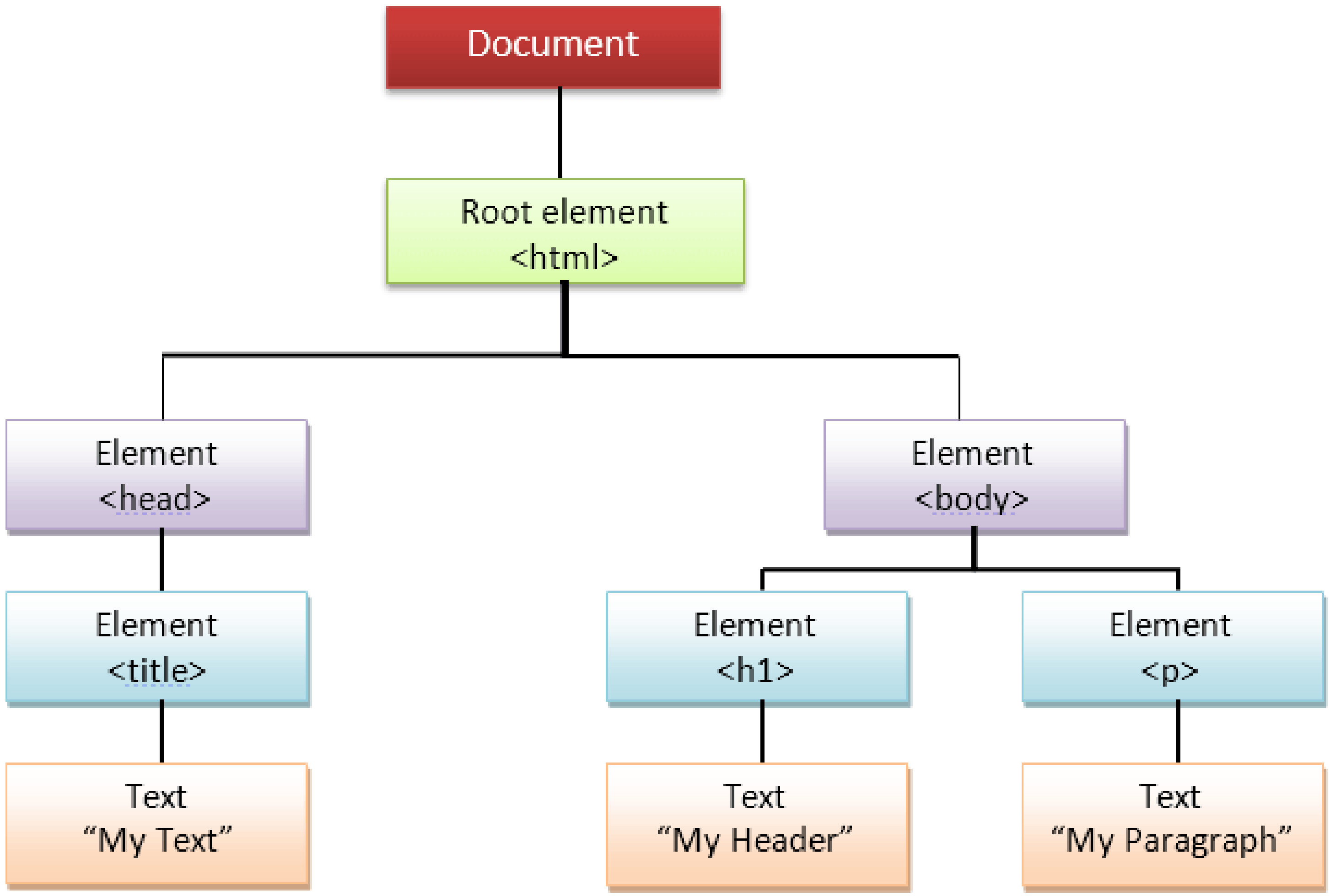
HTML DOM

What is DOM?

- **D**ocument **O**bject **M**odel
  - Constructed as a tree of objects
  - Represents a web page
- JavaScript gets all the power it needs to create dynamic HTML
  - JavaScript can change
    - All Elements, Attributes, CSS
    - remove existing HTML elements and attributes
    - Add existing HTML elements and attributes
    - new HTML events in the page


Document
Root element <html>
Element <head>
Element <body>
Element <title>
Element <h1>
Element <p>
Text "My Text"
Text "My Header"
Text "My Paragraph"

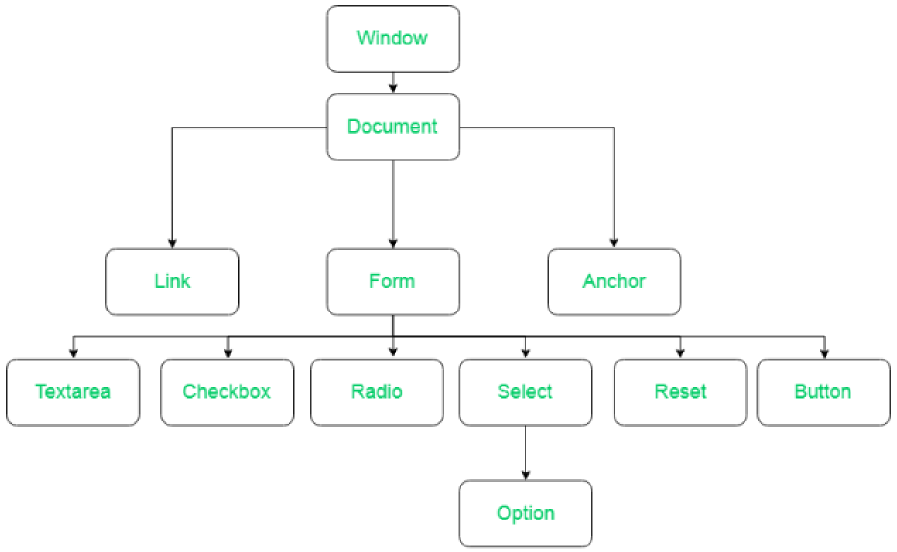
DOM

- A W3C (World Wide Web Consortium) standard.
  - Defines a standard for accessing documents
  - defines the **logical structure** of documents and the way a document is accessed and manipulated by an application program
    - Logical structure because DOM doesn't specify any relationship between objects.

"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."

Why DOM is required?

- HTML is used to **structure** the web pages and JavaScript is used to add **behavior** to our web pages.
- When an HTML file is loaded into the browser, the JavaScript can not understand the HTML document directly.
- So, a corresponding document is created(DOM).
- **DOM is basically the representation of the same HTML document but in a different format with the use of objects**.



What DOM is not?

- not used to describe objects in XML or HTML whereas the DOM **describes XML and HTML documents as objects.**
- not represented by a set of data structures; it is an interface that **specifies object representation**.
- does not show the criticality of objects in documents i.e it doesn’t have information about which object in the document is appropriate to the context and which is not.

Understanding the DOM

- a platform and language independent model to represent the HTML or XML documents
- all parts of the document, such as elements, attributes, text, etc. are organized in a hierarchical tree-like structure
- individual parts of the document are known as **nodes**
- The Document Object Model that represents HTML document is referred to as HTML DOM
- The DOM that represents the XML document is referred to as XML DOM.

HTML DOM

- a standard object model and programming interface for HTML
- It defines:
  - The HTML elements as objects
  - The properties of all HTML elements
  - The methods to access all HTML elements
  - The events for all HTML elements

- a standard for how to get, change, add, or delete HTML elements.

HTML DOM

- DOM methods are **actions** you can perform (on HTML Elements).

- DOM properties are **values** (of HTML Elements) that you can set or change.

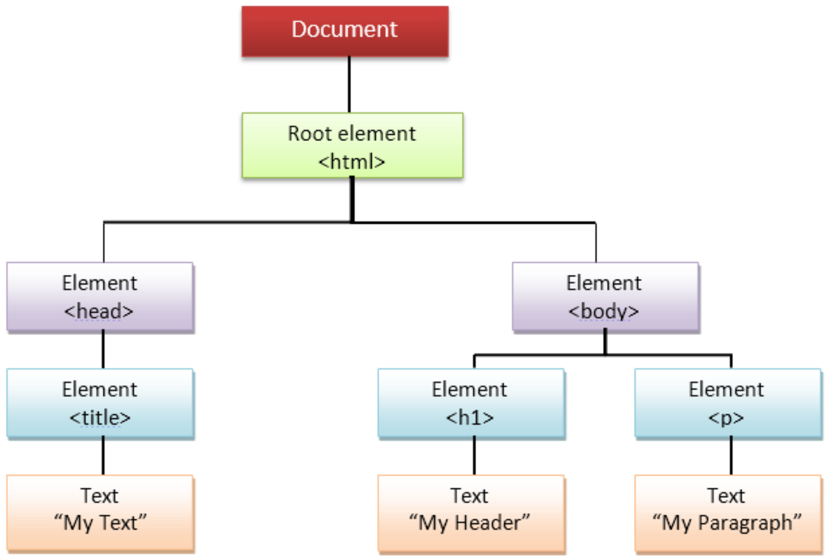
DOM

- Documents are modeled using objects
- The model includes not only the structure of a document but also the behavior of a document and the objects of which it is composed of like tag elements with attributes in HTML
- With DOM one can use JavaScript
  - to build HTML documents
  - navigate their hierarchical structure, and
  - add, modify, or delete elements and attributes or their content
  - so on. . .

Example

```
<!DOCTYPE html>
<html>
<head>
   <title>My Page</title>
</head>
<body>
   <h1>Human Body</h1>
   <ul>
      <li>Circulatory System</li>
      <li>Respiratory System</li>
      <li>Nervous System</li>
   </ul>
</body>
</html>
```

- parent/child relationships between the nodes
- topmost node - the Document node is the root node of the DOM tree
- <head> and <body> siblings
- Everything is node
  - Elements, comments
  - HTML attributes such as id, class, title, style, etc. - are accessed as properties of the element node that contains them.



Understanding the hierarchy

- Window Object - Always at top of the hierarchy.
- Document object - When an HTML document is loaded into a window, it becomes a document object.
- Form Object - Represented by *form* tags
- Link Object - Represented by *link* tag
- Anchor Object - Represented by *a href* tags
- Form Control Elements - Form can have many control elements such as text fields, buttons, radio buttons, and checkboxes, etc.

Understanding the hierarchy

- topmost elements in an HTML document are available directly as document properties
  - document.documentElement
- <head> - can be accessed with document.head property
- <body> - an be accessed with document.body property

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>JS Select Topmost Elements</title>
</head>
<body>
<p>PAragraph1</p>
   <script>
   // Display lang attribute value of html element
   alert(document.documentElement.getAttribute("lang")); // O/P: en

   // Set background color of body element
   document.body.style.background = “blue";

   // Display tag name of the head element's first child
   alert(document.head.firstElementChild.nodeName); // O/P: title
   </script>
</body>
</html>
```

```
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JS Select Element by ID</title>
</head>
<body>
    <p id = “p1">This is a paragraph of text.</p>
    <p>This is another paragraph of text.</p>

    <script>
    // Selecting element with id mark
    var m1 = document.getElementById(“p1");
     m1.innerHTML = “text changed…”;

    // Highlighting element's background
    m1.style.background = “cyan";
    </script>
</body>
</html>
```

getElementById - will return the element as an object if the matching element was found else `null`

Methods of Document Object

1. write(“string”): Writes the given string on the document.
2. getElementById(): returns the element having the given id value.
3. getElementsByName(): returns all the elements having the given name value.
4. getElementsByTagName(): returns all the elements having the given tag name.
5. getElementsByClassName(): returns all the elements having the given class name.

DOM Programming Interface

- In DOM, all HTML elements are objects.
- Methods and properties are available
  - **property** is a value that you can get or set (like changing the content of an HTML element)
  - **method** is an action you can do (like add or deleting an HTML element)

```
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

Example: Method and property

- getElementById()
  - common way to access an HTML element is to use the id of the element
- innerHTML
  - way to get the content of an element (getter and setter)

Selecting by class name

```
<p class="test">This is a paragraph of text.</p>

<script>
   // Selecting elements with class test
   var arr = document.getElementsByClassName("test");

   // Displaying the selected elements count
   document.write("Number of selected elements: " + arr.length);

// Applying bold style to first element in selection
arr[0].style.fontWeight = "bold";

<script>
```

Selecting by tag name

```
<p class="test">This is a paragraph of text.</p>

<script>
   // Selecting elements with class test
   var arr = document.getElementsByTagName(“p");

   // Displaying the selected paragraphs count
   document.write("Number of selected elements: " + arr.length);

// Applying bold style to first element in selection
arr[0].style.background = “cyan";

<script>
```