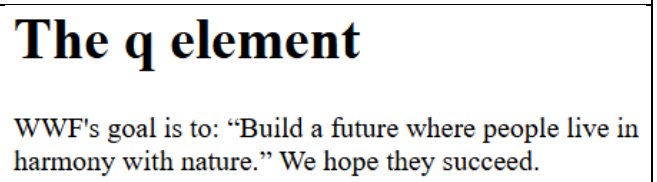
		</html>	
	Defines a paragraph The tag defines a paragraph.	<!DOCTYPE html> <html> <body> The p element This is a paragraph. This is a paragraph. This is a paragraph. </body> </html>	
 	Inserts a single line break The tag inserts a single line break. The tag is an empty tag which means that it has no end tag.	<!DOCTYPE html> <html> <body> The br element To force line breaks in a text, use the br element. </body> </html>	

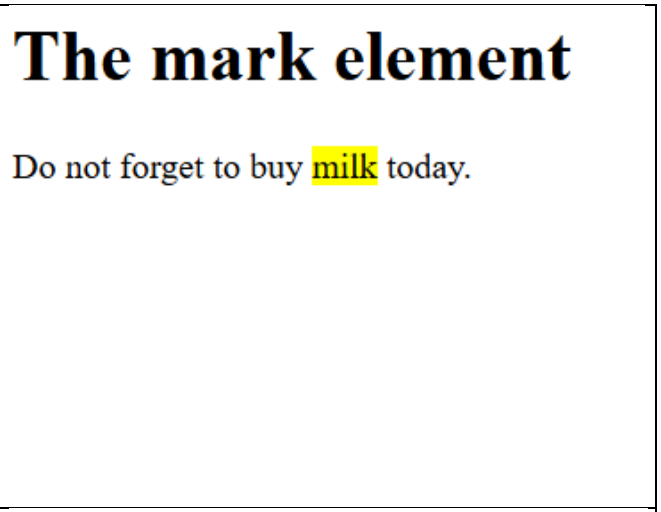
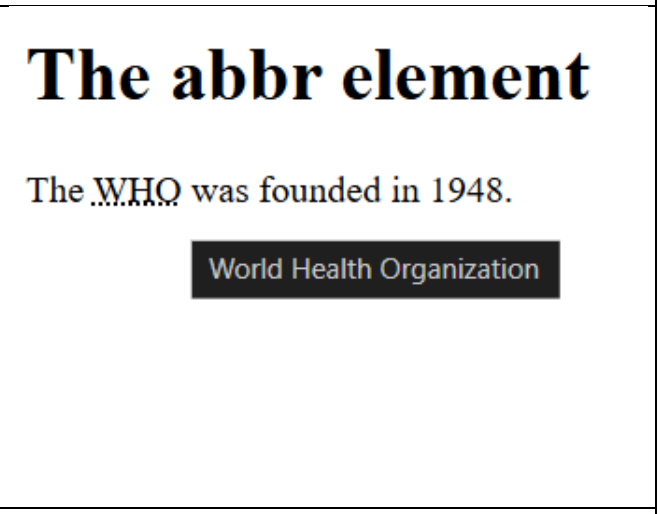
<hr>	Defines a thematic change in the content The <hr> element is most often displayed as a horizontal rule that is used to separate content (or define a change) in an HTML page.	<!DOCTYPE html> <html> <body> The Main Languages of the Web HTML is the standard markup language for creating Web pages. <hr> </body> </html>	

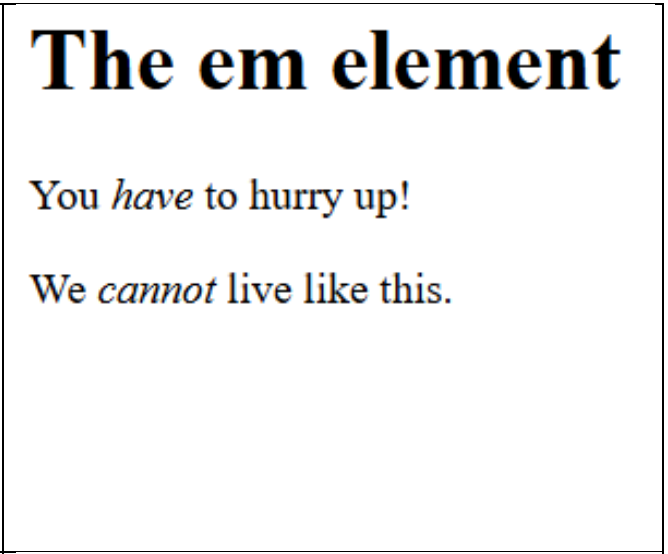
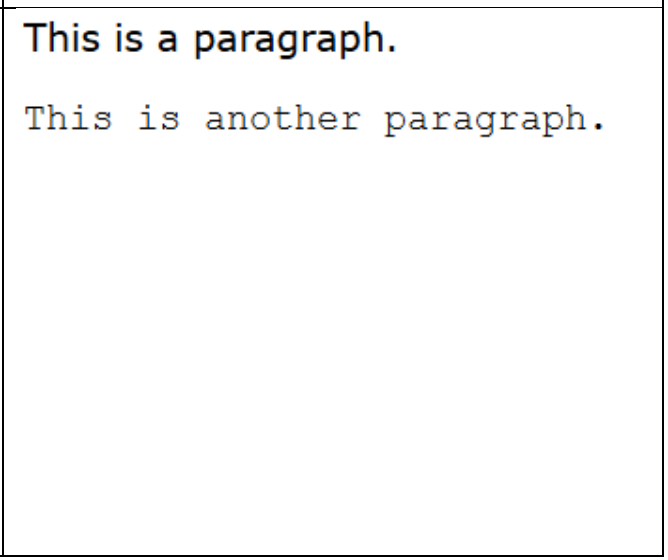
Formatting			
Tag	**Def**	**Code Example**	**Browser View**
<b>	Defines bold text The <b> tag specifies bold text without any extra importance.	<!DOCTYPE html> <html> <body> The b element	

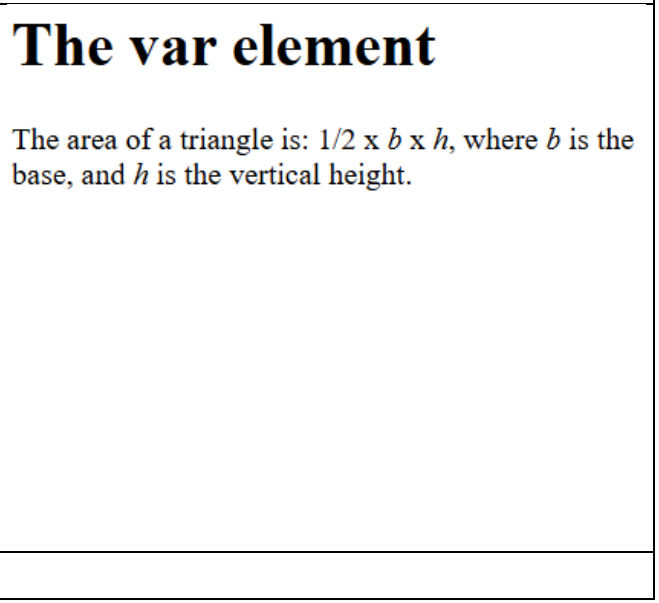
		This is normal text – and this is bold text. </body> </html>	
	Defines centered text	<!DOCTYPE html> <html> <body> This is a heading This is a paragraph. This is a div. </body> </html>	
	Defines a part of text in an alternate voice or mood	<!DOCTYPE html> <html> <body> The i element Lorem ipsum is the most popular filler text in history.	

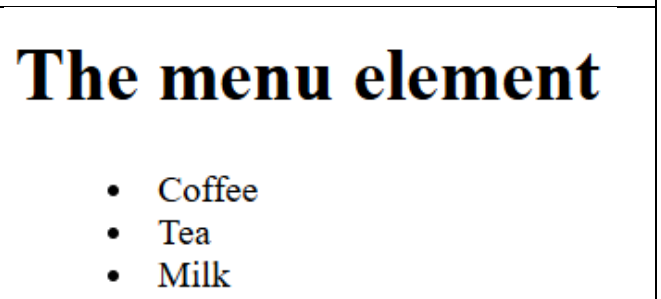
		The RMS Titanic, a luxury steamship, sank on April 15, 1912 after striking an iceberg. </body> </html>	
<u>	Defines some text that is unarticulated and styled differently from normal text	<!DOCTYPE html> <html> <body> The u element This is some <u>mispeled</u> text. </body> </html>	
	Defines important text The tag is used to define text with strong importance. The content inside is typically displayed in **bold.**	<!DOCTYPE html> <html> <body> The strong element This text is normal. This text is important! 	

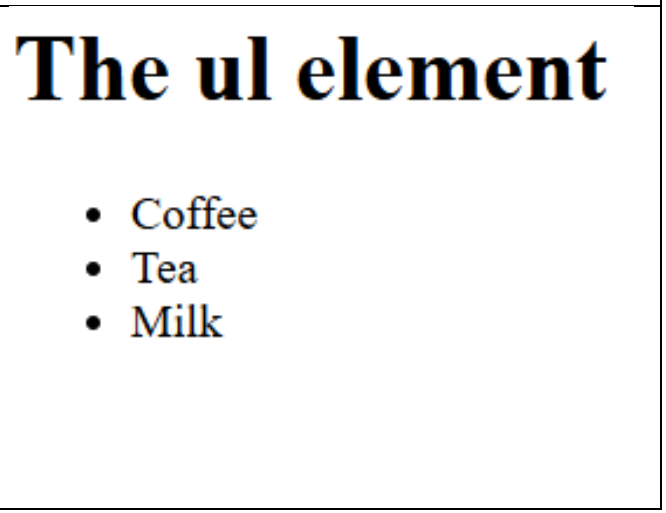
		</body> </html>	
<small>	Defines smaller text	<!DOCTYPE html> <html> <body> The small element This is some normal text. This is some smaller text. </body> </html>	
<q>	Defines a short quotation	<!DOCTYPE html> <html> <body> The q element WWF's goal is to: <q>Build a future where people live in harmony with nature.</q> We hope they succeed. </body> </html>	

<mark>	Defines marked/highlighted text	<!DOCTYPE html> <html> <body> The mark element Do not forget to buy <mark>milk</mark> today. </body> </html>	
<abbr>	Defines an abbreviation or an acronym	<!DOCTYPE html> <html> <body> The abbr element The <abbr title="World Health Organization">WHO</abbr> was founded in 1948. </body> </html>	

	Defines emphasized text	<!DOCTYPE html> <html> <body> The em element You have to hurry up! We cannot live like this. </body> </html>	
<font>	Not supported in HTML5. Use CSS instead.	<!DOCTYPE html> <html> <body> This is a paragraph. This is another paragraph. </body> </html>	

<strike>	Not supported in HTML5. Use <del> or <s> instead. Defines strikethrough text	<!DOCTYPE html> <html> <body> The del element My favorite color is ~~blue~~ <ins>red</ins>! </body> </html>	
<var>	Defines a variable	<!DOCTYPE html> <html> <body> The var element The area of a triangle is: 1/2 x <var>b</var> x <var>h</var>, where <var>b</var> is the base, and <var>h</var> is the vertical height. </body> </html>	

List			
Tag	**Def**	**Code Example**	**Browser View**
<menu>	Defines an alternative unordered list	<!DOCTYPE html> <html> <body> The menu element <menu> - Coffee - Tea - Milk </menu>	

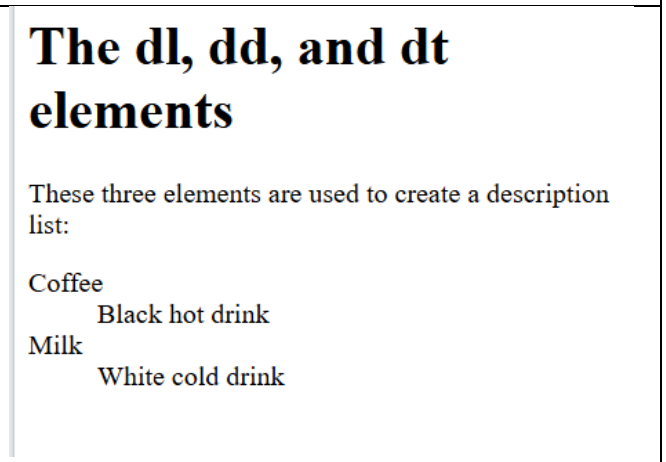
		</body> </html>	
	Defines an unordered list Use the tag together with the tag to create unordered lists.	<!DOCTYPE html> <html> <body> The ul element - Coffee - Tea - Milk </body> </html>	

	Defines an ordered list The tag is used to define each list item. An ordered list can be numerical or alphabetical.	<!DOCTYPE html> <html> <body> The ol element - Coffee - Tea - Milk <ol start="50"> - Coffee - Tea - Milk Numbered - List item 1 - List item 2 - List item 3 - List item 4 Numbered Special Start <ol start="5">	**The ol element** 1. Coffee 2. Tea 3. Milk 50. Coffee 51. Tea 52. Milk Numbered 1. List item 1 2. List item 2 3. List item 3 4. List item 4 Numbered Special Start 5. List item 1 6. List item 2 7. List item 3 8. List item 4 Lowercase Letters a. List item 1 b. List item 2 c. List item 3 d. List item 4 Capital Letters A. List item 1 B. List item 2 C. List item 3 D. List item 4

		- List item 1 - List item 2 - List item 3 - List item 4 Lowercase Letters <ol type="a"> - List item 1 - List item 2 - List item 3 - List item 4 Capital Letters <ol type="A"> - List item 1 - List item 2 - List item 3 - List item 4 Capital Letters Special Start <ol type="A" start="3">	Capital Letters Special Start C. List item 1 D. List item 2 E. List item 3 F. List item 4 Lowercase Roman Numerals i. List item 1 ii. List item 2 iii. List item 3 iv. List item 4 Capital Roman Numerals I. List item 1 II. List item 2 III. List item 3 IV. List item 4 Capital Roman Numerals Special Start VII. List item 1 VIII. List item 2 IX. List item 3 X. List item 4

		- List item 1 - List item 2 - List item 3 - List item 4 Lowercase Roman Numerals <ol type="i"> - List item 1 - List item 2 - List item 3 - List item 4 Capital Roman Numerals <ol type="I"> - List item 1 - List item 2 - List item 3 - List item 4 Capital Roman Numerals Special Start <ol type="I" start="7">	

		- List item 1 - List item 2 - List item 3 - List item 4 </body> </html>	
	Defines a list item	<!DOCTYPE html> <html> <body> The ol and ul elements The ol element defines an ordered list: - Coffee - Tea - Milk The ul element defines an unordered list: - Coffee - Tea - Milk	**The ol and ul elements** The ol element defines an ordered list: 1. Coffee 2. Tea 3. Milk The ul element defines an unordered list: • Coffee • Tea • Milk

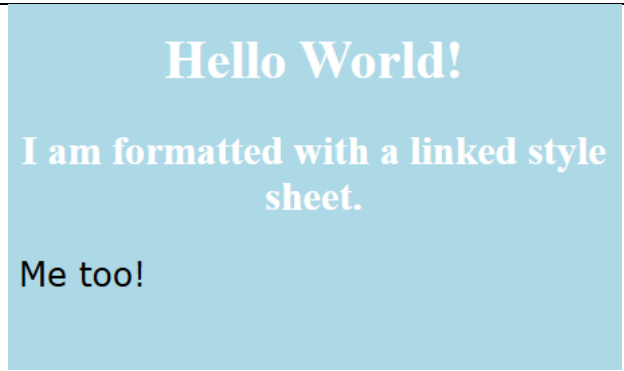
		 </body> </html>	
<dl>	Defines a description list	<!DOCTYPE html> <html> <body> The dl, dd, and dt elements These three elements are used to create a description list: <dl> <dt>Coffee</dt> <dd>Black hot drink</dd> <dt>Milk</dt> <dd>White cold drink</dd> </dl> </body> </html>	
<dt>	Defines a term/name in a description list		

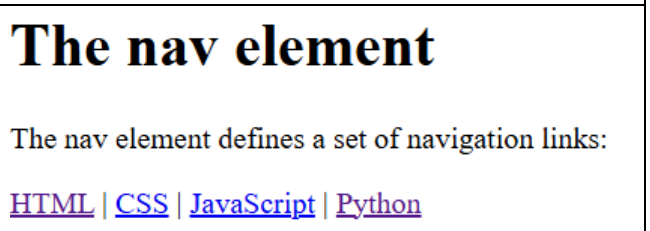
<dd>	Defines a description of a term/name in a description list		

Image/Audio/Video			
Tag	**Def**	**Code Example**	**Browser View**
<img>	Defines an image Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image. The <img> tag has two required attributes: src - Specifies the path to the image alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed	<!DOCTYPE html> <html> <body> The img element <img src="img_girl.jpg" alt="Girl in a jacket" width="200" height="200"> </body> </html>	

<figcaption>	Defines a caption for a <figure> element	<!DOCTYPE html> <html> <body> The figure and figcaption element <figure> <img src=" img_girl.jpg " alt="Trulli" style="width:100%"> <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption> </figure> </body> </html>	**The figure and figcaption element**  Fig.1 - Trulli, Puglia, Italy.
<audio>	Defines sound content	<!DOCTYPE html> <html> <body> The audio element Click on the play button to play a sound: <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>	**The audio element** Click on the play button to play a sound: 0:00 / 0:01

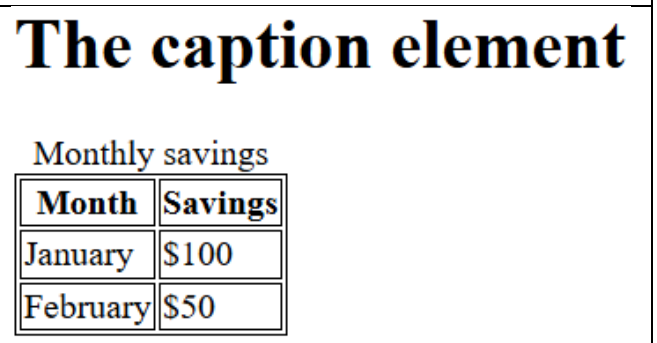
<source>	Defines multiple media resources for media elements (<video>, <audio> and <picture>)	<!DOCTYPE html> <html> <body> The source element Click on the play button to play a sound: <audio controls> <source src="horse.ogg" type="audio/ogg"> <source src ="horse.mp3" type="audio/mpeg"> Your browser does not support the audio element. </audio> </body> </html>	**The source element** Click on the play button to play a sound: 0:00 / 0:01
<video>	Defines a video or movie	<!DOCTYPE html> <html> <body> The video element <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>	**The video element** 

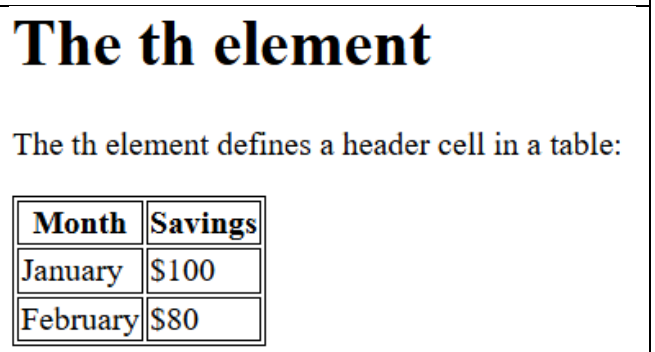
Links			
Tag	**Def**	**Code Example**	**Browser View**
<a>	Defines a hyperlink The <a> tag defines a hyperlink, which is used to link from one page to another.	<!DOCTYPE html> <html> <body> The a element Visit W3Schools.com! </body> </html>	
<link>	Defines the relationship between a document and an external resource (most used to link to style sheets)	<!DOCTYPE html> <html> <head> <link rel="stylesheet" href="styles.css"> </head> <body> Hello World! I am formatted with a linked style sheet.	

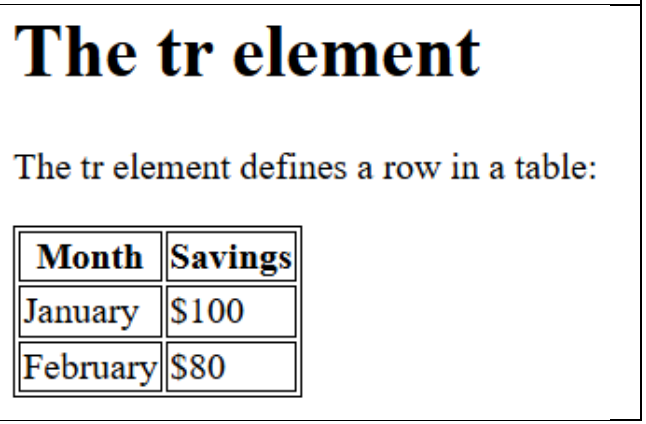
		Me too! </body> </html>	
<nav>	Defines navigation links	<!DOCTYPE html> <html> <body> The nav element The nav element defines a set of navigation links: <nav> HTML | CSS | JavaScript | Python </nav> </body> </html>	

Table

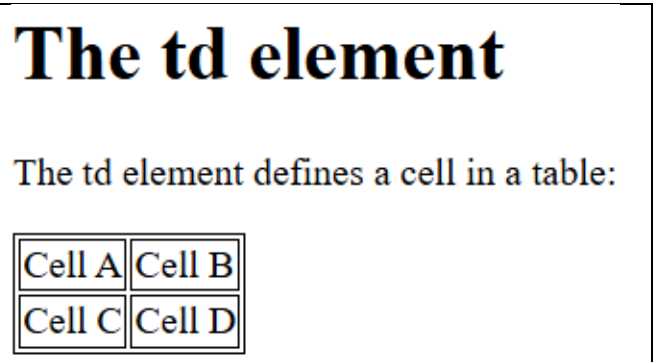
Tag	Def	Code Example	Browser View
<table>	Defines a table An HTML table consists of one <table> element and one or more <tr>, <th>, and <td> elements.	<html> <head> <style> table, th, td { border: 1px solid black; } </style> </head> <body> The table element <table> <tr> <th>Month</th> <th>Savings</th> </tr> <tr> <td>January</td> <td>$100</td> </tr> <tr> <td>February</td>	The table element Month | Savings January | $100 February | $80

		<td>$80</td> </tr> </table> </body> </html>	
<caption>	Defines a table caption	<!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } </style> </head> <body> The caption element <table> <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> </body> </html>	
<th>	Defines a header cell in a table The <th> tag defines a header cell in an HTML table.	<!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } </style> </head> <body> The th element The th element defines a header cell in a table:	

		<table> <tr> <th>Month</th> <th>Savings</th> </tr> <tr> <td>January</td> <td>$100</td> </tr> <tr> <td>February</td> <td>$80</td> </tr> </table> </body> </html>	
<tr>	Defines a row in a table The <tr> tag defines a row in an HTML table. A <tr> element contains one or more <th> or <td> elements.	<!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } </style> </head>	

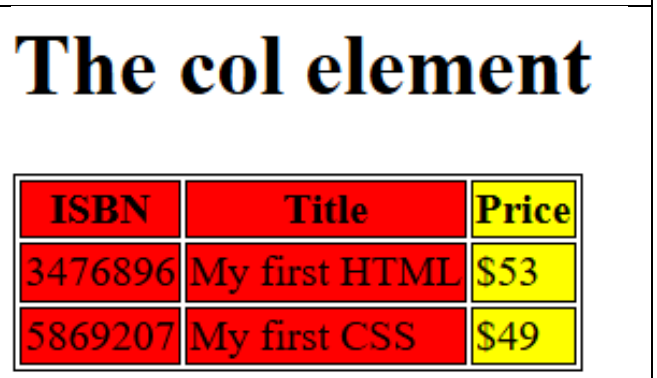
		<body> The tr element The tr element defines a row in a table: <table> <tr> <th>Month</th> <th>Savings</th> </tr> <tr> <td>January</td> <td>$100</td> </tr> <tr> <td>February</td> <td>$80</td> </tr> </table> </body> </html>	

<td>	Defines a cell in a table The <td> tag defines a standard data cell in an HTML table. An HTML table has two kinds of cells: • Header cells - contains header information (created with the <th> element) • Data cells - contains data (created with the <td> element)	<!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } </style> </head> <body> The td element The td element defines a cell in a table: <table> <tr> <td>Cell A</td> <td>Cell B</td> </tr> <tr> <td>Cell C</td> <td>Cell D</td> </tr> </table> </body> </html>	

<thead>	Groups the header content in a table	<!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } </style> </head> <body> The thead, tbody, and tfoot elements <table> <thead> <tr> <th>Month</th> <th>Savings</th> </tr> </thead> <tbody> <tr> <td>January</td> <td>$100</td> </tr> <tr>	**The thead, tbody, and tfoot elements** **Month** **Savings** January $100 February $80 Sum $180

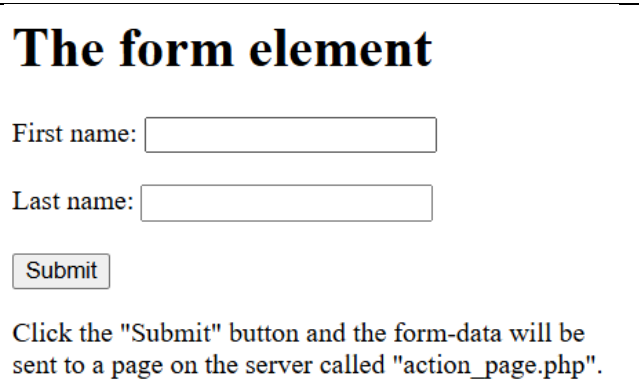
		<td>February</td> <td>$80</td> </tr> </tbody> <tfoot> <tr> <td>Sum</td> <td>$180</td> </tr> </tfoot> </table> </body> </html>	
<tbody>	Groups the body content in a table		
<tfoot>	Groups the footer content in a table		
<colgroup>	Specifies a group of one or more columns in a table for formatting	<!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } </style>	**The colgroup element** ISBN | Title | Price 3476896 | My first HTML | $53 5869207 | My first CSS | $49

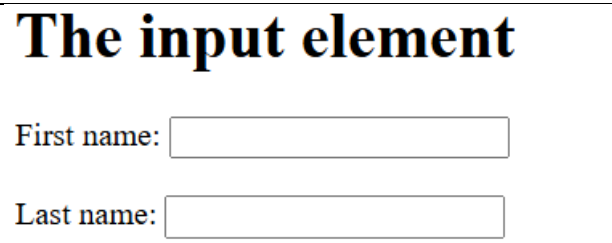
```
</head>
<body>

<h1>The colgroup element</h1>

<table>
 <colgroup>
  <col span="2" style="background-color:red">
  <col style="background-color:yellow">
 </colgroup>
 <tr>
  <th>ISBN</th>
  <th>Title</th>
  <th>Price</th>
 </tr>
 <tr>
  <td>3476896</td>
  <td>My first HTML</td>
  <td>$53</td>
 </tr>
 <tr>
  <td>5869207</td>
  <td>My first CSS</td>
  <td>$49</td>
 </tr>
</table>
```

		</body> </html>	
<col>	Specifies column properties for each column within a <colgroup> element	<!DOCTYPE html> <html> <head> <style> table, th, td { border: 1px solid black; } </style> </head> <body> The col element <table> <colgroup> <col span="2" style="background-color:red"> <col style="background-color:yellow"> </colgroup> <tr> <th>ISBN</th> <th>Title</th> <th>Price</th> </tr>	

		<tr> <td>3476896</td> <td>My first HTML</td> <td>$53</td> </tr> <tr> <td>5869207</td> <td>My first CSS</td> <td>$49</td> </tr> </table> </body> </html>	

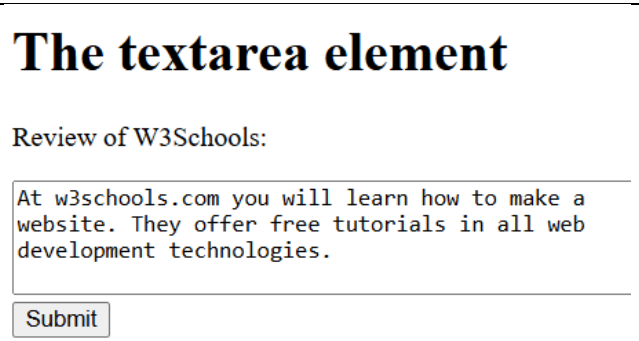
Form

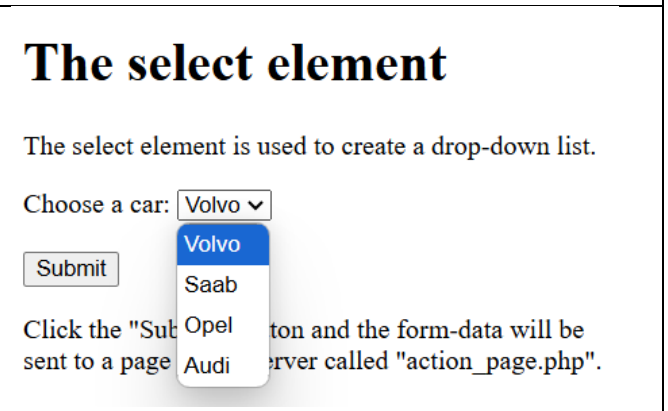
Tag	Def	Code Example	Browser View
<form>	Defines an HTML form for user input The <form> tag is used to create an HTML form for user input. The <form> element can contain one or more of the following form elements: • <input> • <textarea> • <button> • <select> • <option> • <optgroup> • <fieldset> • <label> • <output>	<!DOCTYPE html> <html> <body> The form element <form> <label for="fname">First name:</label> <input type="text" id="fname" name="fname"> <label for="lname">Last name:</label> <input type="text" id="lname" name="lname"> <input type="submit" value="Submit"> </form> Click the "Submit" button and the form-data will be sent to a page on the server called "action_page.php". </body> </html>	

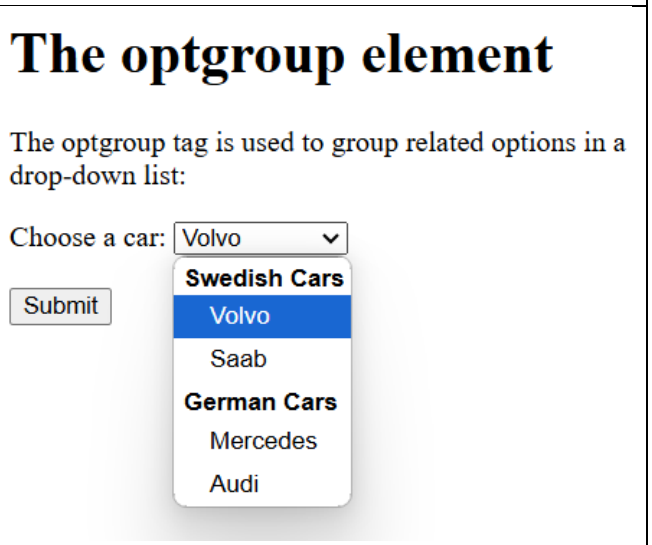
<input>	• Defines an input control • The <input> tag specifies an input field where the user can enter data. • The <input> element is the most important form element. • The <input> element can be displayed in several ways, depending on the type attribute. • The different input types are as follows: • <input type="button">	<!DOCTYPE html> <html> <body> The input element <form action="/action_page.php"> <label for="fname">First name:</label> <input type="text" id="fname" name="fname"> <label for="lname">Last name:</label> <input type="text" id="lname" name="lname"> <input type="submit" value="Submit"> </form> Click the "Submit" button and the form-data will be sent to a page on the server called "action_page.php". </body> </html>	

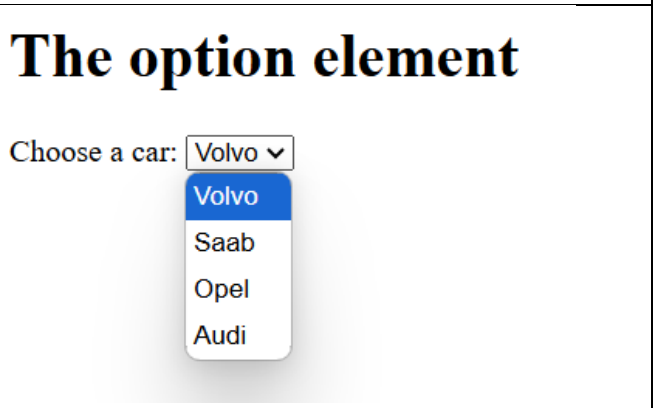
	• <input type="checkbox"> • <input type="color"> • <input type="date"> • <input type="datetime-local"> • <input type="email"> • <input type="file"> • <input type="hidden"> • <input type="image"> • <input type="month"> • <input type="number"> • <input type="password"> • <input type="radio">		

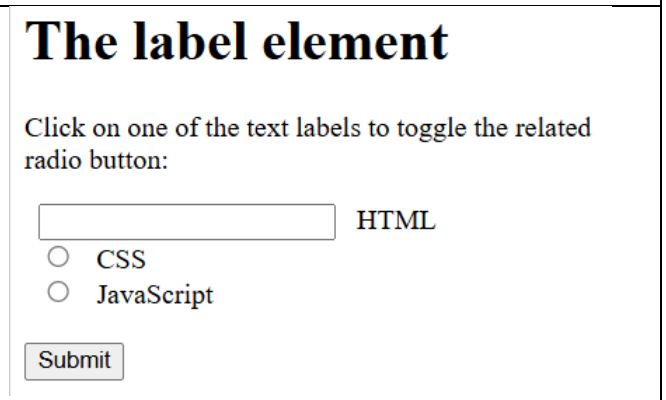
	• <input type="range"> • <input type="reset"> • <input type="search"> • <input type="submit"> • <input type="tel"> • <input type="text"> (default value) • <input type="time"> • <input type="url"> • <input type="week"> • Look at the type attribute to see examples for each input type!		

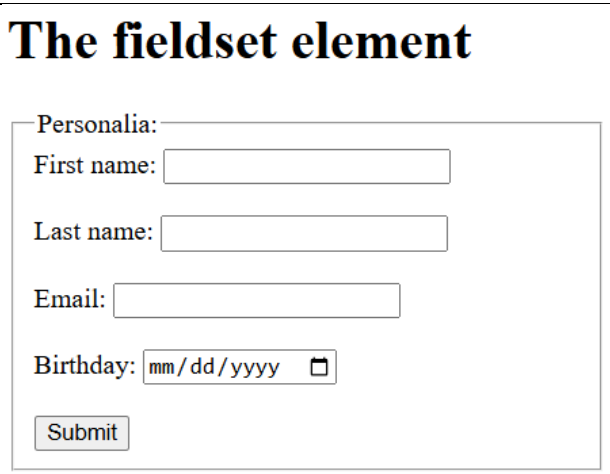
<textarea>	Defines a multiline input control (text area)	<!DOCTYPE html> <html> <body> The textarea element <form action="/action_page.php"> <label for="w3review"> Review of W3Schools:</label> <textarea id="w3review" name="w3review" rows="4" cols="50"> At w3schools.com you will learn how to make a website. They offer free tutorials in all web development technologies. </textarea> <input type="submit" value="Submit"> </form> Click the "Submit" button and the form-data will be sent to a page on the server called "action_page.php". </body> </html>	

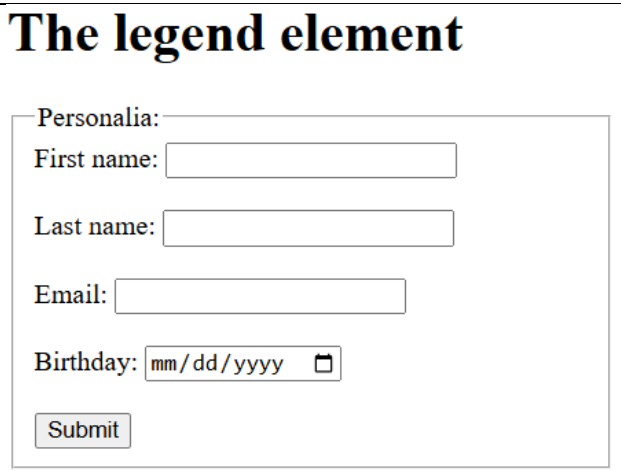
<button>	Defines a clickable button	<!DOCTYPE html> <html> <body> The button Element <button type="button" onclick="alert('Hello world!')"> Click Me! </button> </body> </html>	
<select>	Defines a drop-down list The <select> element is used to create a drop-down list. The <select> element is most often used in a form, to collect user input.	<!DOCTYPE html> <html> <body> The select element The select element is used to create a drop-down list. <form action="/action_page.php"> <label for="cars">Choose a car:</label> <select name="cars" id="cars"> <option value="volvo">Volvo</option> <option value="saab">Saab</option>	

		<option value="opel">Opel</option> <option value="audi">Audi</option> </select> </form> Click the "Submit" button and the form-data will be sent to a page on the server called "action_page.php". </body> </html>	
<optgroup>	Defines a group of related options in a drop-down list	<!DOCTYPE html> <html> <body> The optgroup element The optgroup tag is used to group related options in a drop-down list: <form action="/action_page.php"> <label for="cars">Choose a car:</label> <select name="cars" id="cars"> <optgroup label="Swedish Cars">	

		<option value="volvo">Volvo</option> <option value="saab">Saab</option> </optgroup> <optgroup label="German Cars"> <option value="mercedes">Mercedes</option> <option value="audi">Audi</option> </optgroup> </select> <input type="submit" value="Submit"> </form> </body> </html>	
<option>	Defines an option in a drop-down list	<!DOCTYPE html> <html> <body> The option element <label for="cars">Choose a car:</label> <select id="cars"> <option value="volvo">Volvo</option> <option value="saab">Saab</option> <option value="opel">Opel</option>	

		<option value="audi">Audi</option> </select> </body> </html>	
<label>	Defines a label for an <input> element	<!DOCTYPE html> <html> <body> The label element Click on one of the text labels to toggle the related radio button: <form action="/action_page.php"> <input type="text" id="html" name="fav_language"> <label for="html">HTML</label> <input type="radio" id="css" name="fav_language" value="CSS"> <label for="css">CSS</label> <input type="radio" id="javascript" name="fav_language" value="JavaScript">	

		<label for="javascript">JavaScript</label> <input type="submit" value="Submit"></form> </body> </html>	
<fieldset>	Groups related elements in a form	<!DOCTYPE html> <html> <body> The fieldset element <form action="/action_page.php"> <fieldset> <legend>Personalia:</legend> <label for="fname">First name:</label> <input type="text" id="fname" name="fname"> <label for="lname">Last name:</label> <input type="text" id="lname" name="lname"> <label for="email">Email:</label> <input type="email" id="email" name="email"> <label for="birthday">Birthday:</label>	

		<input type="date" id="birthday" name="birthday"> <input type="submit" value="Submit"> </fieldset> </form> </body> </html>	
<legend>	Defines a caption for a <fieldset> element	<!DOCTYPE html> <html> <body> The legend element <form action="/action_page.php"> <fieldset> <legend>Personalia:</legend> <label for="fname">First name:</label> <input type="text" id="fname" name="fname"> <label for="lname">Last name:</label> <input type="text" id="lname" name="lname"> <label for="email">Email:</label> <input type="email" id="email" name="email"> <label for="birthday">Birthday:</label>	

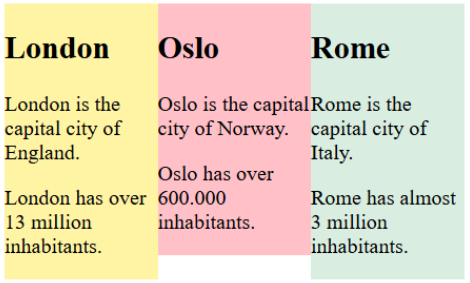
		<input type="date" id="birthday" name="birthday"> <input type="submit" value="Submit"> </fieldset> </form> </body> </html>	

DIV with CSS

Tag	Def	Code Example	Browser View
** **	The element is by default a block element, meaning that it takes all available width, and comes with line breaks before and after.	<!DOCTYPE html> <html> <style> div { background-color: #FFF4A3; } </style> <body> HTML DIV Example Lorem Ipsum I am a div dolor sit amet. The yellow background is added to demonstrate the footprint of the DIV element. </body> </html>	**HTML DIV Example** Lorem Ipsum I am a div dolor sit amet. The yellow background is added to demonstrate the footprint of the DIV element.
	** as a container** The element is often used to group sections of a web page together.	<!DOCTYPE html> <html> <style> div { background-color: #FFF4A3; } </style> <body>	**HTML DIV Example** **London** London is the capital city of England. London has over 13 million inhabitants. The yellow background is added to demonstrate the footprint of the DIV element.

		HTML DIV Example London London is the capital city of England. London has over 13 million inhabitants. The yellow background is added to demonstrate the footprint of the DIV element. </body> </html>	
	Center align a element If you have a `<div>` element that is not 100% wide, and you want to center-align it, set the CSS `margin` property to `auto`.	<!DOCTYPE html> <html> <style> div { width: 300px; margin: auto; background-color: #FFF4A3; } </style> <body> Center align a DIV element	**Center align a DIV element** **London** London is the capital city of England. London has over 13 million inhabitants.

		 London London is the capital city of England. London has over 13 million inhabitants. </body> </html>	
	Multiple elements You can have many containers on the same page.	<!DOCTYPE html> <html> <body> Multiple DIV Elements London London is the capital city of England. London has over 13 million inhabitants. Oslo Oslo is the capital city of Norway. Oslo has over 600.000 inhabitants. Rome Rome is the capital city of Italy. Rome has almost 3 million inhabitants. 	**Multiple DIV Elements** **London** London is the capital city of England. London has over 13 million inhabitants. **Oslo** Oslo is the capital city of Norway. Oslo has over 600.000 inhabitants. **Rome** Rome is the capital city of Italy. Rome has almost 3 million inhabitants. CSS styles are added to make it easier to separate the divs, and to make them more pretty:)

		CSS styles are added to make it easier to separate the divs, and to make them more pretty:) </body> </html>	
	Aligning elements side by side When building web pages, you often want to have two or more `<div>` elements side by side, like this:		
	Float The CSS `float` property was not originally meant to align `<div>` elements side-by-side, but has been used for this purpose for many years. The CSS `float` property is used for positioning and formatting content and allow elements float next to each other instead of on top of each other.	<!DOCTYPE html> <html> <style> div.mycontainer { width:100%; overflow:auto; } div.mycontainer div { width:33%; float:left; } </style> <body> London London is the capital city of England.	

<table>
<tr>
<td></td>
<td></td>
<td>

```
  <p>London has over 13 million inhabitants.</p>
</div>

<div style="background-color:#FFC0C7;">
  <h2>Oslo</h2>
  <p>Oslo is the capital city of Norway.</p>
  <p>Oslo has over 600.000 inhabitants.</p>
</div>

<div style="background-color:#D9EEE1;">
  <h2>Rome</h2>
  <p>Rome is the capital city of Italy.</p>
  <p>Rome has almost 3 million inhabitants.</p>
</div>

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

</td>
<td></td>
</tr>
<tr>
<td></td>
<td>

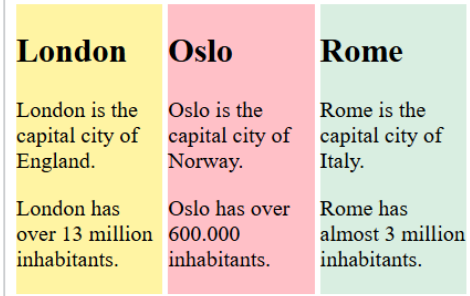
Inline-block

If you change the `<div>` element's `display` property from `block` to `inline-block`, the `<div>` elements will no longer add a line break before and after, and will be displayed side by side instead of on top of each other.

</td>
<td>

```
<!DOCTYPE html>
<html>
<style>
div {
 width:30%;
 display:inline-block;
}
</style>
<body>
```

</td>
<td>



</td>
</tr>
</table>

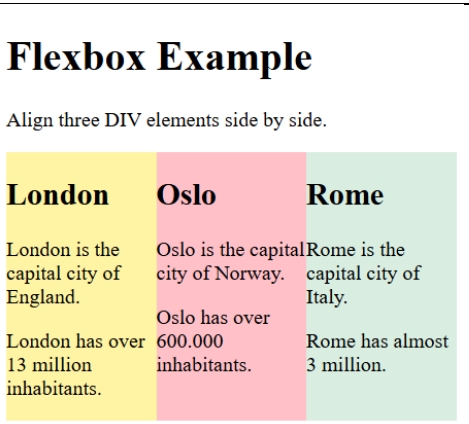
		 London London is the capital city of England. London has over 13 million inhabitants. Oslo Oslo is the capital city of Norway. Oslo has over 600.000 inhabitants. Rome Rome is the capital city of Italy. Rome has almost 3 million inhabitants. </body> </html>	

Flex

The CSS Flexbox Layout Module was introduced to make it easier to design flexible responsive layout structure without using float or positioning.

To make the CSS flex method work, surround the `<div>` elements with another `<div>` element and give it the status as a flex container.

```
<!DOCTYPE html>
<html>
<head>
<style>
.mycontainer {
  display: flex;
}
.mycontainer > div {
  width:33%;
}
</style>
</head>
<body>

<h1>Flexbox Example</h1>

<p>Align three DIV elements side by side.</p>

<div class="mycontainer">

  <div style="background-color:#FFF4A3;">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>London has over 13 million inhabitants.</p>
  </div>

  <div style="background-color:#FFC0C7;">
    <h2>Oslo</h2>
    <p>Oslo is the capital city of Norway.</p>
    <p>Oslo has over 600.000 inhabitants.</p>
```



		 Rome Rome is the capital city of Italy. Rome has almost 3 million. </body> </html>	
	Grid The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning. Sounds almost the same as flex, but has the ability to define more than one row and position each row individually. The CSS grid method requires that you surround the `<div>` elements with another `<div>` element and give the status as a grid container, and you	<!DOCTYPE html> <html> <head> <style> .grid-container { display: grid; grid-template-columns: 33% 33% 33%; } </style> </head> <body> Grid Example Align three DIV elements side by side.	**Grid Example** Align three DIV elements side by side. **London** London is the capital city of England. London has over 13 million inhabitants. **Oslo** Oslo is the capital city of Norway. Oslo has over 600.000 inhabitants. **Rome** Rome is the capital city of Italy. Rome has almost 3 million inhabitants.

	must specify the width of each column.	 London London is the capital city of England. London has over 13 million inhabitants. Oslo Oslo is the capital city of Norway. Oslo has over 600.000 inhabitants. Rome Rome is the capital city of Italy. Rome has almost 3 million inhabitants. </body> </html>	

CSS Introduction

CSS is the language we use to style a Web page.

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

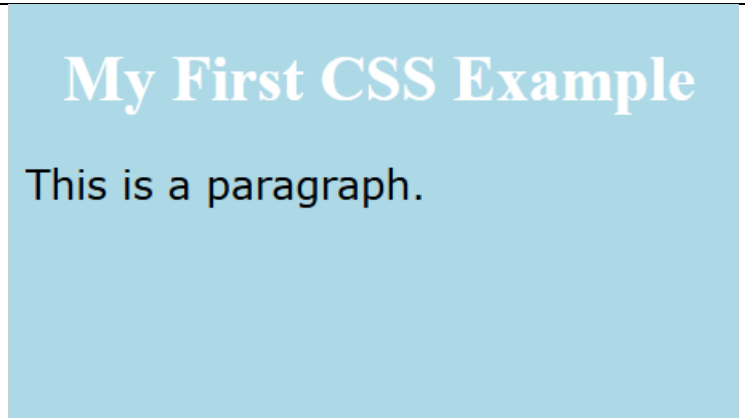
Why Use CSS?

- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Example:

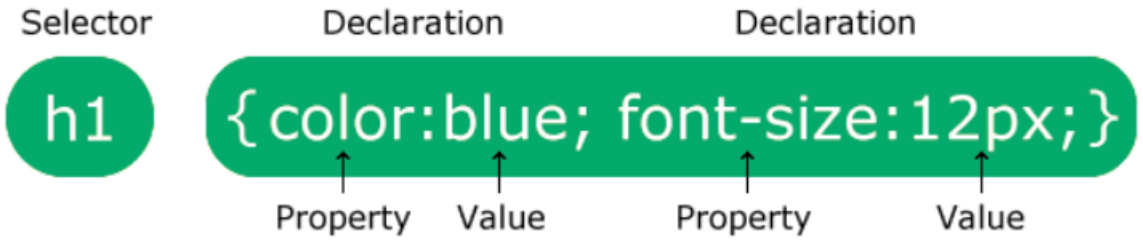
Code	Browser View

```
<!DOCTYPE html>
<html>
<head>
<style>

body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
```



This is a paragraph. </body> </html>	

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.

Example

In this example all <p> elements will be center-aligned, with a red text color:

```
p {
  color: red;
  text-align: center;
}
```

Example Explained

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

Code	Browser View
<!DOCTYPE html> <html> <head> <style> p { color: red; text-align: center; } </style> </head>	Hello World! These paragraphs are styled with CSS.

<body> Hello World! These paragraphs are styled with CSS. </body> </html>	

CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

The CSS element Selector

Example

Here, all elements on the page will be center-aligned, with a red text color:

```
p {
  text-align: center;
  color: red;
}
```

Code	Browser View
<!DOCTYPE html> <html> <head> <style> p { text-align: center; color: red; } </style> </head> <body> Every paragraph will be affected by the style. Me too! And me! </body> </html>	Every paragraph will be affected by the style. Me too! And me!

The CSS id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1 {
  text-align: center;
  color: red;
}
```

Code	Browser View
<!DOCTYPE html> <html> <head> <style> #para1 { text-align: center; color: red;	Hello World! This paragraph is not affected by the style.

} </style> </head> <body> Hello World! This paragraph is not affected by the style. </body> </html>	

The CSS class Selector

- The class selector selects HTML elements with a specific class attribute.

- To select elements with a specific class, write a period (.) character, followed by the class name.

Example1

In this example all HTML elements with class="center" will be red and center-aligned:

```
.center {
  text-align: center;
  color: red;
}
```

Code	Browser View
<!DOCTYPE html> <html> <head> <style> .center { text-align: center; color: red; } </style> </head> <body> <h1 class="center">Red and center-aligned heading</h1> Red and center-aligned paragraph. </body> </html>	

Example2

In this example only elements with class="center" will be red and center-aligned:

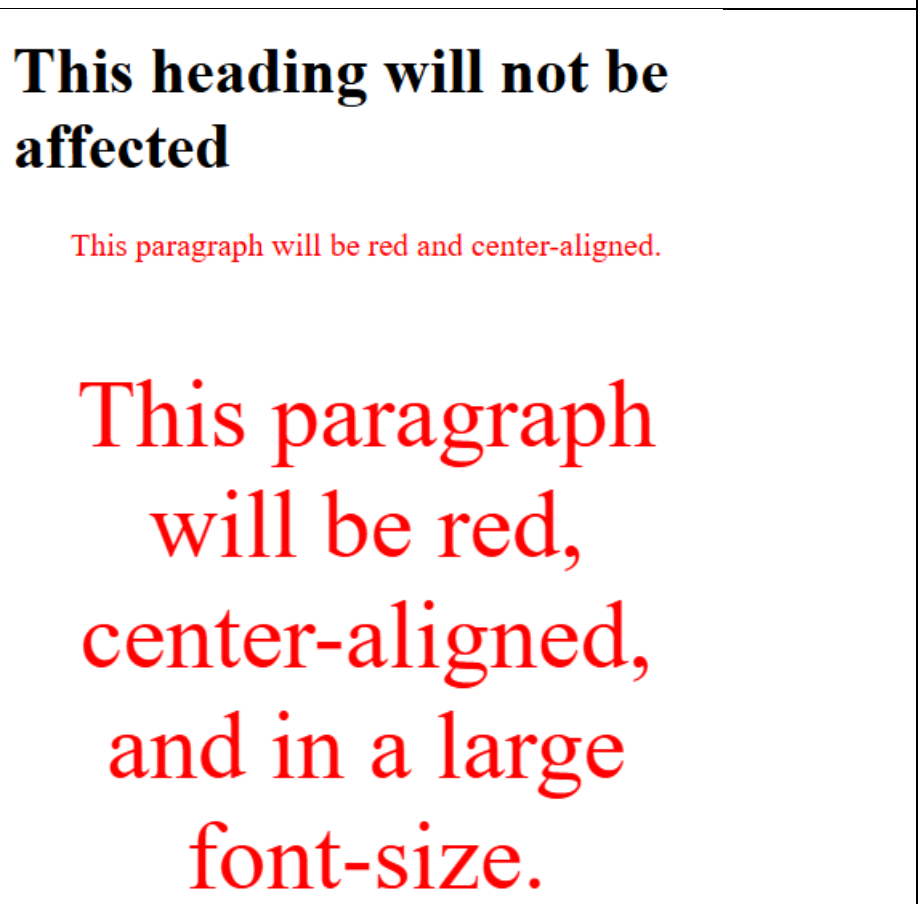
```
p.center {
  text-align: center;
  color: red;
}
```

Code	Browser View
<!DOCTYPE html> <html> <head> <style> p.center { text-align: center; color: red; } </style> </head> <body> <h1 class="center">This heading will not be affected</h1> This paragraph will be red and center-aligned. </body> </html>	

Example3

In this example the element will be styled according to class="center" and to class="large":

```
<p class="center large">This paragraph refers to two classes.</p>
```

Code	Browser View
<!DOCTYPE html> <html> <head> <style> p.center { text-align: center; color: red; } p.large { font-size: 300%; } </style> </head> <body> <h1 class="center">This heading will not be affected</h1> This paragraph will be red and center-aligned. This paragraph will be red, center-aligned, and in a large font-size. </body> </html>	

The CSS Universal Selector

- The universal selector (*) selects all HTML elements on the page.

Example

The CSS rule below will affect every HTML element on the page:

```
* {
  text-align: center;
  color: blue;
}
```

Code	Browser View
<!DOCTYPE html> <html> <head> <style> * { text-align: center; color: blue; } </style> </head> <body>	

Hello world! Every element on the page will be affected by the style. Me too! And me! </body> </html>	

The CSS Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions.

- Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
```

```
  color: red;
}
```

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

Code	Browser View
<!DOCTYPE html> <html> <head> <style> h1, h2, p { text-align: center; color: red; } </style> </head> <body>	

Hello World! Smaller heading! This is a paragraph. </body> </html>	

How To Add CSS

When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet.

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- Inline CSS
- Internal CSS
- External CSS

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.

Example

Inline styles are defined within the "style" attribute of the relevant element:

```
<!DOCTYPE html>
<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>
```

Code	Browser View
<!DOCTYPE html> <html> <body> This is a heading This is a paragraph. </body> </html>	

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:

```
<!DOCTYPE html>
<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>
```

Code	Browser View
<!DOCTYPE html> <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>
```

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:

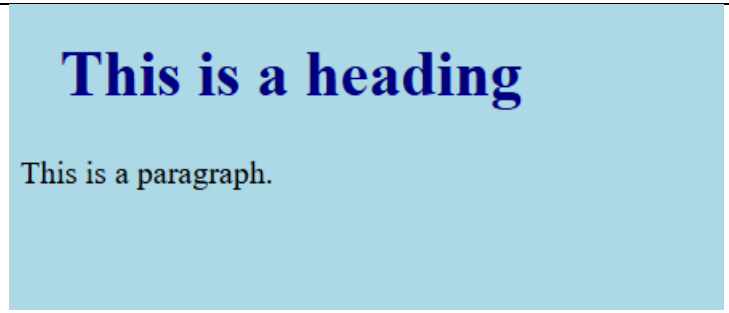
```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
```

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

</body>
</html>
```

- An external style sheet can be written in any text editor, and must be saved with a .css extension.
- The external .css file should not contain any HTML tags.
- Here is how the "mystyle.css" file looks:

"mystyle.css"

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

h1 {
  color: navy;
  margin-left: 20px;
}
```

Code	Browser View
<!DOCTYPE html> <html> <head> <link rel="stylesheet" href="mystyle.css"> </head> <body>	

This is a heading This is a paragraph. </body> </html>	

Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

Assume that an **external style sheet** has the following style for the <h1> element:

```
h1 {
  color: navy;
}
```

Then, assume that an **internal style sheet** also has the following style for the <h1> element:

```
h1 {
  color: orange;
}
```

Example

If the internal style is defined **after** the link to the external style sheet, the <h1> elements will be "orange":

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

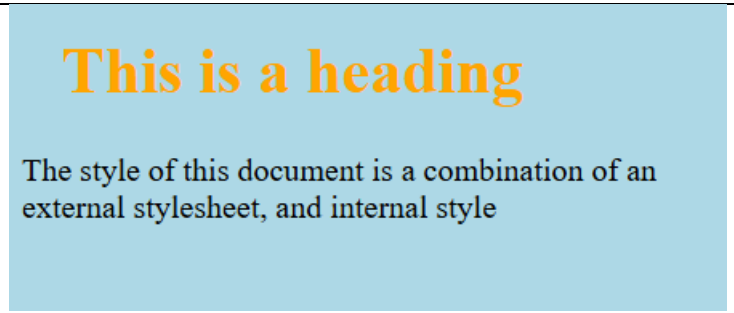
Try it Yourself »

Example

However, if the internal style is defined **before** the link to the external style sheet, the <h1> elements will be "navy":

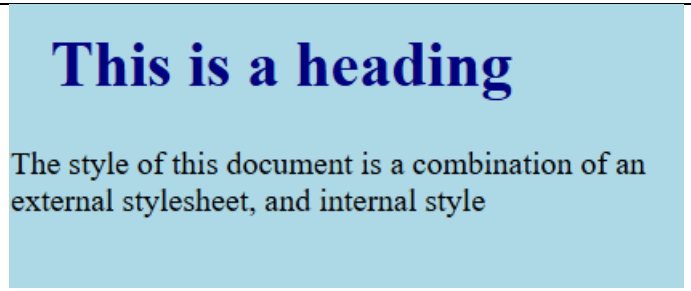
```
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Try it Yourself »

Code	Browser View
If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange":	
<!DOCTYPE html> <html> <head> <link rel="stylesheet" type="text/css" href="mystyle.css"> <style> h1 { color: orange; } </style> </head> <body> This is a heading The style of this document is a combination of an external stylesheet, and internal style </body> </html>	
However, if the internal style is defined before the link to the external style sheet, the <h1> elements will be "navy":	

<table>
<tr>
<td>

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>The style of this document is a combination of an
external stylesheet, and internal style</p>

</body>
</html>
```

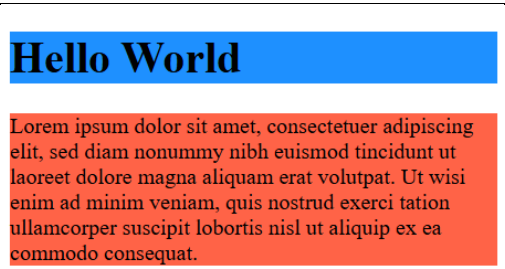
</td>
<td>


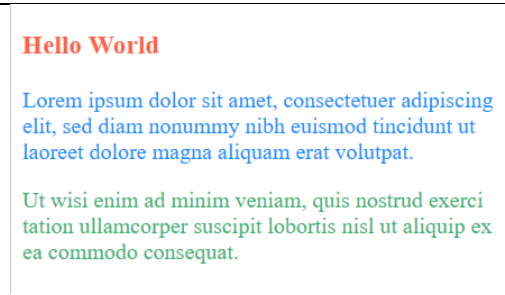
</td>
</tr>
<tr>
<td>Comment:</td>
<td>

```
<style>
p {
  color: red; /* Set text color to red */
}
</style>
```

</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</table>

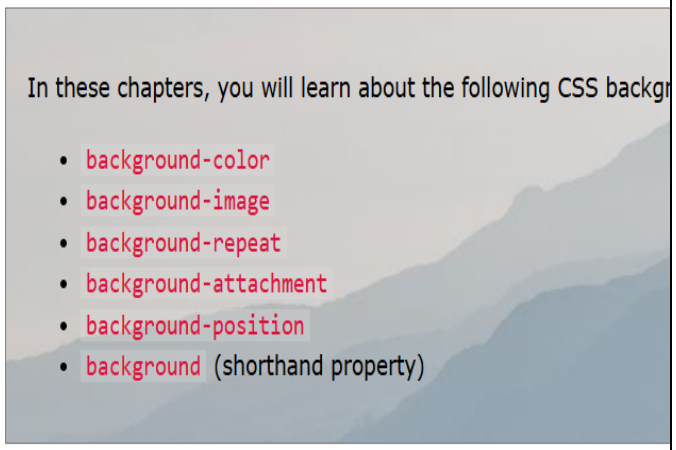
	Code	Browser View
Color Names	<!DOCTYPE html> <html> <body> Tomato Orange DodgerBlue MediumSeaGreen Gray SlateBlue Violet LightGray </body> </html>	

CSS Background Color	<!DOCTYPE html> <html> <body> Hello World Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. </body> </html>	

CSS Text Color	<!DOCTYPE html> <html> <body> Hello World Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. </body> </html>	

CSS Border Color	<!DOCTYPE html> <html> <body> Hello World Hello World Hello World </body> </html>	

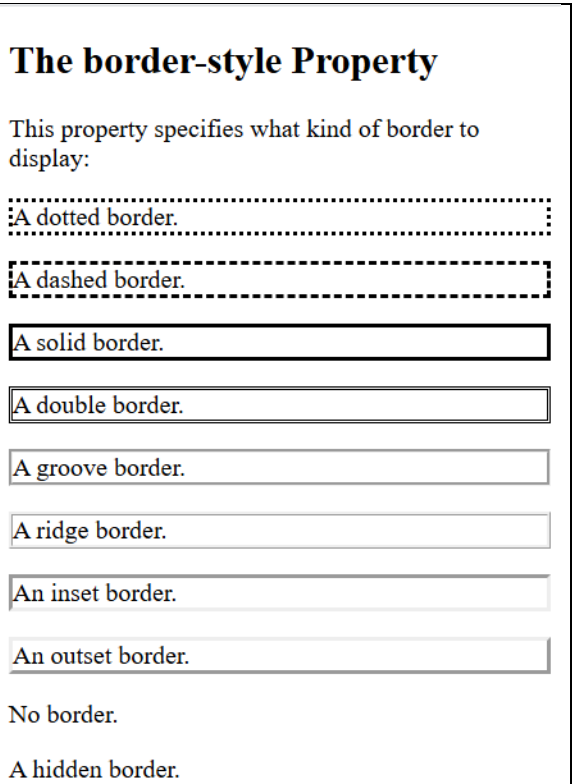
CSS Color Values	<!DOCTYPE html> <html> <body> Same as color name "Tomato": rgb(255, 99, 71) #ff6347 hsl(9, 100%, 64%) Same as color name "Tomato", but 50% transparent: rgba(255, 99, 71, 0.5) hsla(9, 100%, 64%, 0.5) In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values. </body> </html>	Same as color name "Tomato": **rgb(255, 99, 71)** **#ff6347** **hsl(9, 100%, 64%)** Same as color name "Tomato", but 50% transparent: **rgba(255, 99, 71, 0.5)** **hsla(9, 100%, 64%, 0.5)** In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.

CSS Backgrounds	``` body { background-color: lightblue; } h1 { background-color: green; } div { background-color: lightblue; } p { background-color: yellow; } div { background-color: green; opacity: 0.3; } div { background: rgba(0, 128, 0, 0.3) /* Green background with 30% opacity */ } ``` **IMAGES** **CSS background-image**	The CSS background properties are used to add background e 

	`` body { background-image: url("paper.gif"); } body { background-image: url("bgdesert.jpg"); } p { background-image: url("paper.gif"); } ```	

	Code	Browser View
CSS Borders	# CSS Border Style The `border-style` property specifies what kind of border to display. The following values are allowed: • `dotted` - Defines a dotted border • `dashed` - Defines a dashed border • `solid` - Defines a solid border • `double` - Defines a double border • `groove` - Defines a 3D grooved border. The effect depends on the border-color value • `ridge` - Defines a 3D ridged border. The effect depends on the border-color value • `inset` - Defines a 3D inset border. The effect depends on the border-color value • `outset` - Defines a 3D outset border. The effect depends on the border-color value • `none` - Defines no border • `hidden` - Defines a hidden border The `border-style` property can have from one to four values (for the top border, right border, bottom border, and the left border).	

```
<!DOCTYPE html>
<html>
<head>
<style>
        p.dotted {border-style: dotted;}
        p.dashed {border-style: dashed;}
        p.solid {border-style: solid;}
        p.double {border-style: double;}
        p.groove {border-style: groove;}
        p.ridge {border-style: ridge;}
        p.inset {border-style: inset;}
        p.outset {border-style: outset;}
        p.none {border-style: none;}
        p.hidden {border-style: hidden;}
        p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>

<h2>The border-style Property</h2>
<p>This property specifies what kind of border to display:</p>

<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
```



	A groove border. A ridge border. An inset border. An outset border. No border. A hidden border. A mixed border. </body> </html>	

	Code	Browser View
CSS Border Width	p.one { border-style: solid; border-width: 5px; } p.two { border-style: solid; border-width: medium; } p.three { border-style: dotted; border-width: 2px;	

	} p.four { border-style: dotted; border-width: thick; }	
	p.one { border-style: solid; border-width: 5px 20px; /* 5px top and bottom, 20px on the sides */ } p.two { border-style: solid; border-width: 20px 5px; /* 20px top and bottom, 5px on the sides */ } p.three { border-style: solid; border-width: 25px 10px 4px 35px; /* 25px top, 10px right, 4px bottom and 35px left */ }	

CSS Border Color		
	p.one { border-style: solid; border-color: red; } p.two { border-style: solid; border-color: green; } p.three { border-style: dotted; border-color: blue; }	

CSS Margins

CSS Margins

The CSS `margin` properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

All the margin properties can have the following values:

- auto - the browser calculates the margin
- *length* - specifies a margin in px, pt, cm, etc.
- % - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

Tip: Negative values are allowed.

```
<style>
div {
  margin: 70px;
  border: 1px solid #4CAF50;
}
</style>
```

CSS Margins

This element has a margin of 70px.

Margin - Individual Sides	CSS has properties for specifying the margin for each side of an element: • `margin-top` • `margin-right` • `margin-bottom` • `margin-left` All the margin properties can have the following values: • auto - the browser calculates the margin • *length* - specifies a margin in px, pt, cm, etc. • *%* - specifies a margin in % of the width of the containing element • inherit - specifies that the margin should be inherited from the parent element **Tip:** Negative values are allowed.	

	p { margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px; } <style> div { border: 1px solid black; margin-top: 100px; margin-bottom: 100px; margin-right: 150px; margin-left: 80px; background-color: lightblue; } </style>	**Using individual margin properties** This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

Margin - Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property.

The `margin` property is a shorthand property for the following individual margin properties:

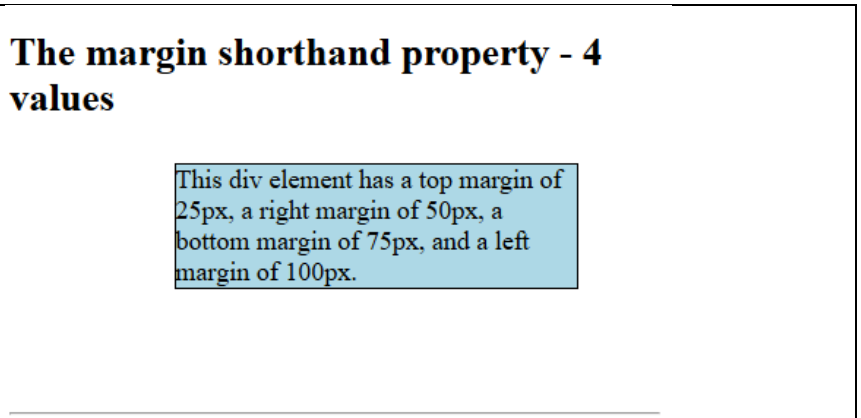
- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

So, here is how it works:

If the `margin` property has four values:

- **margin: 25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px

```
p {
  margin: 25px 50px 75px 100px;
}

<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  margin: 25px 50px 75px 100px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>The margin shorthand property - 4 values</h2>

<div>This div element has a top margin of 25px, a right margin of 50px, a bottom margin of 75px, and a left margin of 100px.</div>

<hr>

</body>
</html>
```

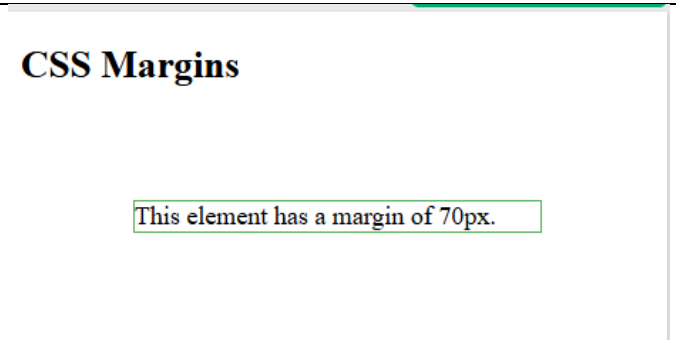


<table>
<tr>
<td>

bottom margin is 75px
left margin is 100px

</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  margin: 70px;
  border: 1px solid #4CAF50;
}
</style>
</head>
<body>

<h2>CSS Margins</h2>

<div>This element has a margin
of 70px.</div>

</body>
</html>
```

</td>
<td>


</td>
</tr>
</table>

CSS Padding

The CSS `padding` properties are used to generate space around an element's content, inside of any defined borders.

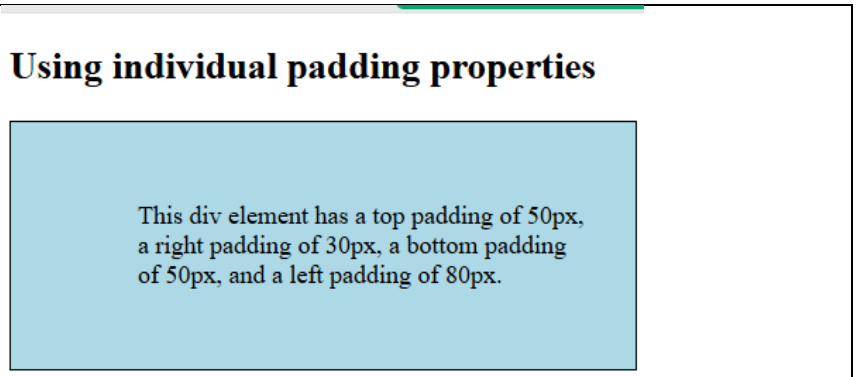
With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

- `padding-top`
- `padding-right`
- `padding-bottom`

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  background-color: lightblue;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
</head>
<body>

<h2>Using individual padding properties</h2>

<div>This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.</div>

</body>
</html>
```



• `padding-left` All the padding properties can have the following values: • *length* - specifies a padding in px, pt, cm, etc. • *%* - specifies a padding in % of the width of the containing element • inherit - specifies that the padding should be inherited from the parent element **Note:** Negative values are not allowed.		

Padding - Shorthand Property

To shorten the code, it is possible to specify all the padding properties in one property.

The `padding` property is a shorthand property for the following individual padding properties:

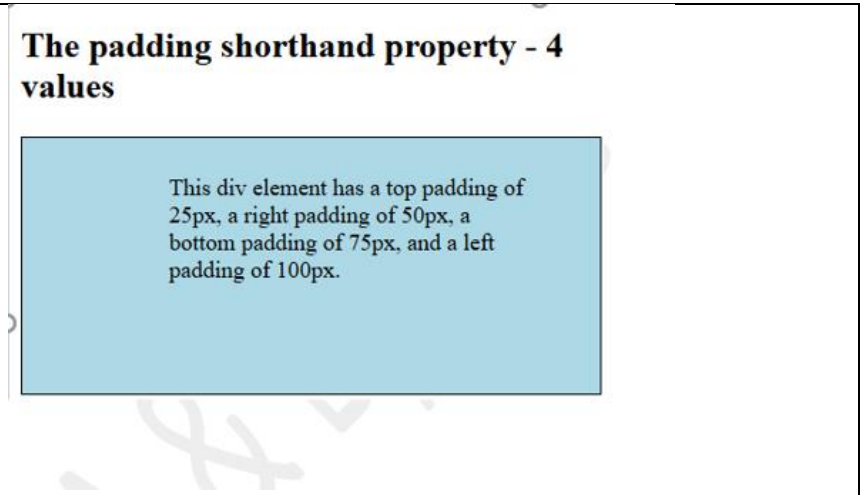
- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

So, here is how it works:

If the `padding` property has four values:

- **padding: 25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 1px solid black;
  padding: 25px 50px 75px 100px;
  background-color: lightblue;
}
</style>
</head>
<body>

<h2>The padding shorthand property - 4 values</h2>

<div>This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.</div>

</body>
</html>
```



○ bottom padding is 75px ○ left padding is 100px		

CSS Setting height and width

The `height` and `width` properties are used to set the height and width of an element.

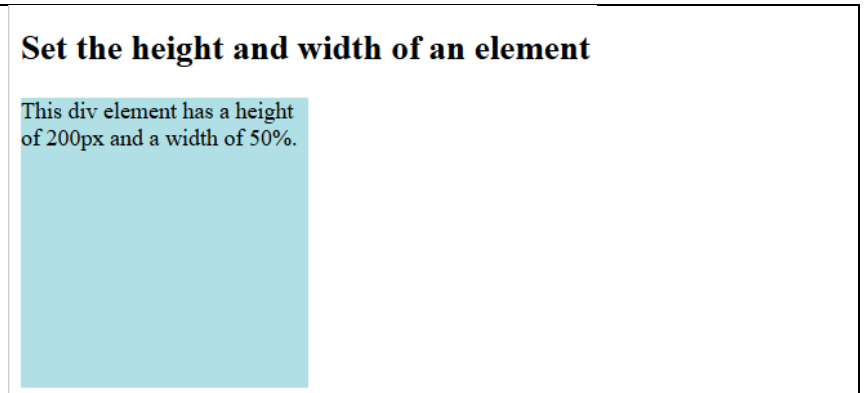
The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

CSS height and width Values

The `height` and `width` properties may have the following values:

- `auto` - This is default. The browser calculates

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the height and width of an element</h2>

<div>This div element has a height of 200px and a width of 50%.</div>

</body>
</html>
```

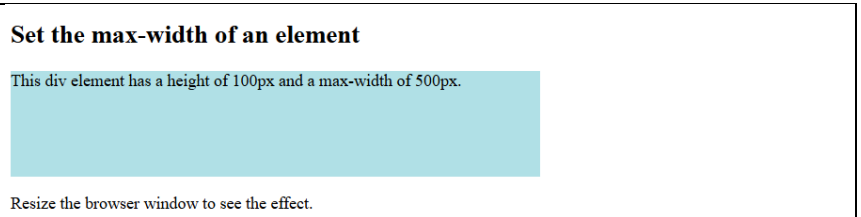


the height and width • `length` - Defines the height/width in px, cm, etc. • `%` - Defines the height/width in percent of the containing block • `initial` - Sets the height/width to its default value • `inherit` - The height/width will be inherited from its parent value		

Setting max-width

The `max-width` property is used to set the maximum width of an element.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  max-width: 500px;
  height: 100px;
  background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the max-width of an element</h2>

<div>This div element has a height of 100px and a max-width of 500px.</div>

<p>Resize the browser window to see the effect.</p>

</body>
</html>
```



The CSS Box Model

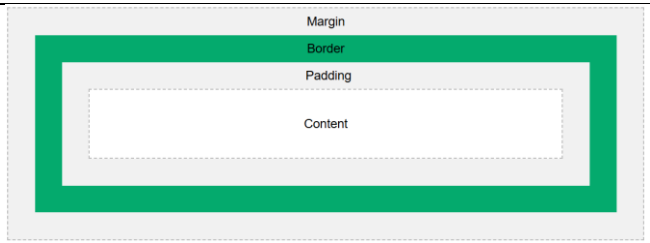
In CSS, the term "box model" is used when talking about design and layout.

Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: lightgrey;
  width: 300px;
  border: 15px solid green;
  padding: 50px;
  margin: 20px;
}
</style>
</head>
<body>

<h2>Demonstrating the Box Model</h2>

<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>

<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
```



)emonstrating the Box Model

he CSS box model is essentially a box that wraps around every HTML element.
ıe actual content.

This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

elements, and to define space between elements.	nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. </body> </html>	

CSS Text	`body {` `  color: blue;` `}` `h1 {` `  color: green;` `}`	
CSS Text Alignment **Text Alignment and Text Direction** In this chapter you will learn about the following properties: • text-align • text-align-last • direction • unicode-bidi • vertical-align	`h1 {` `  text-align: center;` `}` `h2 {` `  text-align: left;` `}` `h3 {` `  text-align: right;` `}` `div {` `  text-align: justify;` `}`	

CSS Text Decoration

Text Decoration

In this chapter you will learn about the following properties:

- text-decoration-line
- text-decoration-color
- text-decoration-style
- text-decoration-thickness
- text-decoration

```
h1 {
  text-decoration-
line: overline;
}

h2 {
  text-decoration-line: line-
through;
}

h3 {
  text-decoration-
line: underline;
}

p {
  text-decoration-
line: overline underline;
}
```

CSS Fonts

Font Selection is Important

Choosing the right font has a huge impact on how the readers experience a website.

The right font can create a strong identity for your brand.

Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

```
.p1 {
  font-family: "Times New Roman", Times, serif;
}

.p2 {
  font-family: Arial, Helvetica, sans-serif;
}

.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
```

FONT STYLE

```
p.normal {
  font-style: normal;
}

p.italic {
  font-style: italic;
}

p.oblique {
  font-style: oblique;
}
```

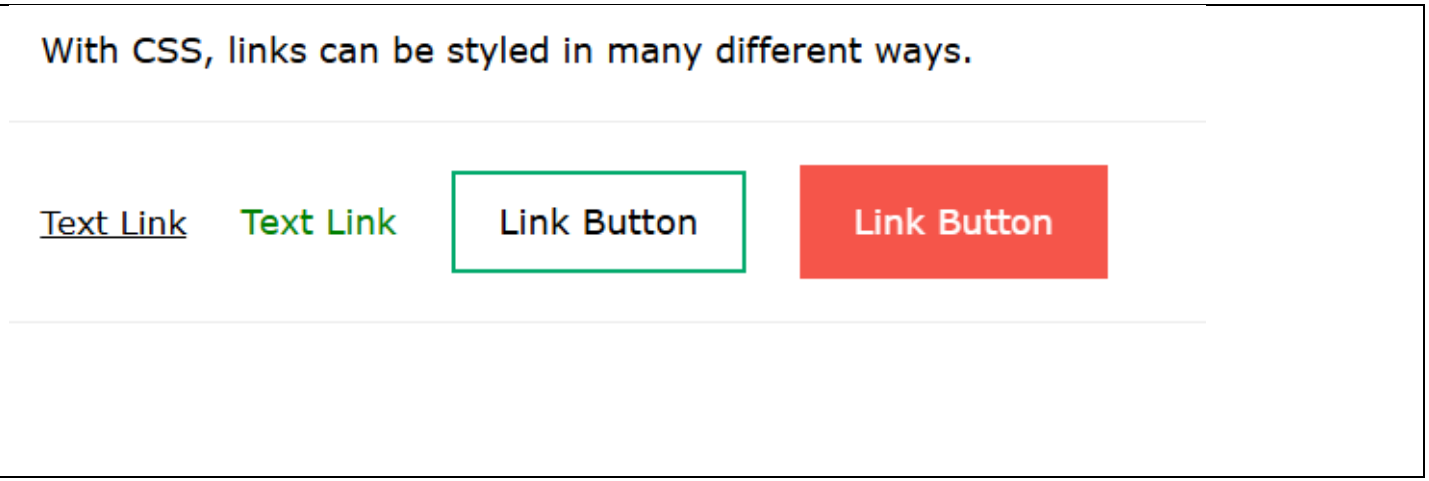
FONT SIZE

	h1 { font-size: 40px; } h2 { font-size: 30px; } p { font-size: 14px; }	

CSS Links

Styling Links

Links can be styled with any CSS property (e.g. `color`, `font-family`, `background`, etc.).



Styling Links

Links can be styled with any CSS property (e.g. `color`, `font-family`, `background`, etc.).

Example

```
a {
  color: hotpink;
}
```

Try it Yourself »

In addition, links can be styled differently depending on what **state** they are in.

The four links states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it

```
<!DOCTYPE html>
<html>
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
</style>
</head>
<body>
```

• a:active - a link the moment it is clicked	Styling a link depending on state This is a link Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective. Note: a:active MUST come after a:hover in the CSS definition in order to be effective. </body> </html>	

Text Decoration

The `text-decoration` property is mostly used to remove underlines from links:

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {
  text-decoration: none;
}

a:visited {
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

a:active {
  text-decoration: underline;
}
</style>
</head>
<body>

<h2>Styling a link with text-decoration property</h2>
```

Styling a link with text-decoration property

This is a link

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

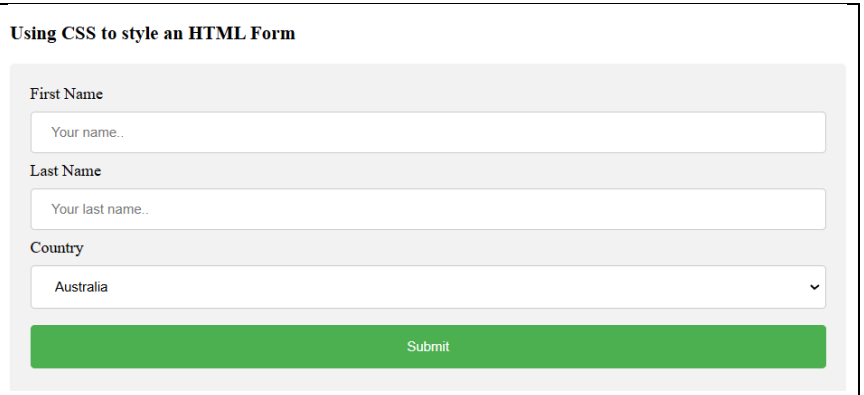
Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

	This is a link Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective. Note: a:active MUST come after a:hover in the CSS definition in order to be effective. </body> </html>	

CSS Forms

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
}
</style>
</head>
<body>

<h2>Padded input fields</h2>

<form>
  <label for="fname">First
Name</label>
  <input type="text" id="fname"
name="fname">
  <label for="lname">Last
Name</label>
  <input type="text" id="lname"
name="lname">
</form>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<style>
input[type=text], select {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 100%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type=submit]:hover {
  background-color: #45a049;
```



```
}

div {
  border-radius: 5px;
  background-color: #f2f2f2;
  padding: 20px;
}
</style>
<body>

<h3>Using CSS to style an HTML
Form</h3>

<div>
  <form
action="/action_page.php">
    <label for="fname">First
Name</label>
    <input type="text"
id="fname" name="firstname"
placeholder="Your name..">

    <label for="lname">Last
Name</label>
    <input type="text"
id="lname" name="lastname"
placeholder="Your last name..">

    <label
for="country">Country</label>
```

	<select id="country" name="country"> <option value="australia">Australia</option> <option value="canada">Canada</option> <option value="usa">USA</option> </select> <input type="submit" value="Submit"> </form> </body> </html>	

JavaScript Tutorial

JavaScript is the world's most popular programming language.

JavaScript is the programming language of the Web.

JavaScript is easy to learn.

This tutorial will teach you JavaScript from basic to advanced.

JavaScript Can Change HTML Content

One of many JavaScript HTML methods is `getElementById()`.

The example below "finds" an HTML element (with id="demo"), and changes the element content (innerHTML) to "Hello JavaScript":

Code	Browser View
<!DOCTYPE html> <html> <body> What Can JavaScript Do? JavaScript can change HTML content. <button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"'>Click Me!</button> </body> </html>	

```
<!DOCTYPE html>

<html>

<body>



<h2>What Can JavaScript Do?</h2>



<p>JavaScript can change HTML attribute values.</p>



<p>In this case JavaScript changes the value of the src (source) attribute of an image.</p>



<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the light</button>



<img id="myImage" src="pic_bulboff.gif" style="width:100px">
```

What Can JavaScript Do?

JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.

Turn on the light Turn off the light

<table>
<tr>
<td>

```
<button
onclick="document.getElementById('myImage').src
='pic_bulboff.gif'">Turn off the light</button>


</body>

</html>
```

</td>
<td>

<h2>What Can JavaScript Do?</h2>

JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.

Turn on the light Turn off the light

</td>
<td></td>
</tr>
<tr>
<td>

```
<!DOCTYPE html>

<html>

<body>


<h2>What Can JavaScript Do?</h2>
```

</td>
<td>

<h2>What Can JavaScript Do?</h2>

JavaScript can change the style of an HTML element.

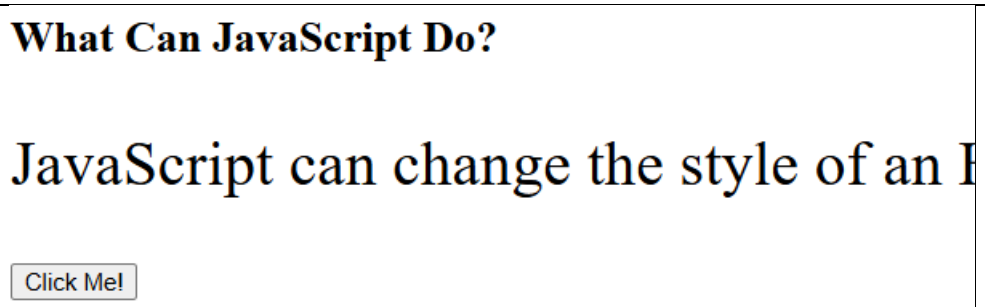
Click Me!

</td>
<td></td>
</tr>
</table>

<table>
<tr>
<td>

```
<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button" onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>

</body>

</html>
```

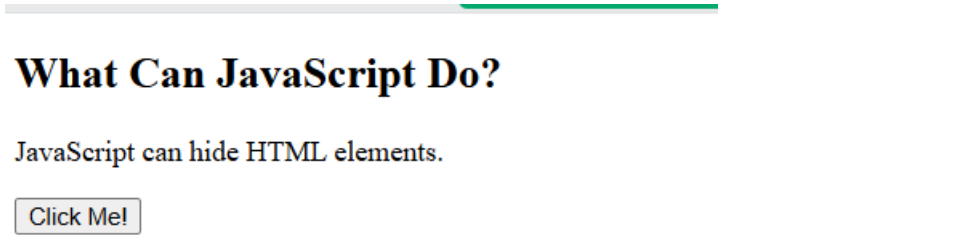
</td>
<td>



</td>
</tr>
<tr>
<td>

```
<!DOCTYPE html>

<html>

<body>



<h2>What Can JavaScript Do?</h2>



<p id="demo">JavaScript can hide HTML elements.</p>
```

</td>
<td>



</td>
</tr>
</table>

<table>
<tr>
<td>

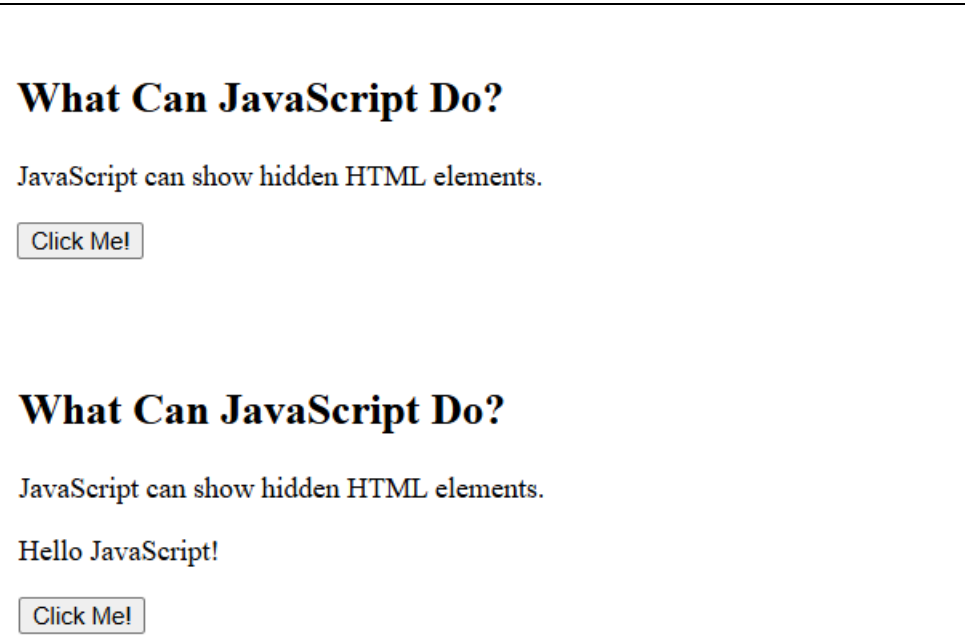
```
<button type="button"
onclick="document.getElementById('demo').style.display='none'">Click Me!</button>



</body>

</html>
```

</td>
<td>

</td>
<td></td>
</tr>
<tr>
<td>

```
<!DOCTYPE html>

<html>

<body>



<h2>What Can JavaScript Do?</h2>



<p>JavaScript can show hidden HTML
elements.</p>



<p id="demo" style="display:none">Hello
JavaScript!</p>
```

</td>
<td>



</td>
<td></td>
</tr>
</table>

<button type="button" onclick="document.getElementById('demo').style.display='block'">Click Me!</button> </body> </html>		
<!DOCTYPE html> <html> <body> JavaScript in Body <script> document.getElementById("demo").innerHTML = "My First JavaScript"; </script> </body> </html>	**JavaScript in Body** My First JavaScript	

JavaScript Functions and Events

A JavaScript `function` is a block of JavaScript code, that can be executed when "called" for.

For example, a function can be called when an **event** occurs, like when the user clicks a button.

You will learn much more about functions and events in later chapters.

JavaScript in <head> or <body>

You can place any number of scripts in an HTML document.

Scripts can be placed in the `<body>`, or in the `<head>` section of an HTML page, or in both.

JavaScript in <head>

In this example, a JavaScript `function` is placed in the `<head>` section of an HTML page.

The function is invoked (called) when a button is clicked:

Code	Browser View
<!DOCTYPE html> <html> <head> <script> function myFunction() { document.getElementById("demo").innerHTML = "Paragraph changed."; } </script> </head> <body> Demo JavaScript in Head A Paragraph. <button type="button" onclick="myFunction()">Try it</button> </body> </html>	

JavaScript in <body>

In this example, a JavaScript `function` is placed in the `<body>` section of an HTML page.

The function is invoked (called) when a button is clicked:

Code	Browser View	
<!DOCTYPE html> <html> <body> Demo JavaScript in Body A Paragraph. <button type="button" onclick="myFunction()">Try it</button> <script> function myFunction() { document.getElementById("demo").innerHTML = "Paragraph changed."; } </script> </body> </html>		

External JavaScript

Scripts can also be placed in external files:

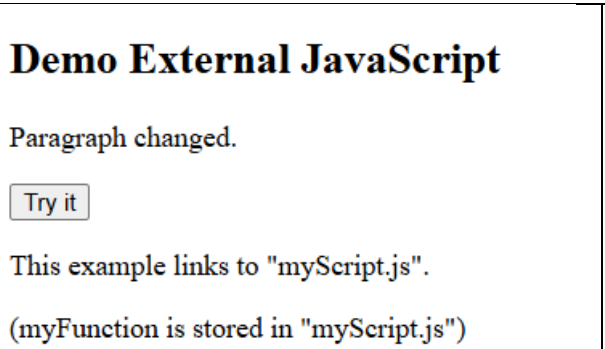
External file: myScript.js

External scripts are practical when the same code is used in many different web pages.

JavaScript files have the file extension **.js**.

To use an external script, put the name of the script file in the `src` (source) attribute of a `<script>` tag:

Code	Browser View	
<!DOCTYPE html> <html> <body> Demo External JavaScript A Paragraph. <button type="button" onclick="myFunction()">Try it</button>	**Demo External JavaScript** A Paragraph. Try it This example links to "myScript.js". (myFunction is stored in "myScript.js")	

This example links to "myScript.js". (myFunction is stored in "myScript.js") <script src="myScript.js"></script> </body> </html>	 	

JavaScript Output

JavaScript Display Possibilities

JavaScript can "display" data in different ways:

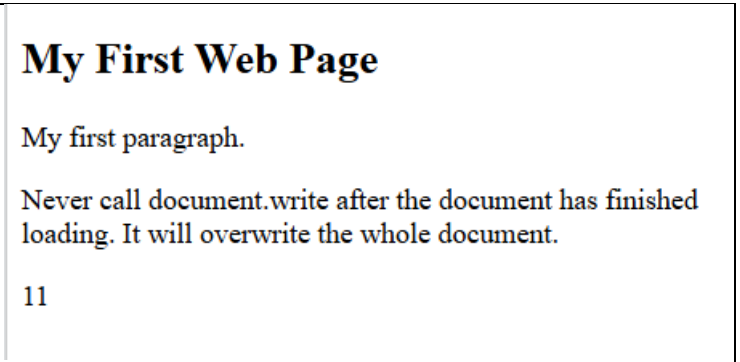
- Writing into an HTML element, using `innerHTML`.
- Writing into the HTML output using `document.write()`.
- Writing into an alert box, using `window.alert()`.
- Writing into the browser console, using `console.log()`.

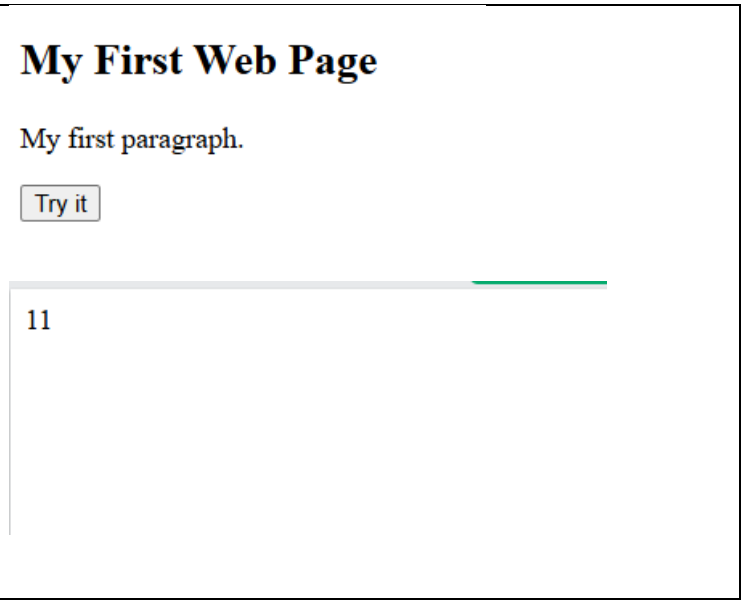
	Code	Browser View
Using innerHTML To access an HTML element, JavaScript can use the `document.getElementById(id)` method. The `id` attribute defines the HTML element. The `innerHTML` property defines the HTML content:	**<!DOCTYPE html> <html> <body> My First Web Page My First Paragraph. <script> document.getElementById("demo").innerHTML = 5 + 6; </script> </body> </html>**	**My First Web Page** My First Paragraph. 11

Using document.write()

For testing purposes, it is convenient to use `document.write()`:

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<p>Never call document.write after the document has finished loading.
It will overwrite the whole document.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```



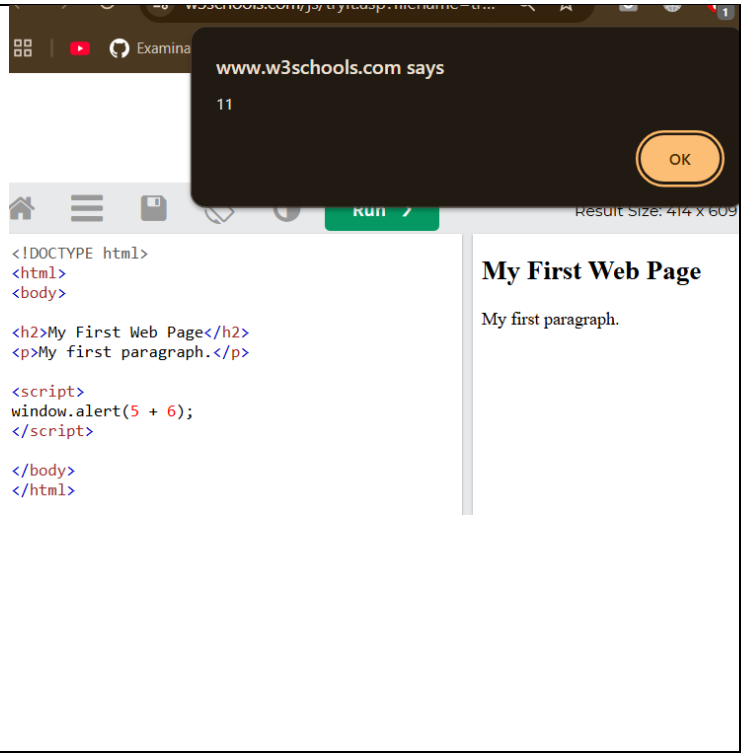
Using document.write() after an HTML document is loaded, will **delete all existing HTML**: The document.write() method should only be used for testing.	**<!DOCTYPE html> <html> <body> My First Web Page My first paragraph. <button type="button" onclick="document.write(5 + 6)">Try it</button> </body> </html>**	

Using window.alert()

You can use an alert box to display data:

```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

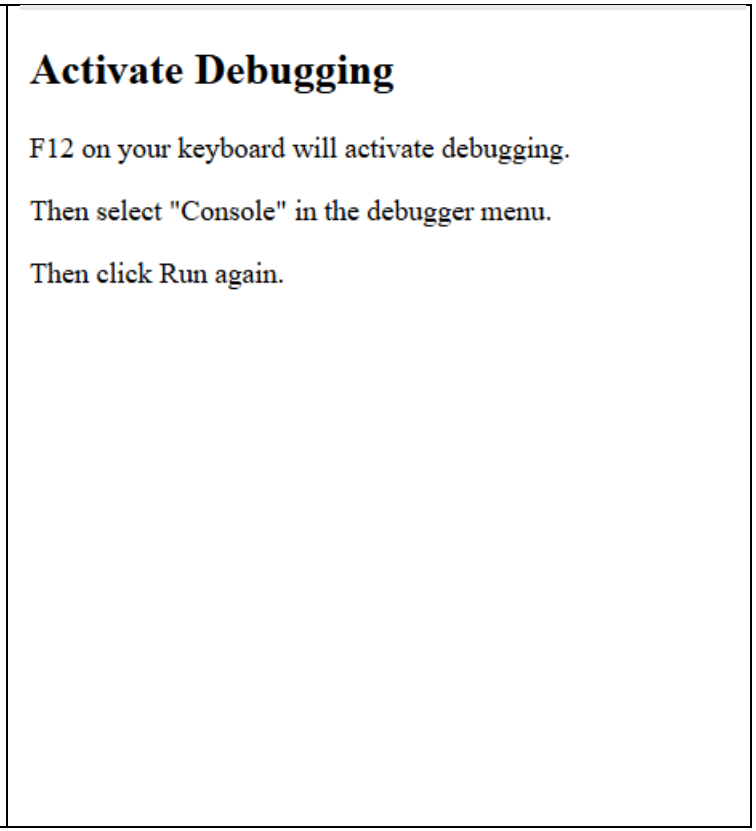
```
<script>
alert(5 + 6);
</script>
```



Using console.log()

For debugging purposes, you can call the `console.log()` method in the browser to display data.

```
<!DOCTYPE html>
<html>
<body>

<h2>Activate Debugging</h2>

<p>F12 on your keyboard will activate debugging.</p>
<p>Then select "Console" in the debugger menu.</p>
<p>Then click Run again.</p>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```



JavaScript Print

JavaScript does not have any print object or print methods.

You cannot access output devices from JavaScript.

The only exception is that you can call the `window.print()` method in the browser to print the content of the current window.

```
<!DOCTYPE html>
<html>
<body>

<h2>The window.print() Method</h2>

<p>Click the button to print the current page.</p>

<button onclick="window.print()">Print this page</button>

</body>
</html>
```