SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

Course Code and Name : 19TS601 FULL STACK DEVELOPMENT

Unit 1 : JAVASCRIPT AND BASICS OF MERN STACK

Topic : Node Properties

Node properties

- In JavaScript, when working with the Document Object Model (DOM), a "node" typically refers to an element or component of the HTML document.
- Each node in the DOM can have several properties and methods associated with it.

some common node properties in JavaScript:

1. nodeName

Represents the name of the node (e.g., for an element node, it's the tag name).

Example:

```
const element = document.getElementById('myElement');
  console.log(element.nodeName); // Outputs: 'DIV'
  (or the tag name of the node)
```

- **2. nodeType**
- Provides the type of the node (e.g., 1 for element nodes, 3 for text nodes, etc.).
- Example:

```
const element = document.getElementById('myElement');
console.log(element.nodeType); // Outputs: 1 (Element node)
```

- **3. nodeValue**
- Represents the value of the node. For element nodes, it’s null, but for text nodes, it contains the actual text.
- Example:
- const textNode = document.createTextNode('Hello World'); console.log(textNode.nodeValue); // Outputs: 'Hello World'

- **4. parentNode**
- References the parent node of the current node.
- Example:
- const element = document.getElementById('myElement'); console.log(element.parentNode); // Outputs: the parent node of the element

5. childNodes

Returns a NodeList of the child nodes of the current node (includes all types of nodes, such as text nodes).

Example:

```
const element =
  document.getElementById('myElement');
  console.log(element.childNodes); // Outputs a
  NodeList of child nodes
```

6. firstChild

Refers to the first child node of the current node.

Example:

```
const element =
  document.getElementById('myElement');
  console.log(element.firstChild); // Outputs the first
  child node
```

- **7. lastChild**
- Refers to the last child node of the current node.
- Example:
- const element = document.getElementById('myElement');
  console.log(element.lastChild); // Outputs the last child node

8. nextSibling

Points to the next sibling node of the current node.

Example:

```
const element =
  document.getElementById('myElement');
  console.log(element.nextSibling); // Outputs the next
  sibling node
```

- **9. previousSibling**
- Points to the previous sibling node of the current node.
- Example:
- const element = document.getElementById('myElement');
  console.log(element.previousSibling); // Outputs the previous siblir
  node

- **10. attributes**
- Returns a NamedNodeMap of all attributes of an element node.
- Example:
- const element = document.getElementById('myElement'); console.log(element.attributes); // Outputs all attributes of the element
- These properties are typically used when manipulating the DOM, allowing you to navigate, access, and modify nodes.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Node Properties Example</title></head>
<body>
<div id="parent">
 <h1 id="header">Hello, World!</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p></div>
<script src="script.js"></script>
</body>
</html>
```

```
// Get the parent div element by its ID
const parentNode = document.getElementById('parent');
// Log the nodeName (should be 'DIV')
console.log('Parent Node Name:', parentNode.nodeName);
 // Log the nodeType (should be 1 for an element node)
console.log('Parent Node Type:', parentNode.nodeType);
// Get the first child node (which should be the <h1> element)
const firstChild = parentNode.firstChild;
console.log('First Child Node:', firstChild);
// Log the nodeValue of the first child (should be null because it's an element node)
console.log('First Child Node Value:', firstChild.nodeValue);
```

```
// Get the list of child nodes (should include both <h1> and <p> elements)
const childNodes = parentNode.childNodes;
console.log('Child Nodes:', childNodes);
// Get the next sibling of the header (which should be the first <p> element)
const header = document.getElementById('header');
const nextSibling = header.nextSibling;
console.log('Next Sibling of Header:', nextSibling);
// Get the attributes of the header element
const headerAttributes = header.attributes;
console.log('Attributes of Header:', headerAttributes);
 // Traverse through all child nodes and log their nodeName and nodeType
console.log('Iterating over child nodes:');
childNodes.forEach((node, index) => { console.log(`Child Node ${index + 1} - Name: ${node.nodeName}, Type: ${node.nodeType}`); });
```

Parent Node Name: DIV

Parent Node Type: 1

First Child Node: [object Text]

First Child Node Value: null

Child Nodes: NodeList(5) [text, h1, text, p, text]

Next Sibling of Header: [object Text]

Attributes of Header: NamedNodeMap {id: id="header"}

Iterating over child nodes:

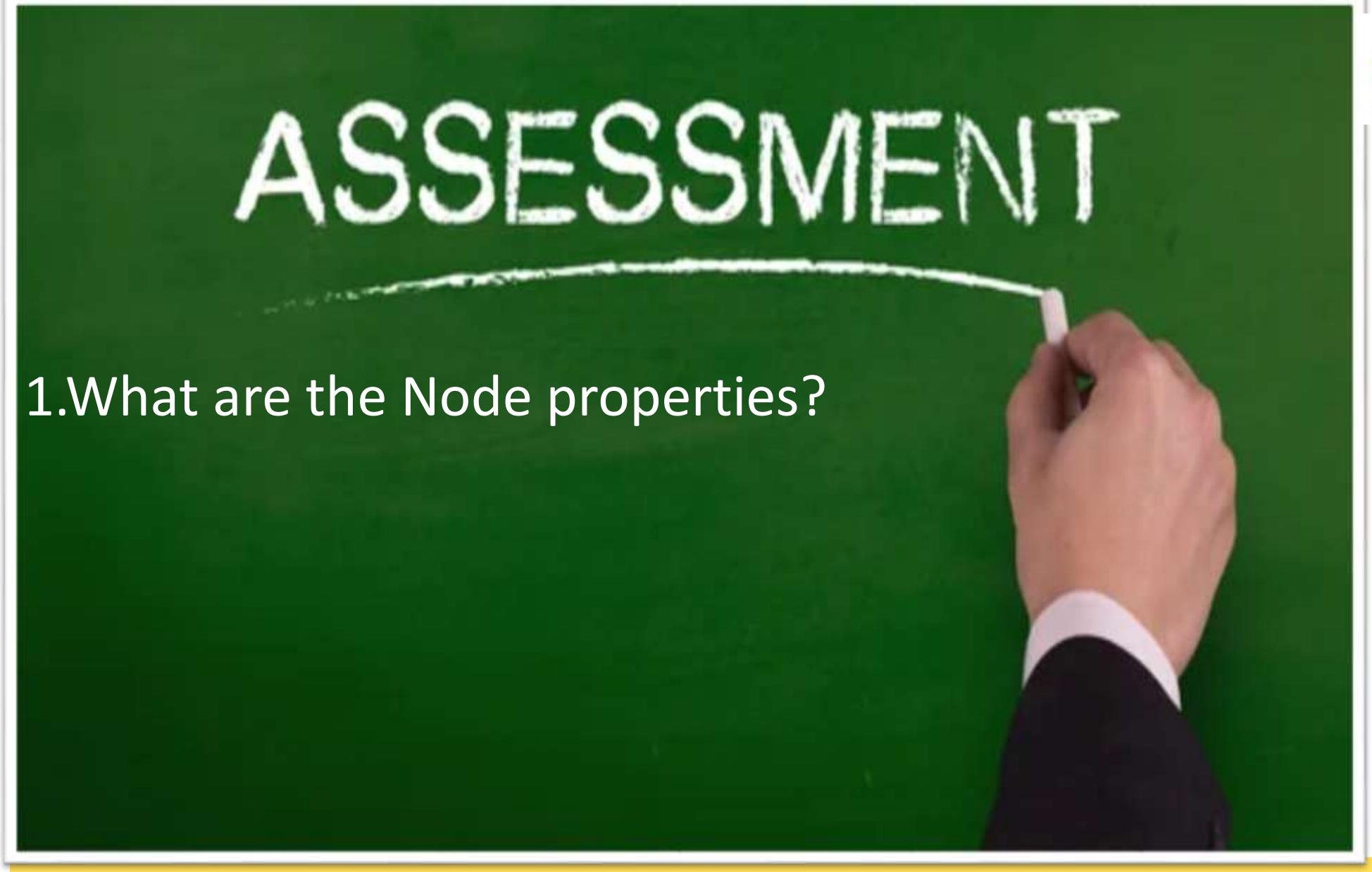
Child Node 1 - Name: #text, Type: 3

Child Node 2 - Name: H1, Type: 1

Child Node 3 - Name: #text, Type: 3

Child Node 4 - Name: P, Type: 1

Child Node 5 - Name: #text, Type: 3



1.What are the Node properties?

Text Book:

1.Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, A Press Publisher, 2019.

Reference:

David Flanagan, “Java Script: The Definitive Guide”, O’Reilly Media, Inc, 7 th Edition, 2020

2. Matt Frisbie, “Professional JavaScript for Web Developers” Wiley Publishing, Inc, 4th Edition, ISBN: 978-1-119-36656-0, 2019


Thank
You!