

# HTML - Document Object Model

## HTML Document Object Model (DOM)

The **HTML Document Object Model** (in short, **HTML DOM**) represents all the elements of an HTML document in a hierarchical order (or tree structure). Where each node of this tree represents an element in the HTML document.

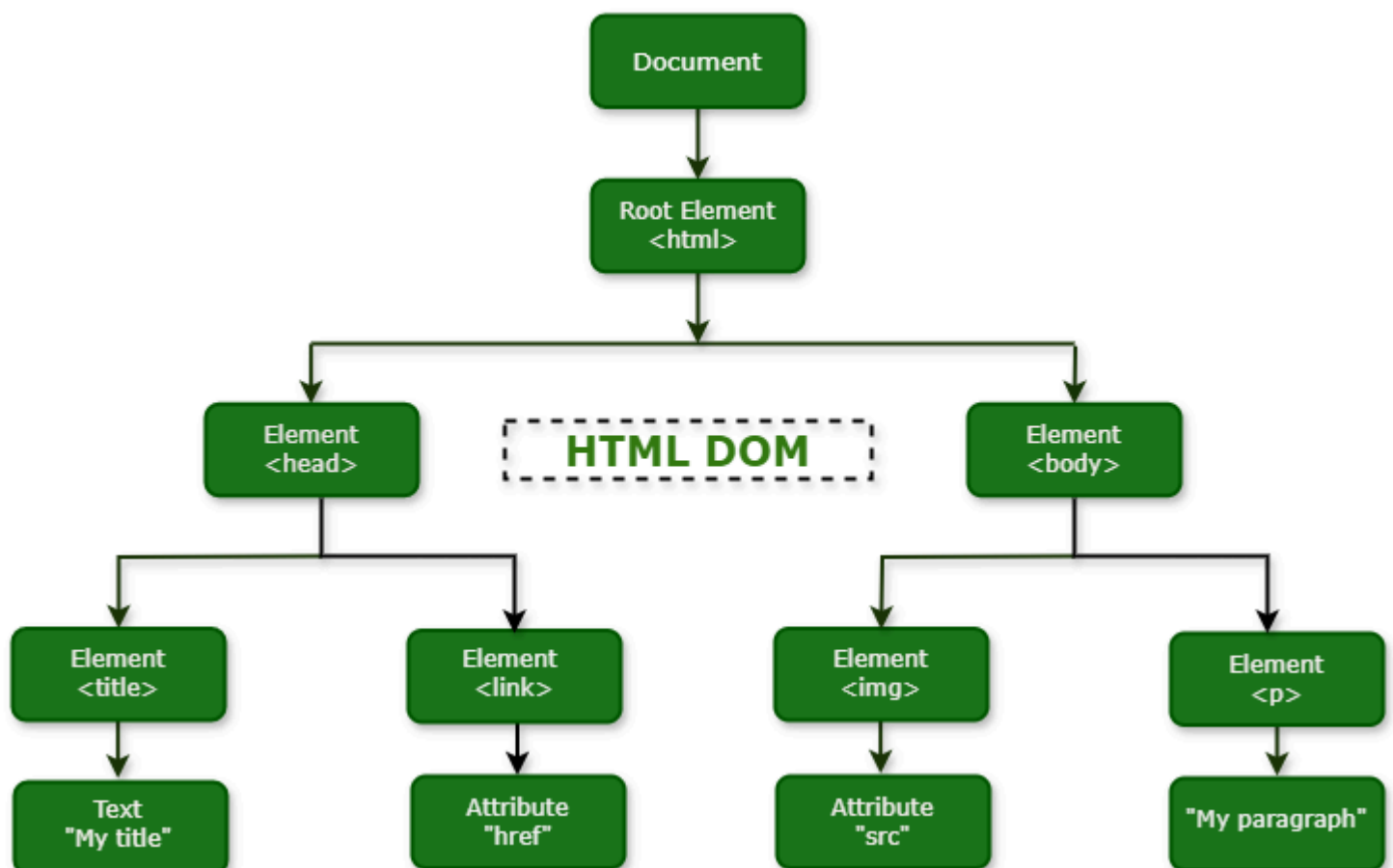
## Accessing and Modifying HTML DOM

Using the **HTML DOM methods** we can access the tree and modify the structure or, contents of the respective HTML document. We can also have events attached to the nodes.

Explore our **latest online courses** and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## HTML DOM Tree Structure

For instance, if your HTML document contains elements like `<html>`, `<head>`, `<body>`, `<title>`, `<link>`, `<img>` and `<p>`, the browser will create a DOM Tree of the HTML document that can be represented as shown in the diagram given below –



Please note that each HTML document contain `<html>`, `<head>` and `<body>` tags. The root element is `<html>`, and `<head>` and `<body>` tags are its child element.

## What is Document Object Model?

The Document Object Model (DOM) is a programming interface that works as a bridge between web pages and scripts or programming languages. It represents a web document (like an HTML or XML) as a tree of objects where each branch of the tree ends with a node, and each node contains objects.

Click on the button given below to understand it properly. It will generate a DOM Tree.

Create DOM Tree



The DOM provides a set of methods that allow programming languages, such as JavaScript, to access the DOM tree. Using these methods, you can change the document's structure, style, or content. It makes a web page interactive and dynamic.

The DOM is not a programming language, it is designed to be language independent. Most web developers use the DOM through JavaScript, however, implementations of the DOM can be built for any language.

## HTML DOM vs JavaScript DOM

We use HTML to structure a web page and JavaScript to make it interactive. However, JavaScript can not understand the a web page directly. It takes help of the HTML DOM. When an HTML page is loaded, the browser creates an Object Model of the page, which JavaScript can then interact with to manipulate the content, structure, and styles of the page.

JavaScript can perform the below operations with the help of object model –

- Access and replace HTML elements.
- Access and replace HTML attributes.
- Change all the CSS styles in the page.
- Respond to user events.
- Add animation to the web page.

The below table explains the difference between HTML DOM and JavaScript DOM –

HTML DOM	JavaScript DOM
The HTML DOM is an Object Model for HTML that represents all elements of an HTML document in a tree like structure.	The HTML DOM is an API for JavaScript that helps add, change, and replace elements, attributes and events of an HTML document.

## DOM Methods Reference

Below is a list of important DOM methods –

Sr.No	Method & Description
-------	----------------------

1.	<b>toString()</b> It is used to convert an HTML element into a string format.
2.	<b>setAttribute()</b> This method allows you to define attributes for a particular element.
3.	<b>setAttributeNode()</b> This method allows you to define a particular attribute node on a particular element.
4.	<b>scrollIntoView()</b> It ensures that a particular element on a web page of scrollable container becomes visible by adjusting the scroll position.
5.	<b>querySelector()</b> It is used to select and access the first HTML element that matches a given CSS selector(s) within the document.
6.	<b>querySelectorAll()</b> This method allows you to select and access all HTML element that matches a given CSS selector(s) within the document.
7.	<b>remove()</b> This method can delete an element completely from the web page.
8.	<b>removeAttribute()</b> This method is used to delete any attribute that has been set on an HTML element within the DOM structure.
9.	<b>removeAttributeNode()</b> It allows you to delete a specific attribute node from an element.
10.	<b>removeChild()</b> It is used to delete the selected child element from its parent element.
11.	<b>removeEventListener()</b> This method allows you to delete an event listener that was previously assigned to an element.
12.	<b>replaceChild()</b> This method enable us to replace one child element with another in a parent element.
13.	<b>hasAttribute()</b> It is used to check whether an attribute exists within an HTML element.
14.	<b>hasAttributes()</b> This method checks whether an element in HTML DOM has attributes.
15.	<b>hasChildNodes()</b> It is used to check if an HTML element has any child element inside it.

Sr.No	Method & Description
16.	<b>getAttribute()</b> It returns the value of a specified attribute that belongs to an HTML element.
17.	<b>getBoundingClientRect()</b> This method is used to get the size of an element and its position relative to the viewport.
18.	<b>closest()</b> This method returns the closest ancestor of the current element (or the current element itself) that matches a given CSS selector. If no such ancestor exists, it returns null.
19.	<b>contains()</b> You can check if a DOM Element contains another element within its subtree using this method.
20.	<b>click()</b> The click() method activates a mouse click on an element.
21.	<b>normalize()</b> It is used to combine adjacent text nodes within an HTML element by removing any empty text nodes.
22.	<b>isDefaultNamespace()</b> It is used to check if a specific namespace URI is the default namespace for elements within a document or element.
23.	<b>isEqualNode()</b> This method checks if two nodes are equal by comparing their attributes, types, and child nodes.
24.	<b>isSameNode()</b> It checks whether two node references point to the same node object within an HTML document.
25.	<b>isSupported()</b> This method is used to check if a web browser can supports or handle a particular feature or functionality on a web page.
26.	<b>insertAdjacentElement()</b> This method allows you to insert a new HTML element exactly where you need it relative to another element on a webpage.
27.	<b>insertBefore()</b>

	This method allows you to add a new child element inside a parent element, specifying its position relative to another child element.
28.	<b>blur()</b> This method allows you to dynamically remove focus from an element in the HTML DOM.
29.	<b>matches()</b> This method checks if an element matches a given CSS selectors.
30.	<b>insertAdjacentHTML()</b> It helps you to insert a specified HTML code at a specific position relative to the element that calls this method.
31.	<b>addEventListener()</b> It is used to register event handlers to elements.
32.	<b>cloneNode()</b> It duplicates a node, including it's attributes and child nodes.
33.	<b>appendChild()</b> This method is used to add a new child node(element) as the last child node of the specified parent node.
34.	<b>compareDocumentPosition()</b> It enables understanding of document structure by comparing the positions of two DOM elements(nodes) and returns a bitmask(numeric value).
35.	<b>focus()</b> This method sets focus to specific webpage elements.
36.	<b>getAttributeNode()</b> It is used to get the attribute node object.
37.	<b>getElementsByClassName()</b> This method retrieves a live HTMLCollection of elements that match a specified class name within a document or a specific element.
38.	<b>getElementsByTagName()</b> It retrieves a live HTMLCollection of elements that match a specified tag name.
39.	<b>insertAdjacentText()</b> It allows you to insert a text content at a specific position relative to the element that calls this method.
40.	<b>namedItem()</b>

	It is used to get the first element of collection whose id or name matches with the name mentioned in parameter.
41.	<b>item()</b> It returns the element from the HTMLCollection located at specified index in parameter.
42.	<b>entries()</b> This method retrieves an iterator, that allows us to iterate through all the key/value pairs in the object.
43.	<b>forEach()</b> This method calls the callback function mentioned in the parameter once for each value pair in the list in the order of their insertion.
44.	<b>item()</b> This method retrieves a node from the NodeList specified by the index in the parameter.
45.	<b>keys()</b> This method retrieves an iterator that allows us to go through all the keys contained in the NodeList.
46.	<b>writeln()</b> This method provides the functionality to write HTML or JavaScript expressions directly to a document. It adds a newline character after each statement.
47.	<b>write()</b> This method provides the functionality to write HTML or JavaScript expressions directly to a document.
48.	<b>hasFocus()</b> It is used for determining if the document or any element inside the document has focus or not.
49.	<b>renameNode()</b> It is used to rename the nodes.
50.	<b>open()</b> This method opens a document for writing.
51.	<b>normalizeDocument()</b> This method normalize an entire HTML document.
52.	<b>normalize()</b> It removes empty text nodes, and joins adjacent text nodes from parent node.
53.	<b>adoptNode()</b>

	This method adopts a node from another document into the current document.
54.	<b>addEventListener()</b> It is used to set up a function that will be called whenever the specified event is delivered to the target.
55.	<b>execCommand()</b> This method is used for executing a command specified on the editable section that is being selected by the user.
56.	<b>createTextNode()</b> It is used to create a text node with the specified text.
57.	<b>createEvent()</b> It is used for creating an event object whose event type is specified in the parameter.
58.	<b>createDocumentFragment()</b> It creates a new empty document fragment which resides in memory.
59.	<b>createComment()</b> This method is used for creating a comment node with the given text.
60.	<b>createAttribute()</b> It is used to create an attribute with a specific name using JavaScript for an HTML element and return the Attr object.
61.	<b>close()</b> It closes the output stream.
62.	<b>getElementsByTagName()</b> It is a read-only method which is used to get the collection of all the elements in the document having the specified tag name in parameter.
63.	<b>getElementsByName()</b> This method is used to return collection of elements with the name attribute specified in the parameter.
64.	<b>getElementsByClassName()</b> This method is used for getting the collection of all the elements in the document with specified class name.
65.	<b>getElementById()</b> It returns the element of the given ID.
66.	<b>createElement()</b> This method creates an HTML element specified by tag name or element name.

67.	<b>add()</b> This method adds one or more tokens specified in the parameter to the DOMTokenList.
68.	<b>contains()</b> This method checks whether the list contains the specified token, and returns a boolean value accordingly.
69.	<b>entries()</b> This method returns an iterator that is allowing to go through all the key/value pairs.
70.	<b>forEach()</b> This method calls the callback function mentioned in the parameter once for each value pair in the list in the order of their insertion.
71.	<b>item()</b> This method returns a token from the DOMTokenList specified by the index in the parameter.
72.	<b>keys()</b> This method returns an iterator which allows you to go through all the keys contained in the token list.
73.	<b>remove()</b> This method removes one or more tokens specified in the parameter to the DOMTokenList.
74.	<b>replace()</b> This method replaces the existing token in DomTokenList with a new token specified in the parameter.
75.	<b>supports()</b> This method checks whether the token specified in the parameter is supported in DOMTokenList.
76.	<b>toggle()</b> This method dynamically adds or removes a token or class from an element class attribute.
77.	<b>values()</b> This method returns an iterator allowing us to go through all values contained in the token list.

## DOM Properties Reference

Below is a list of important DOM properties –



Sr.No	Method & Description
1.	<b>title</b> It helps us to access and update the value stored in an element's title attribute.
2.	<b>textContent</b> This property is used to access and update the text content of an HTML element and all its child elements as a single string.
3.	<b>tagName</b> It gives you the name of the HTML tag in uppercase that defines an element on a webpage.
4.	<b>style</b> Using this property, you can get the CSS styles that are directly set for a particular element.
5.	<b>tabIndex</b> It is used to access and update the value of the tabIndex attribute for an element.
6.	<b>scrollLeft</b> This property is used to access and update how far an element's content is scrolled horizontally.
7.	<b>scrollTop</b> It is used to access and update how far an element's content is scrolled vertically.
8.	<b>scrollWidth</b> This property gives you the total horizontal width of an element's content in pixels.
9.	<b>scrollHeight</b> You can get the total vertical height of an element's content in pixels using this property.
10.	<b>id</b> The id property is used for setting and retrieving the value of the element's id attribute.
11.	<b>innerText</b> It allows us to retrieve or change the visible text content directly within an HTML element on a web page.
12.	<b>isContentEditable</b> It is used to check if a webpage element can be edited by users directly.
13.	<b>lang</b> The lang property is an attribute of an HTML element that specifies the language code.
14.	<b>lastChild</b> The lastChild property returns a node that points to the last child node of a specific parent element.

15. **lastElementChild**  
It returns a node that holds the last child element of a parent element.

[Click here to view list of all methods....](#)



Sr.No	Properties & Description
16.	<b>namespaceURI</b> The namespaceURI property of an element provides the namespace URI assigned to the element.
17.	<b>nextElementSibling</b> Using this property, you can get the immediate next node in the sequence of a particular element.
18.	<b>nextSibling</b> It is used to access the immediate next node in sequence from the current node within the DOM tree.
19.	<b>nodeName</b> This property is used to identify a node's type and name based on its content.
20.	<b>nodeType</b> The nodeType Property returns an integer value representing the type of a node.
21.	<b>nodeValue</b> It is used for accessing and updating the value of a node.
22.	<b>offsetHeight</b> This property is used to get the complete visible height of that element on the webpage, including both its vertical padding and borders, measured in pixels.
23.	<b>offsetLeft</b> It returns the pixel based measurement of the distance from the element's left boundary to its nearest positioned parent's left boundary.
24.	<b>offsetParent</b> It is used to find the closest ancestor element that affects the positioning of another element.
25.	<b>offsetWidth</b> The offsetWidth property of an element gives you the total visible width of that element on the webpage.
26.	<b>outerHTML</b>

	The outerHTML property fetches the entire HTML code of an element.
27.	<b>outerText</b> This property can access or update the text content of an HTML element, including all its nested text and elements.
28.	<b>ownerDocument</b> It gives you the object for the document node that contains a specific element.
29.	<b>parentElement</b> It allows you to access the immediate parent element of a particular element within the HTML element.
30.	<b>parentNode</b> This property is used to access the immediate parent node of a particular node within the HTML element.
31.	<b>classList</b> It acts as an active container that holds a collection of CSS classes assigned to an element's class attribute.
32.	<b>childNodes</b> It is used to retrieve all child nodes of an element, including elements, text nodes, and comments.
33.	<b>className</b> You can access or modify the value of the class attribute of an element using this property.
34.	<b>clientTop</b> It is used to get the accurate element positioning and size calculation in webpage layouts.
35.	<b>firstElementChild</b> It provides the first child element within a given parent element.
36.	<b>offsetTop</b> Using this property, you can get the pixel-based vertical distance from the element's top edge of its nearest positioned ancestor's top edge.
37.	<b>attributes</b> It returns 'NameNodeMap' containing the collection of attributes of an HTML element
38.	<b>accesskey</b> This property allows you to assign a keyboard shortcut to an element on a webpage.

39.	<b>firstChild</b> It helps you to access the very first child node of a given parent element in the HTML DOM.
40.	<b>innerHTML</b> This property allows us to read the existing HTML content of an element and update it with new HTML content.
41.	<b>dir</b> It provides access for setting and retrieving the value of dir attributes of any element in HTML.
42.	<b>contentEditable</b> You can make text content inside the HTML elements editable using this property.
43.	<b>clientWidth</b> It returns the width of an element, including padding but excluding margin, border, or scrollbar width.
44.	<b>clientLeft</b> This property is used to get the width of an element's left border, excluding left padding or margin.
45.	<b>clientHeight</b> It is used to get the inner height of an element, including padding but not margin, border, or scrollbar width.
46.	<b>children</b> This property returns a collection of child elements.
47.	<b>childElementCount</b> It returns the count of direct child elements of a specified element.
48.	<b>src</b> This property used for elements like <img>, <script>, or <iframe> to get or set the source URL.
49.	<b>href</b> This property is used for anchor <a> elements to get or set the hyperlink reference.
50.	<b>checked</b> This property is used to get or set the checked state of a checkbox or radio button.
51.	<b>disabled</b> This property is used to get or set the disabled state of an input element.

52.	<b>length</b> It returns the number of elements in an HTMLCollection.
53.	<b>length</b> This method returns the number of items in the NodeList.
54.	<b>value</b> This property is used to set or get the value of an attribute.
55.	<b>specified</b> It checks whether the mentioned attribute is specified or not.
56.	<b>name</b> It is used get the name of the used attribute on an element.
57.	<b>isId</b> This property determines whether a HTML attribute is an 'id' attribute or not.
58.	<b>URL</b> It is a read-only property which returns the complete URL of the document including the protocol.
59.	<b>title</b> This property is used to set or get the title of the document.
60.	<b>strictErrorChecking</b> It is used to determine whether document enforce strict error checking or not.
61.	<b>scripts</b> It is a read-only property used for returning all the script elements present within HTML document as a collection.
62.	<b>links</b> The links is a read-only property which returns a collection of all the links corresponding to a and area elements with href attribute.
63.	<b>lastModified</b> This property returns the date and time of the current document when it was last modified.
64.	<b>inputEncoding</b> The inputEncoding property returns a String which represents the document's character encoding.
65.	<b>implementation</b>

	This returns a DOMImplementation object which is associated with the current document.
66.	<b>images</b> It is a read-only property used for returning all the img elements present within HTML document as a collection.
67.	<b>head</b> The head property returns the HTML head element.
68.	<b>forms</b> The forms is a read-only property used for returning all the form elements present within HTML document as a collection.
69.	<b>embeds</b> It is a read-only property which returns all the embed elements present within HTML document as a collection.
70.	<b>domConfig</b> This property is deprecated due to which will return undefined in all new browsers.
71.	<b>documentURI</b> This property is used to set or return the document's location.
72.	<b>documentMode</b> The documentMode property is an IE8 property which determines the rendering mode used by browser.
73.	<b>documentElement</b> It returns the documentElement as an Element Object.
74.	<b>doctype</b> This property returns the DTD (Document Type Declaration) that are associated with the current HTML document.
75.	<b>designMode</b> It helps us to determine if the entire document is editable or not.
76.	<b>defaultView</b> It is used for returning document's window object.
77.	<b>cookie</b> The HTML DOM document cookie property is used for creating, reading and deleting cookies.
78.	<b>charset</b>

	It returns the character encoding of the HTML document.
79.	<b>applets</b> It is used to return a list of all the applets elements within a document but this property is now deprecated.
80.	<b>characterSet</b> This property is used to get character encoding of the document.
81.	<b>anchors</b> The anchors property is a read only property which lists all the "a" tag with name attribute in the document.
82.	<b>baseURI</b> It is used to return the base Uniform Resource Identifier (URI) of the document.
83.	<b>length</b> This method returns the number of tokens in a token list.
84.	<b>value</b> This method returns the DOMTokenList serialized as a string.
85.	<b>domain</b> It is used for getting the domain name of the server on which the document is currently being executed.

## Frequently Asked Questions about DOM

There are several Frequently Asked Questions(FAQ) about DOM, this section tries to answer some of them briefly.

What is the full form of DOM?



The full form of DOM is Document Object Model.

Why DOM is used in JavaScript?



In JavaScript, DOM is used to interact with HTML web pages and manipulate the content, structure, and styles of the page.

Is DOM a programming language?



No, DOM is not a programming language. It is a programming interface that connects web pages to programming languages.

The DOM interfaces provide various methods and properties to interact with and manipulate web pages. Some common interfaces are Document, Element, Event and so on.