

Document Object Model DOM and Validations

Module 1 - Introduction & Structure of the DOM

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Document Object Model

- The DOM (Document Object Model) is a tree-like structure representing HTML elements of a web page.
- It allows JavaScript to access, modify, add, or delete content and structure dynamically.
- Each part of the webpage (like headings, paragraphs, buttons) becomes a node in the DOM.
- JavaScript uses the DOM to make pages interactive and responsive to user actions.

Purpose of Dom in JS

- The DOM (Document Object Model) represents the webpage as a tree of elements/nodes.
- It allows JavaScript to access and select HTML elements on the page.
- Enables JavaScript to read and modify content of elements (text, HTML).
- Allows changing the style and attributes of HTML elements dynamically.
- Supports adding new elements or removing existing ones from the webpage.
- Enables handling user events like clicks, mouse movements, keyboard input, etc.
- Facilitates the creation of interactive and dynamic web pages.
- Acts as a bridge between JavaScript and the webpage structure.

Why is DOM Important?

- Allows JavaScript to interact with an HTML page.
- Enables dynamic content updates without reloading the page.
- Helps in handling events like button clicks, form submissions, etc.
- Allows manipulation of CSS styles and HTML elements dynamically.

What is XML?

- XML (eXtensible Markup Language) is a way to store and organize data using tags, similar to HTML.
- Unlike HTML, which focuses on how data is displayed, XML focuses on what data represents.

**Course & Test Series**



**Introduction to 'JavaScript' Language**

**How Browsers Create the DOM from HTML/XML**

1. Loading the Document
  - Browser reads HTML or XML from the server or local file.
2. Tokenization
  - Browser breaks the code into tokens (e.g., <html>, <body>, <p>).
3. Lexing (Parsing Tokens)
  - Tokens are analyzed for meaning (e.g., start tag, end tag, text).
4. Tree Construction
  - Browser builds a DOM Tree, where:
    - <html> becomes the root.
    - Nested tags become child nodes.
    - Text becomes leaf nodes.
5. Attachment to Rendering Engine
  - DOM is then passed to the rendering engine to display content visually.

**Example of DOM Tree Creation**

For HTML:

```
html
<html>
  <body>
    <h1>Hello</h1>
  </body>
</html>
```

Browser makes a DOM Tree like:

- html
  - body
    - h1
      - "Hello" (text node)

**Important Terms**

**Tokenization:**

- What it is:

The process of breaking down the HTML or XML code into small chunks called tokens.

- Tokens are:

Pieces of code like start tags (<p>), end tags (</p>), attributes (class="title"), or text (Hello).

**Lexing**

- What it is:

The process of analyzing each token to figure out its meaning (type).

- What browser learns:

- <p> is an opening paragraph tag.
- class="intro" is an attribute of <p>.
- Hello is text content.
- </p> is a closing paragraph tag.

**Rendering**

Rendering is the process where the browser takes the DOM (structure of the page), applies styles (CSS), and then draws everything on the screen so you can see it.



**Course & Test Series**

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking & Insurance**

 **Central Govt. Service**

 **State Govt. Services**

 **LAW Entrance**

 **MBA Entrance**

 **Railways & Metro Services**

...many more

**abhyasonline.in**

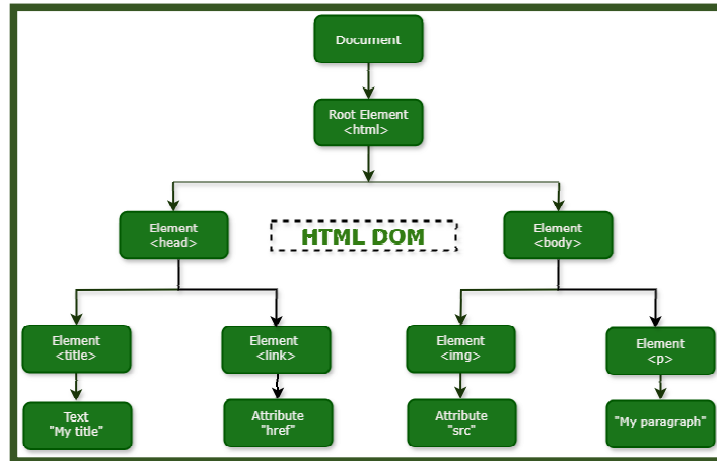
**Introduction to 'JavaScript' Language**

**DOM Tree**

The DOM Tree in JavaScript refers to the Document Object Model (DOM) structure that represents the HTML content of a web page as a tree of objects. This model allows JavaScript to interact with and manipulate the content, structure, and style of a webpage dynamically.

**Root**

- The top-most node of the DOM tree.
- Represents the entire document.
- In JavaScript, it's accessed using document.
- It's the starting point from which all other nodes (elements, text, etc.) branch out.



**Elements**

- Represent HTML tags (like <div>, <p>, <a>, etc.).
- Element nodes are the building blocks of the webpage.
- Each element can contain:
  - Other elements (nested)
  - Text nodes
  - Attribute nodes

**Nodes**

- Nodes are the general term for everything in the DOM tree.
- Every part of an HTML document is a node, including:
  - Element nodes (HTML tags)
  - Text nodes (the content inside tags)
  - Comment nodes (<!-- comments -->)
  - Attribute nodes (e.g., id="demo")
- All elements are nodes, but not all nodes are elements.

**Course  
&  
Test Series**

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking &  
Insurance**

 **Central Govt.  
Service**

 **State Govt.  
Services**

 **LAW  
Entrance**

 **MBA  
Entrance**

 **Railways & Metro  
Services**

**...many more**





**abhyasonline.in**

**Introduction to 'JavaScript' Language**

```
<!DOCTYPE html>
<html>
<head>
  <title>DOM Example</title>
</head>
<body>
  <h1 id="heading">Hello</h1>
  <p>This is a paragraph.</p>

  <script>
    const root = document;
    const element = document.getElementById("heading");
    const textNode = element.firstChild;
    console.log(root);
    console.log(element);
    console.log(textNode);
  </script>
</body>
</html>
```

**HTML Source Code**





Feature	Description
 <b>Definition</b>	The raw <b>HTML code</b> you write in your .html file.
 <b>Where it's from</b>	Comes directly from the <b>server</b> or file.
 <b>How to view it</b>	Right-click → <b>View Page Source</b> in browser.
 <b>Static</b>	It doesn't reflect any <b>JavaScript changes</b> .

```
<!DOCTYPE html>
<html>
<body>
  <p>Hello</p>
</body>
</html>
```

Course  
&  
Test Series

Introduction to 'JavaScript' Language

DOM Source Code

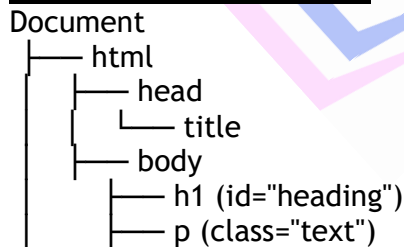
Feature	Description
 <b>Definition</b>	A <b>tree structure</b> the browser creates from the HTML.
 <b>Dynamic</b>	JavaScript can <b>read and modify</b> it.
 <b>How to view it</b>	Right-click → <b>Inspect</b> (DevTools).
 <b>Reflects changes</b>	If JS modifies content, the DOM updates in real time.

```
<!DOCTYPE html>
<html>
  <body>
    <p>Hello, World!</p> <!-- Changed by JS -->
  </body>
</html>
```

Solved Example HTML Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>DOM Example</title>
</head>
<body>
  <h1 id="heading">Welcome to DOM</h1>
  <p class="text">This is a paragraph.</p>
</body>
</html>
```

DOM Tree Representation



-  **CBSE**
-  **ICSE**
-  **NTSE**
-  **Banking & Insurance**
-  **Central Govt. Service**
-  **State Govt. Services**
-  **LAW Entrance**
-  **MBA Entrance**
-  **Railways & Metro Services**
- ...many more

Course  
&  
Test Series

Introduction to 'JavaScript' Language

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Parent-Child-Sibling Relationships

**Parent**

- A node that has one or more child nodes.
- Example: In `<div><p>Hello</p></div>`, the `<div>` is the parent of `<p>`.

**Child**

- A node that is directly inside another node (nested).
- In the same example, `<p>` is a child of `<div>`.

**Siblings**

- Nodes that share the same parent.
- Example: In `<div><h1>Hi</h1><p>Hello</p></div>`, both `<h1>` and `<p>` are siblings.

