Cheat Sheet: Introduction to Web Development with HTML, CSS, JavaScript

Estimated reading time: 20 minutes

HTML		
Element	**Description**	**Example**
<!-- -->	This tag denotes a comment in HTML, which is not displayed by a browser but can be useful to hide and document code.	<!-- This is a comment -->
<!DOCTYPE html>	All HTML documents must start with this declaration. It tells the browser what document type to expect. Note that this element has no ending tag.	<!DOCTYPE html> <html> <head> <!-- Metadata Here --> </head> <body> <!-- Document Body Here --> </body> </html>
<a href= "path">link name</a>	This tag, called an "anchor tag" creates hyperlinks using the href attribute. In place of path enter the URL or path name to the page you want to link to.	<a href="https://www.ibm.com">IBM</a>
<body>	Contains the contents of the HTML document. It should contain all other tags besides the <head> element to display the body of the document.	<!DOCTYPE html> <html> <head> <!-- Metadata Here --> </head> <body> <!-- Document Body Here --> </body> </html>
<div>	Often used to separate sections in the body of a document in order to style that content with CSS.	<div> This element has no particular semantic meaning but is often used in conjunction wit </div>
<h1>	Adds a level 1 heading to the HTML document.	<h1>Thomas J. Watson</h1>
<head>	Contains metadata and should be placed after the <html> tag and before the <body> tag.	<!DOCTYPE html> <html> <head> <!-- Metadata Here --> </head> <body> <!-- Document Body Here --> </body> </html>
<html>	The root element of an HTML document. All other tags in the document should be contained in this tag.	<!DOCTYPE html> <html> <head> <!-- Metadata Here --> </head> <body> <!-- Document Body Here --> </body> </html>
<img src="path" width="dim1" height="dim2">	This tag is used to place an img. In place of path insert a URL or a relative file path to the image location. Other optional attributes include width and height of the image in pixels.	<img src="https://upload.wikimedia.org/wikipedia/commons/7/7e/Thomas_J_Watson_Sr.jpg" wid
<li>	Element that creates bulleted line items in an ordered or unordered list. Should be used in conjunction with the <ul> or <ol> tags.	<ul> <l1>Bullet point 1<l1> <l1>Bullet point 2<l1> </ul>
<link>	Used to link an external document, such as a CSS file, to an HTML document.	<head> <link rel="stylesheet" href="styles.css"> </head>
<meta>	Used to provide metadata about the HTML document.	<head> <meta name="author" content="Christopher Moore"> </head>
<ol>	Element that creates an ordered list using numbers. Should be used in conjunction with the <li> tag.	<ol> <li>Numbered bullet point 1<li> <li>Numbered bullet point 2<li> </ol>
<p>	This tag is used to identify a paragraph. It places a line break after the text it is enclosed in.	<p>This is a paragraph of text. It can be as short or as long as needed.</p>
<script>	Used to embed JavaScript in an HTML document.	<script> alert("Hello World"); </script>

HTML		

<table>	This tag is used to denote a table. Should be used with <tr> (defines a table row) and <td> (defines a table cell within a row) tags. The <th> tag can also be used to define the table header row.	`<table>`
		`<tr>`
		`<th>Header cell 1</th>`
		`<th>Header cell 2</th>`
		`</tr>`
		`<tr>`
		`<td>First row first cell</td>`
		`<td>First row second cell</td>`
		`</tr>`
		`<tr>`
		`<td>Second row first cell</td>`
		`<td>Second row second cell</td>`
		`</tr>`
		`</table>`
		
		`</td>`

<td>	Denotes a cell within a row, within a table.	`<td>Cell Content</td>`

<th>	Denotes the header cells within a row within a table.	`<table>`
		`<tr>`
		`<th>Header cell 1</th>`
		`<th>Header cell 2</th>`
		`</tr>`
		`<tr>`
		`<td>First row first cell</td>`
		`<td>First row second cell</td>`
		`</tr>`
		`<tr>`
		`<td>Second row first cell</td>`
		`<td>Second row second cell</td>`
		`</tr>`
		`</table>`

<title>	Defines the title of the HTML document displayed in the browser's title bar and tabs. It is required in all HTML documents. It should be contained in the <head> tag.	`<!DOCTYPE html>`
		`<html>`
		`<head>`
		`<title>Document Title</title>`
		`</head>`
		`<body>`
		`<!-- Document Body Here -->`
		`</body>`
		`</html>`

<tr>	Denotes a row within a table.	`<table>`
		`<tr>`
		`<th>Header cell 1</th>`
		`<th>Header cell 2</th>`
		`</tr>`
		`<tr>`
		`<td>First row first cell</td>`
		`<td>First row second cell</td>`
		`</tr>`
		`<tr>`
		`<td>Second row first cell</td>`
		`<td>Second row second cell</td>`
		`</tr>`
		`</table>`

<ul>	Element that creates an unordered list using bullets. Should be used in conjunction with thetag.	`<ul>`
		`<li>Bullet point 1<li>`
		`<li>Bullet point 2<li>`
		`</ul>`

CSS		
Element	**Description**	**Example**

OCTYPE html>	All HTML documents must start with this declaration. It tells the browser what document type to expect. Note that this element has no ending tag.	`<!DOCTYPE html>`
		`<html>`
		`<head>`
		`<title>Document Title</title>`
		`</head>`
		`<body>`
		`Document body here`
		`</body>`
		`</html>`

<a href= "path">	This tag, called an "anchor tag" creates hyperlinks using the href attribute. In place of path enter the URL or path name to the page you want to link to.	`<a`
		`href="https://www.ibm.com">IBM</a>`

<article>	Identifies a self-contained piece of content that could be distributed to other websites and platforms as a stand- alone unit. Similar to <div> tag in that it does not render as anything special in the browser unless it is styled with CSS.	`<article`
		`class="CSS-Style-Reference">`
		`<h2>HTML</h2>`
		`<p>HTML stands for Hypertext Markup Language.`
		`It is a standardized system for tagging text files to display specific`
		`font, color, graphics, and hyperlinks on World Wide Web pages.</p>`
		`</article>`

<aside>	Defines some content aside from the content it is placed in. Similar to <div> tag in that it does not render as anything special in the browser unless it is styled with CSS.	`<aside>`
		`Use the aside tag to represent a section of a page that consists`
		`of content tangentially related to the content.`
		`</aside>`

<body>	Contains the contents of the HTML document. It should contain all other tags besides the <head> element	`<!DOCTYPE html>`
		`<html>`
		`<head>`
		`<title>Document Title</title>`
		`</head>`
		`<body>`

HTML		

	to display the body of the document.	`Document body here`
		`</body>`
		`</html>`

<div>	Often used to separate sections in the body of a document in order to style that content with CSS.	`<div>`
		`This element has no particular semantic meaning but is often used in conjunction wit`
		`</div>`

<fieldset>	Groups related elements in a form and puts a box around them.	`<form action= "/script.php">`
		`<fieldset>`
		`<legend>User:</legend>`
		`<label for="fname">First name:</label>`
		`<input type="text" id="fname"`
		`name="fname"><br>
		`<br>`<label for="lname">Last name:</label>`
		`<input type="text" id="lname"`
		`name="lname"><br>
		`<br>`<input type="submit" value="Submit">`
		`</fieldset>`
		`</form>`

<figcaption>	Used in conjunction with the <figure> tag to mark up an image.	`<figure>`
		`<img src="durham.jpg">`
		`<figcaption>Fig.1 - Durham, NC</figcaption>`
		`</figure>`

<figure>	Used to mark up an image in conjunction with the <figcaption> tag.	`<figure>`
		`<img src="durham.jpg">`
		`<figcaption>Fig.1 - Durham, NC</figcaption>`
		`</figure>`

<footer>	Contains a footer of a document and often contains information such as authoring, copyright info, contact info, sitemap, and related documents.	`<footer>`
		`<p>Author: Christopher Moore</p>`
		`</footer>`
		`<form action="/script.php">`

<form>	Creates an HTML form for user input.	`<form action="/script.php">`
		`<fieldset>`
		`<legend>User:</legend>`
		`<label for= "fname">First name:</label>`
		`<input type="text" id="fname"`
		`name="fname"><br>
		`<br>`<label for= "lname">Last name:</label>`
		`<input type="text" id="lname"`
		`name="lname"><br>
		`<br>`<input type="submit" value="Submit">`
		`</fieldset>`
		`</form>`

<h1>	Adds a level 1 heading to the HTML document.	`<h1>Thomas J.`
		`Watson</h1>`

<head>	Contains metadata and should be placed after the <html> tag and before the <body> tag.	`<!DOCTYPE html>`
		`<html>`
		`<head>`
		`<title>Document Title</title>`
		`</head>`
		`<body>`
		`Document body here`
		`</body>`
		`</html>`

<header>	A container for introductory content such as heading elements, logo, or authoring information.	`<header>`
		`<h1>Header tag example</h1>`
		`<p>It works as a container for introductory content such as heading elements, logo,`
		`<p>Author: Christopher Moore</p>`
		`</header>`

<html>	The root element of an HTML document. All other tags in the document should be contained in this tag.	`<!DOCTYPE html>`
		`<html>`
		`<head>`
		`<title>Document Title</title>`
		`</head>`
		`<body>`
		`Document body here`
		`</body>`
		`</html>`

<img src="path" width="dim1" height="dim2">	This tag is used to place an img. In place of path insert a URL or a relative file path to the image location. Other optional attributes include width and height of the image in pixels.	`<img`
		`src="https://upload.wikimedia.org/wikipedia/commons/7/7e/Thomas_J_Watson_Sr.jpg"`
		`width="300"`
		`height="300">`

<input type="?">	Specifies an input field on a form with the type attribute. Common input types include: "color," "date," "datetime-local," "email," "number," "range," "search," "url," "tel," "datalist," "select," "text," "option," and "placeholder."	`<form action="/script.php">`
		`<fieldset>`
		`<legend>User:</legend>`
		`<label for="fname">First name:</label>`
		`<input type="text" id="fname"`
		`name="fname"><br>
		`<br>`<label for="lname">Last name:</label>`
		`<input type="text" id="lname"`
		`name="lname"><br>
		`<br>`<input type="submit" value="Submit">`
		`</fieldset>`
		`</form>`

<li>	Element that creates bulleted line items in an ordered or unordered list. Should be used in conjunction with the <ul> or <ol> tags.	`<ul>`
		`<li>Bullet point 1</li>`
		`<li>Bullet point 2</li>`
		`</ul>`

HTML		
<link>	Used to link a CSS document to an HTML document.	<head> <link rel="stylesheet" href="styles.css"> </head>
<meta>	Used to provide metadata about the HTML document.	<head> <meta name="author" content="Christopher Moore"> </head>
<nav>	Used to define a set of navigational elements.	<nav> <a href="home.html">Home</a> <a href="page1.html">Page 1</a> <a href="page2.html">Page 2</a> </nav>
<ol>	Element that creates an ordered list using numbers. Should be used in conjunction with the <li> tag.	<ol> <li>Numbered bullet point 1</li> <li>Numbered bullet point 2</li> </ol>
<p>	This tag is used to identify a paragraph. It places a line break after the text it is enclosed in.	<p>Thomas J. Watson, Sr. is the American industrialist, who built the International Business Machines Corporation (IBM) into the largest manufacturer of electric typewriters and data processing equipment in the world. </p
<script>	Used to embed Javascript in an HTML document.	<script> alert("Hello World"); </script>
<section>	Defines an important section of a document. Can be used within headers and footers as well.	<section> <h2>Introduction</h2> <p>This document provides some examples of how to use a variety of HTML elements</p> </section>
<style>	Used to apply simple CSS to an HTML document.	<head> <style> p {color:red} </style> </head> <body> <p>This paragraph will be red because I've styled the paragraph tag with CSS.</p> </body>
<table>	This tag is used to denote a table. Should be used with <tr> (defines a table row) and <td> (defines a table cell within a row) tags. The <th> tag can also be used to define the table header row.	<table> <tr> <th>Header cell 1</th> <th>Header cell 2</th></tr> <tr> <td>First row first cell</td> <td>First row second cell</td> </tr> <tr> <td>Second row first cell</td> <td>Second row second cell</td> </tr> </table>
<td>	Denotes a cell within a row, within a table.	<table> <tr> <th>Header cell 1</th> <th>Header cell 2</th> </tr> <tr> <td>First row first cell</td> <td>First row second cell</td> </tr> <tr> <td>Second row first cell</td> <td>Second row second cell</td> </tr> </table>
<th>	Denotes the header cells within a row within a table.	<table> <tr> <th>Header cell 1</th> <th>Header cell 2</th> </tr> <tr> <td>First row first cell</td> <td>First row second cell</td> </tr> <tr> <td>Second row first cell</td> <td>Second row second cell</td> </tr> </table>
<title>	Defines the title of the HTML document displayed in the browser's title bar and tabs. It is required in all HTML documents. It should be contained in the <head> tag.	<!DOCTYPE html> <html> <head> <title>Document Title</title> </head> <body> Document body here </body> </html>
<tr>	Denotes a row within a table.	<table> <tr> <th>Header cell 1</th> <th>Header cell 2</th> </tr> <tr> <td>First row first cell</td> <td>First row second cell</td> </tr> <tr> <td>Second row first cell</td> <td>Second row second cell</td>

HTML		
		</tr> </table>
<ul>	Element that creates an unordered list using bullets. Should be used in conjunction with the <li> tag.	<ul> <li>Bullet point 1</li> <li>Bullet point 2</li> </ul>
JAVASCRIPT		
Element	**Description**	**Example**

Child()	An HTML DOM method that after creating an element, you can use this function to place the element in the appropriate location within the document. The element to append is the only parameter.	//Creates the element <p> and text "Hello World". Appends Hello World <p> to the HTM
		<head>
		<script>
		function addPara() {
		var newPara = document.createElement("p");
		var newText = document.createTextNode("Hello World!");
		newPara.appendChild(newText);
		document.body.appendChild(newPara);
		}
		</script>
		</head>
		<body onload="addPara()">
		</body>

Arrays	Created by declaring the array elements in [ ]. An array can be assigned to a variable, usually using the keyword const or var. Arrays use zero based indexing to access their elements.	const Beatles = ["Ringo", "Paul", "George", "John"]; //Here Beatles[0] is "Ringo".

Date()	Constructor is new Date([optional parameters]). If the constructor is declared with no parameters, it returns current local date and time. New dates can be created by passing parameters to new Date function.	//create a new date from a string
		var newDate = new Date("2021-1-17 13:15:30");
		
		//create a new date instance representing 17 Jan 2021 00:00:00
		//note that the month number is zero-based
		var newDate = new Date(2021, 0, 17);

document.createElement()	Takes one tag name parameter and creates an element with that name. Can place the element elsewhere on the page using functions like insertBefore(), appendChild(), replaceChild().	//Creates the element <p> and text "Hello World". Appends Hello World <p> to the HTM
		<head>
		<script>
		function addPara() {
		var newPara = document.createElement("p");
		var newText = document.createTextNode("Hello World!");
		newPara.appendChild(newText);
		document.body.appendChild(newPara);
		}
		</script>
		</head>
		<body onload="addPara()">
		</body>

document.createTextNode()	Takes a string as input text and returns a text node with the input text.	//Creates the element <p> and text "Hello World". Appends Hello World <p> to the HTM
		<head>
		<script>
		function addPara() {
		var newPara = document.createElement("p");
		var newText = document.createTextNode("Hello World!");
		newPara.appendChild(newText);
		document.body.appendChild(newPara);
		}
		</script>
		</head>
		<body onload="addPara()">
		</body>

document.getElementByID()	A method of the DOM that takes an ID value parameter and returns an element that matches the id.	//Changes the content of the div to "Hello World!" <div id="div1"> <p>Hello</p> <p>Hello</p> </div> <script> document.getElementById("div1").innerHTML = "<p>Hello World!</p>"; </script>
document.getElementsByTagName()	A method of the DOM that takes a tag name parameter and returns an array called "NodeList" that contains elements with the specified tag name.	//Gets an array of all elements in a document with the <p> tag. var tagNameArray = document.getElementsByTagName("p");

document.write()	Writes HTML or JavaScript to a document. Note that it overwrites any other text in the document so is mostly used for testing purposes only.	//Writes "Hello World" to the output stream.
		document.write("Hello World");

element.getAttribute()	Returns the value of the specified attribute. Takes one parameter: the attribute name whose value is to be returned.	//Removes the CSS style color blue
		<div id="div1" style="color: blue"></div>
		<script>
		var div1 = document.getelementById("div1").getAttribute("style");
		</script>

HTML		
element.innerHTML()	A property of the Element class that returns or alters contents of an HTML element as a text string.	//Changes the content of the div to "Hello World!" <div id="div1"> <p>Hello</p> <p>Hello</p> </div> <script> document.getElementById("div1").innerHTML = "<p>Hello World!</p>"; </script>

element.removeAttribute()	A property of the Element class that removes all previously set inline CSS styles for a particular element. Takes one parameter: the attribute name that is being removed.	//Removes the CSS style color blue
		<div id="div1" style="color: blue"></div>
		<script>
		var div1 = document.getelementById("div1").getAttribute("style");
		</script>

element.setAttribute()	A property of the Element class that overwrites all previously set inline CSS styles for a particular element. Takes two parameters: the attribute name that is being set and the attribute value the attribute is set to.	//In all elements named "theImage" sets the name of all src attributes to "another.g
		document.getElementById("theImage").setAttribute("src", "another.gif");

element.style()	A property of the Element class that returns or alters inline CSS. Syntax is element.style.propertyName = value	//Changes the CSS style color from blue to red
		<div id="div1" style="color: blue"></div>
		<script>
		var div1 = document.getelementById("div1");
		div1.style.color = "red";
		</script>

Error Objects	Instance creates two properties about the error: message that contains description of the error and the name property identifies the type of error. Generic error plus 6 other core errors: TypeError, RangeError, URIError, EvalError, ReferenceError, SyntaxError.
	Error object can be extended to create custom error messages using the throw keyword.
		//Catch statement defines a block of code to be executed if an error occurs in the t
		catch (err) {
		document.getElementById("myfile").innerHTML = err.name;
		}
		//Creates custom error message
		throw new Error("Only values 1-10 are permitted");

History Objects	The history object is part of the window object and contains the URLs visited by the user within a browser window. It exposes useful methods and properties that let you navigate back and forth through the user's history and manipulate the contents of the history stack.	//Go back two pages if the history exists in the history list.
		history.go(-2);

insertBefore()	An HTML DOM method that, after creating an element, places a child element in the appropriate location before an existing child. The method takes two parameters, the node object to be inserted and the existing node to insert before.	//Creates a new <li> element and places it in the elementList before the first child
		let newLI = document.createElement("li");
		newLI.innerText = "new Element";
		let elementList = document.getElementById("thisList");
		elementList.insertBefore(newLI, elementList.childNodes[0]);

Location Objects	The location object is part of the window object and contains information about the current URL.	//Returns the hostname property
		let myhost = location.hostname;
		newLI.innerText = "new Element";

Navigator Objects	The navigator object is part of the window object class in the DOM that represents the client Internet browser, also called the user agent. There is no standard for this object so what it returns differs from browser to browser.	//Retrieves the name of the browser var browsername = navigator.appName;
onload()	A DOM event that starts a method when a page is loaded.	//Executes myFunction after MyHTMLPage has been loaded document.getElementById("MyHTMLPage").onload = function () {myFunction};

replaceChild()	After creating an element, this function replaces a child node with a new node.	//Creates a new node and replaces the second element in "thisList" with the word "bl
		let secondBullet = document.createTextNode("blue");
		var myList = document.getElementById("thisList").childNodes[1];
		myList.replaceChild(secondBullet, myList.childNodes[1]);

HTML		

Screen Objects	The screen object is part of the window object class in the DOM that can be used to return properties about the user's screen.	//Returns the height and width of the user's screen
		var height=screen.height;
		var width=screen.width;

Window Objects	The DOM window object is at the top of the DOM hierarchy and serves as the global object. Everything in the DOM takes place in a window. The window object controls the environment that contains the document.	//Opens a new browser window with the specified URL
		window.open("http://www.w3schools.com");

window.open()	Opens a new window. The first parameter is a path, a URL, or an empty string, and optional parameters include the window name, features such as the placement of the window or the dimensions, and a Boolean replace value. The feature parameter is a comma separated string of name-value pairs and the replace parameter is an optional Boolean. This parameter has been deprecated so modern browsers may not support it. This method returns a reference to the new window object.	//Opens a new window that opens the IBM home page and has a width of 600 and a heigh let thisWindow = window.open("http://www.ibm.com", "myWindow", "width"=600, "height"=800);

window.scrollTo()	Scrolls to a particular place in a window. Parameters include the x-coordinate which is the left-most pixel and the y-coordinate which is the upper-most pixel.	//Scrolls the window to the pixel located at
		the coordinate (20, 200)
		window.scrollTo(20, 200);

Wrapper Objects	Primitive types can be converted to objects using wrapper objects. They are the same name as the primitive except they start with uppercase letter. The typeof keyword returns a string indicating the data type of the operand.	//Enables the use of properties and methods of the String class such as the property
		let n = new String ("abc");
		
		//Returns string
		typeof "abc";
		
		//Returns object
		typeof new String("abc");

Skills Network