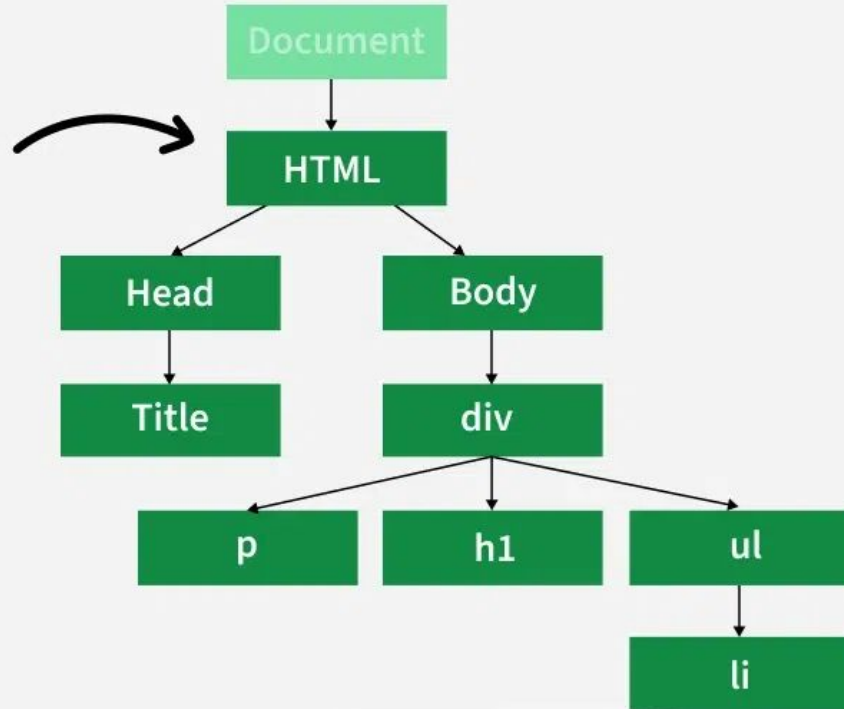

<Semester II>

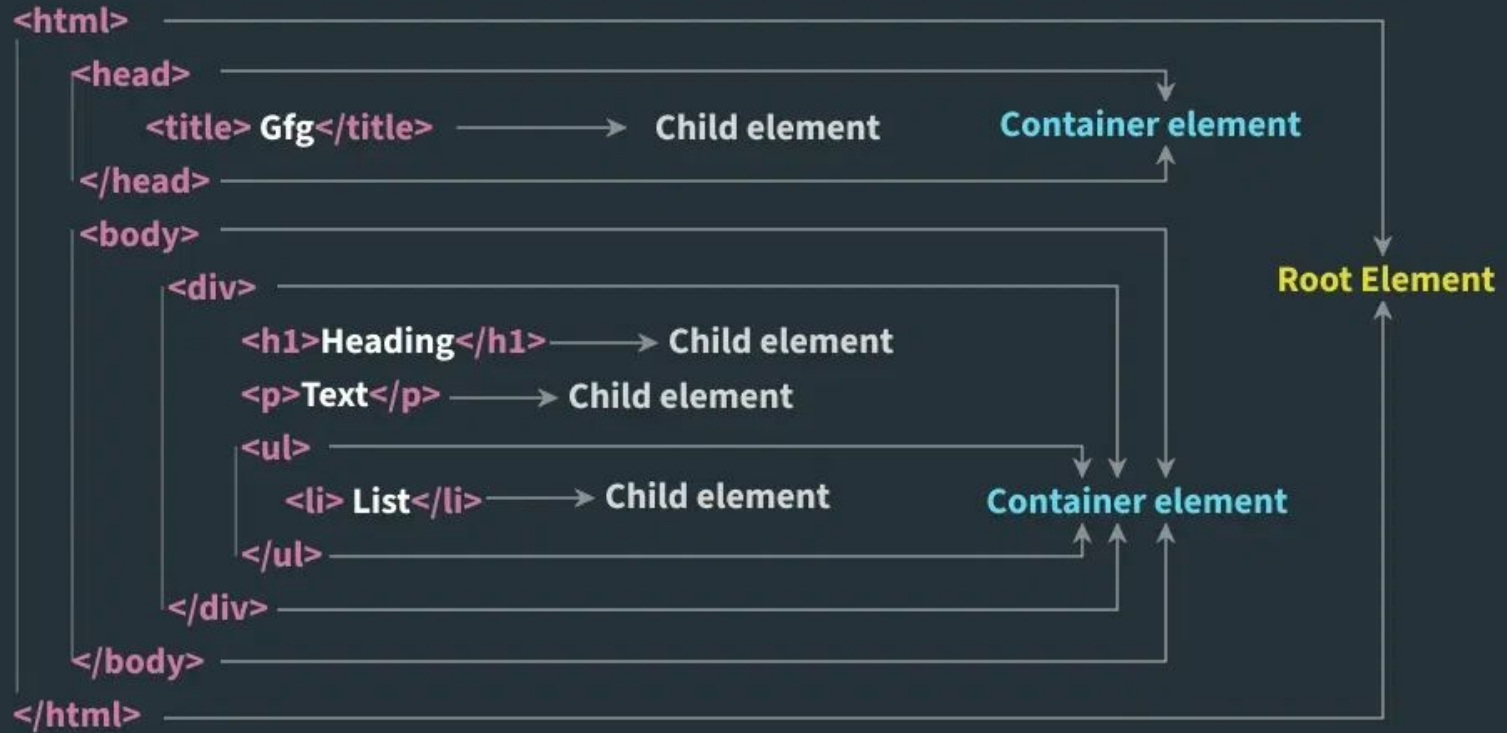
<//Programming for Interactive Interfaces//>

Week 11

```
<html>
<head>
  <title> Webpage Title</title>
</head>
<body>
  <div>
    <h1>This is Heading Tag</h1>
    <p>Some Text Content</p>
    <ul>
      <li> List Item</li>
    </ul>
  </div>
</body>
</html>
```

The “DOM Tree”





DOM Tree Navigation

In JavaScript, with HTML DOM we can navigate the tree nodes using the relationship between the nodes. Each HTML element is represented as a node in the DOM tree. The HTML document object is represented as root node. There are different types of nodes such as root node, parent, child and sibling nodes. The relationship between these nodes help to navigate the DOM tree.

Navigating between nodes means finding the parent, child, sibling, etc. element of a particular element in the DOM tree using JavaScript. Then, elements can be created, removed, and appended exactly where one wants it in the DOM hierarchy. Even the attributes of the HTML tags can be added, modified, deleted, and checked.

DOM Tree Navigation

In the HTML DOM, everything is a node.

Root node – The document node is a root node.

Parent node – Each node has a single parent node.

Child node – Each node can have multiple and nested childrenders.

Sibling node – The sibling nodes are at the same level, have the same parent node

Difference between an Element Node and non-Element Node:

A node can be an element, attribute, text, comment, document, or any other type of DOM object. Elements are a specific type of node that represents an HTML element. Elements have all the properties and methods of a node, but they also have additional properties and methods that are specific to elements. Eg: elements have a `tagName` property that specifies the name of the element, such as “div” or “span”. Elements also have attributes, which can be accessed using the `getAttribute()` method or by simply accessing the attribute as a property of the element.

DOM Navigation Properties

Property	Description
firstChild	To get the first child of the particular node. It can also return the text, comment, etc.
firstElementChild	To get the first child element. For example, <p>, <div>, , etc.
lastChild	To get the last child of the particular node. It can also return the text, comment, etc.
lastElementChild	To get the last child element.
childNodes	To get the node list of all children of the particular element.
children	To get the HTML collection of all children of the particular element.

DOM Navigation Properties

Property	Description
parentNode	To get the parent node of the HTML element.
parentElement	To get the parent element node of the HTML element.
nextSibling	To get the next node from the same level having the same parent node.
nextElementSibling	To get the next element node from the same level having the same parent node.
previousSibling	To get the previous node from the same level having the same parent node.
previousElementSibling	To get the previous element node from the same level having the same parent node.

classList Methods and Properties

```
element.classList.add("class-name");
```

```
element.classList.remove("class-name");
```

```
element.classList.contains("class-name");
```

```
element.classList.toggle("class-name");
```

```
element.classList.item(index);
```

```
element.classList.length;
```

```
element.classList.replace("old-cname", "new-cname");
```


DOM Manipulation

Method	Description
<code>createElement()</code>	Used to create the HTML element. The element specified using <code>elementName</code> is created
<code>append()</code>	Adds content as the last child of the selected element
<code>prepend()</code>	Adds content as the first child of the selected elements
<code>insertBefore()</code>	Inserts a child node before an existing child. The new node will be above the existing node in the DOM
<code>appendChild()</code>	Appends a node (element) as the last child of an element.
<code>remove()</code>	The <code>remove()</code> method removes an element (or node) from the document.

DOM Manipulation

Method	Description
<code>removeChild()</code>	The <code>removeChild()</code> method removes an element's first child.
<code>replaceChild()</code>	The <code>replaceChild()</code> method replaces a child node with a new node.
<code>hasAttribute()</code>	The <code>hasAttributes()</code> method returns true if a node has attributes, otherwise false. The <code>hasAttributes()</code> method always returns false if the node is not an element node.
<code>getAttribute()</code>	The <code>getAttribute()</code> method returns the value of an element's attribute.
<code>setAttribute()</code>	The <code>setAttribute()</code> method sets a new value to an attribute. If the attribute does not exist, it is created first.
<code>removeAttribute()</code>	The <code>removeAttribute()</code> method removes an attribute from an element.



**Homework: Integrate JavaScript into your LCA2
assignment code.**

W3Schools JavaScript Tutorial
JavaScript for Web | MDN

Reference



1. [HTML DOM \(Document Object Model\) | GeeksforGeeks](#)
 2. [What Is the Difference Between Element and Node in DOM | by Brandon Evans | Frontend Weekly | Medium](#)
 3. [JavaScript DOM Navigation | Tutorials Point](#)
 4. [HTML DOM Element classList Property & Methods - W3Schools](#)
 5. [Creating and removing HTML elements with JavaScript | by Tom Hendrych | Medium](#)
 6. [How do you add, remove, and modify HTML elements using JavaScript? | Quiz Interview Questions with Solutions](#)
-

Fonts & colors used

This presentation has been made using the following fonts:

Blinker

(<https://fonts.google.com/specimen/Blinker>)

Inconsolata

(<https://fonts.google.com/specimen/Inconsolata>)

#ffffff

#022a46

#72ffdd

#72fff

#0d3a58