DOM

What is the DOM?

The DOM is a W3C (World Wide Web Consortium) standard. The DOM defines a standard for accessing documents like HTML and XML.
The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.

The DOM is separated into 3 different parts / levels:

1) Core DOM - standard model for any structured document
2) XML DOM - standard model for XML documents
3) HTML DOM - standard model for HTML documents

The DOM defines the **objects and properties** of all document elements, and the **methods** (interface) to access them.

The HTML DOM

- A standard object model for HTML
- A standard programming interface for HTML
- Platform- and language-independent
- A W3C standard

The HTML DOM defines the **objects and properties** of all HTML elements, and the **methods** (interface) to access them. In other words the HTML DOM defines a standard way for accessing and manipulating HTML documents

In the DOM, everything in a HTML document is a node.

A node in an HTML document is:

1) The Document –The entire document
2) An element –Entire elements of a document is an element node
3) An attribute – Attributes of element nodes is an attribute node
4) Text –Text in html elements are text nodes
5) A comment –comment in html is comment nodes.

Dom Example

```
<html>
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
```

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

The root node in the HTML above is <html>. All other nodes in the document are contained within <html>.The <html> node has two child nodes; <head> and <body>.The <head> node holds a <title> node. The <body> node holds a <h1> and <p> node.

Text is Always Stored in Text Nodes

A common error in DOM processing is to expect an element node to contain text.However, the text of an element node is stored in a text node.In this example: **<title>DOM</title>**, the element node <title>, holds a text node with the value "DOM" "DOM" is **not** the value of the <title> element.However, in the HTML DOM the value of the text node can be accessed by the **innerHTML** property.

The HTML DOM Node Tree

The HTML DOM views an HTML document as a tree-structure. The tree structure is called a **node-tree.**All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created.The node tree below shows the set of nodes, and the connections between them. The tree starts at the root node and branches out to the text nodes at the lowest level of the tree:

Node Parents, Children, and Siblings

The nodes in the node tree have a hierarchical relationship to each other.The terms parent, child, and sibling are used to describe the relationships. Parent nodes have children. Children on the same level are called siblings (brothers or sisters).

1) In a node tree, the top node is called the root
2) Every node, except the root, has exactly one parent node
3) A node can have any number of children
4) A leaf is a node with no children
5) Siblings are nodes with the same parent

Node Object Properties:

Property	Description
attributes	Returns a collection of a node's attributes
baseURI	Returns the absolute base URI of a node
childNodes	Returns a NodeList of child nodes for a node
firstChild	Returns the first child of a node
lastChild	Returns the last child of a node
localName	Returns the local part of the name of a node

namespaceURI	Returns the namespace URI of a node
nextSibling	Returns the next node at the same node tree level
nodeName	Returns the name of a node, depending on its type
nodeType	Returns the type of a node
nodeValue	Sets or returns the value of a node, depending on its type
ownerDocument	Returns the root element (document object) for a node
parentNode	Returns the parent node of a node
prefix	Sets or returns the namespace prefix of a node
previousSibling	Returns the previous node at the same node tree level
textContent	Sets or returns the textual content of a node and its descendants

Node Object Methods

Method	**Description**
appendChild()	Adds a new child node, to the specified node, as the last child node
cloneNode()	Clones a node
compareDocumentPosition()	Compares the document position of two nodes
getFeature(feature,version)	Returns a DOM object which implements the specialized APIs of the specified feature and version
getUserData(key)	Returns the object associated to a key on a this node. The object must first have been set to this node by calling setUserData with the same key
hasAttributes()	Returns true if a node has any attributes, otherwise it returns false
hasChildNodes()	Returns true if a node has any child nodes, otherwise it returns false
insertBefore()	Inserts a new child node before a specified, existing, child node
isDefaultNamespace()	Returns true if the specified namespaceURI is the default, otherwise false
isEqualNode()	Checks if two nodes are equal
isSameNode()	Checks if two nodes are the same node
isSupported()	Returns true if a specified feature is supported on a node, otherwise false
lookupNamespaceURI()	Returns the namespace URI matching a specified prefix
lookupPrefix()	Returns the prefix matching a specified namespace URI
normalize()	Joins adjacent text nodes and removes empty text nodes

removeChild()	Removes a child node
replaceChild()	Replaces a child node
setUserData(key,data,handler)	Associates an object to a key on a node

Node Types

Documents, elements, attributes, and other aspects of an HTML document has different node types.There are 12 different node types, which may have children of various node types:

Node	Description	Children
Element	Represents an element	Element, Text, Comment, ProcessingInstruction, CDATASection, EntityReference
Attr	Represents an attribute	Text, EntityReference
Text	Represents textual content in an element or attribute	None
CDATASection	Represents a CDATA section in a document (text that will NOT be parsed by a parser)	None
EntityReference	Represents an entity reference	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
Entity	Represents an entity	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
ProcessingInstruction	Represents a processing instruction	None
Comment	Represents a comment	None
Document	Represents the entire document (the root-node of the DOM tree)	Element, ProcessingInstruction, Comment, DocumentType
DocumentType	Provides an interface to the entities defined for the document	None
DocumentFragment	Represents a "lightweight" Document object, which can hold a portion of a document	Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference
Notation	Represents a notation declared in the DTD	None

Node Types - Return Values

Node Type	nodeName returns	nodeValue returns
Element	element name	Null
Attr	attribute name	attribute value
Text	#text	content of node
CDATASection	#cdata-section	content of node
EntityReference	entity reference name	Null
Entity	entity name	Null
ProcessingInstruction	target	content of node
Comment	#comment	comment text
Document	#document	Null
DocumentType	doctype name	Null
DocumentFragment	#document fragment	Null
Notation	notation name	Null

NodeTypes - Named Constants

NodeType	Named Constant
1	ELEMENT_NODE
2	ATTRIBUTE_NODE
3	TEXT_NODE
4	CDATA_SECTION_NODE
5	ENTITY_REFERENCE_NODE
6	ENTITY_NODE
7	PROCESSING_INSTRUCTION_NODE
8	COMMENT_NODE
9	DOCUMENT_NODE
10	DOCUMENT_TYPE_NODE
11	DOCUMENT_FRAGMENT_NODE
12	NOTATION_NODE

Node Properties

Three important node properties are:

- nodeName
- nodeValue
- nodeType

The nodeName Property

The nodeName property specifies the name of a node.

- nodeName is read-only
- nodeName of an element node is the same as the tag name
- nodeName of an attribute node is the attribute name

- nodeName of a text node is always #text
- nodeName of the document node is always #document

Note: nodeName always contains the uppercase tag name of an HTML element.

The nodeValue Property

The nodeValue property specifies the value of a node.

- nodeValue for element nodes is undefined
- nodeValue for text nodes is the text itself
- nodeValue for attribute nodes is the attribute value

The nodeType Property

The nodeType property returns the type of node. nodeType is read only.
The most important node types are:

Element type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9

Accessing Nodes

You can access a node in three ways:

1) By using the getElementById() method
2) By using the getElementsByTagName() method
3) By navigating the node tree, using the node relationships

Examples: For Accessing the contents of the document- Displaying

Example 1: Using getElementById ()

```
<html>
  <head>
    <title>Exampel-DOM</title>
  <script>
    var pval= document.getElementById("para").innerText();
    document.writeln(pval);
  </script>
  </head>
  <body>
    <h1>Display using getElementById()</h1>
    <p id="para">Hello world!</p>
  </body>
</html>
```

Example 2: Using getElementsByTagName ()

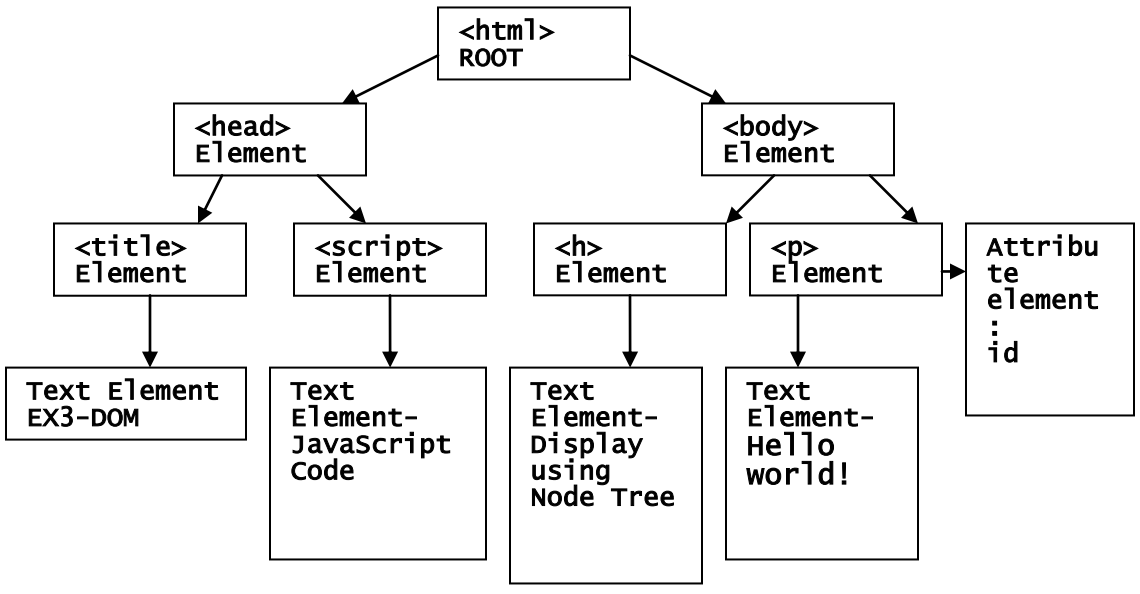
```
<html>
  <head>
```

```
    <title>Exampe2-DOM</title>
  <script>
    var x = document.getElementsByTagname("p");
    document.writeln(x[0].innerText());
  </script>
  </head>
  <body>
    <h1>Display using getElementsByTagname()</h1>
    <p>Hello world!</p>
  </body>
</html>
```

Example 3: Using Node Tree Concept

```
<html>
  <head>
    <title>Ex3-DOM</title>
  <script>
    var x = document.getElementById("para");
    document.writeln(x.firstChild.nodeValue());
  </script>
  </head>
  <body>
    <h1>Display using Node Tree</h1>
    <p id="para" >Hello world!</p>
  </body>
</html>
```

The Node Tree the above Example is




```
    var x = document.getElementById("para");
    document.writeln(x.firstChild.nodeValue());
```

The javascript code document.getElementById("para"); returns the element <p> which is refered by x.

```
Document.write(x.firstChild.nodeValue());
```

firstChild of x is a Text Element whose nodeValue is "Hello world!", which is displayed by write function of document.

DOM Root Nodes

There are two special document properties that allow access to the tags:

- ***document.documentElement*** - returns the root node of the document
- ***document.body*** - gives direct access to the <body> tag

DOM Node List Length

The length property defines the number of nodes in a node-list.

Example 4: length property of Node

```
<html>
  <head>
    <title>Ex3-DOM</title>
  <script>
    var x = document.getElementsByTagname("p");
    for ( I = 0 ; I < x.length ; i++)
       document.writeln(x[ I i.innerText());
  </script>
  </head>
  <body>
    <h1>Display using Node Tree</h1>
    <p>Hello world!</p>
    <p>Hello Vitians</p>
</body>
</html>
```

Change an HTML Element

HTML elements can be changed using JavaScript, the HTML DOM and events.

Examples : For changing the content of the document using DOM properties and method.

Example 1: Change the text of a HTML element- innerHTML

```
<html>
  <head>
    <title>Ex3-DOM</title>
  <script language="javascript">
    document.getElementById("para").innerHTML = "Hello vitians";
  </script>
  </head>
  <body>
    <h1>Display using Node Tree</h1>
    <p id="para" onclick=" ">Hello world!</p>
  </body>
</html>
```

Example 2: Change the back ground color of document

```
<html>
  <head>
    <title>Ex3-DOM</title>
  <script>
    document.getElementById("b").bgColor="aqua";
  </script>
  </head>
  <body id ="b">
    <h1>Display using Node Tree</h1>
    <p id="para" >Hello world!</p>
  </body>
</html>
```

Example 3 Change the back ground color of document

```
<html>
  <head>
    <title>DOM</title>
  <script>
    document.body.bgColor="aqua";
  </script>
  </head>
  <body>
    <h1>Changing body color</h1>
  </body>
</html>
```

Example 4:Change the style of <p> element of document on a click event

```
<html>
  <head>
    <title>DOM</title>
  <script>

    Function changeStyle()
    {
    document.getElementById("para").style.color="aquamarine";
    document.getElementById("para").style.fontfamily="arial";
   }
  </script>
  </head>
  <body>
    <h1>Changing Style</h1>
    <p id="para" >Hello world!</p>
    <input type="button" name="change" value="change"
                         onclick="changeStyle() />
  </body>
</html>
```