NODE RELATIONSHIPS

IN THIS LESSON

Accessing DOM nodes in JavaScript through node relationships

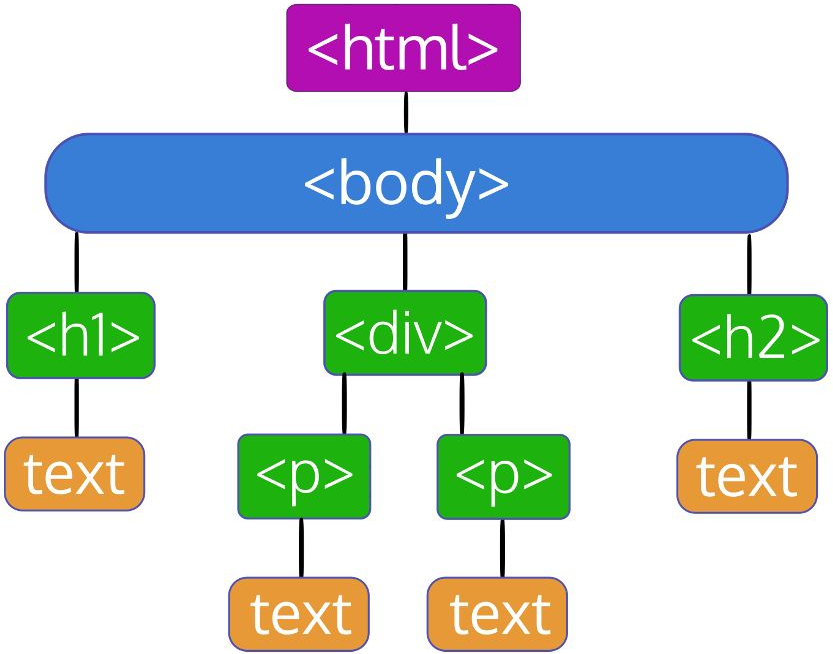
Node relationship properties:
.parentNode, .firstChild, lastChild
.nextSibling, .previousSibling
.childNodes[]

A demonstration of traversing the DOM

DOM nodes can be described in parent/ child/sibling relationships

This concept provides us with an interface we can use to reference nodes in relation to each other (in JavaScript as well as other programming languages)

```
let node = document.body;
```



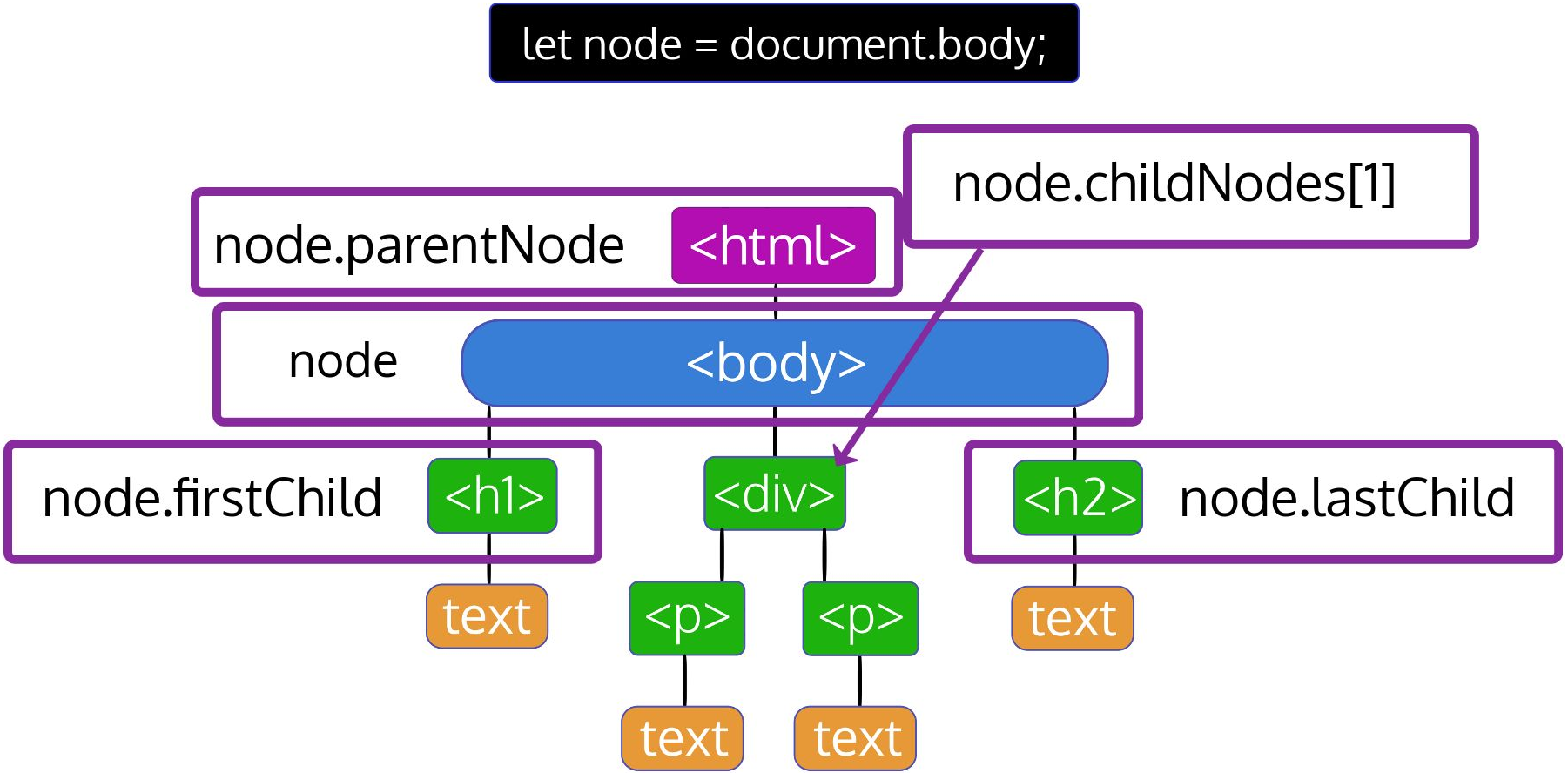
.parentNode - gives the parent of a node

.firstChild & **.lastChild** - first and last child of a node

.nextSibling - very next sibling of a node

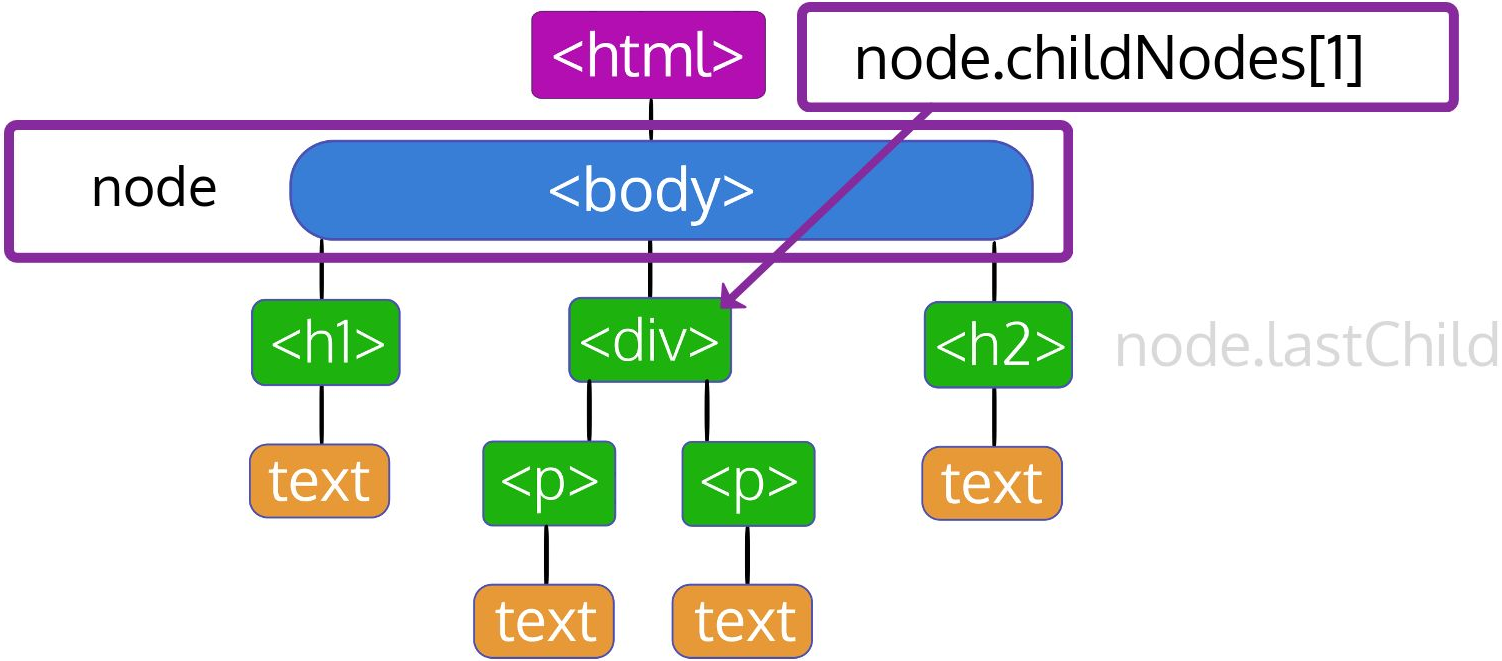
.previousSibling - immediate previous sibling of a node

.childNodes[...] will give an iterable called a NodeList, containing all child nodes of a node - use bracket notation with index to access each child in the list, e.g. **.childNodes[0]** for the first child, **.childNodes[1]** for the second child, etc.

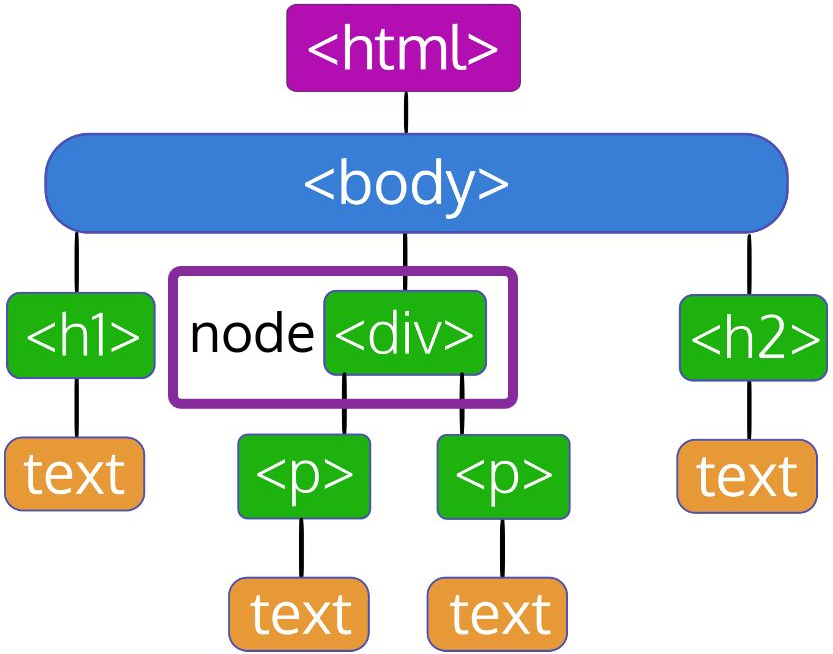

let node = document.body;
node.childNodes[1]
node.parentNode
<html>
node
<body>
node.firstChild
<h1>

<h2>
node.lastChild
text
<p>
<p>
text
text
text

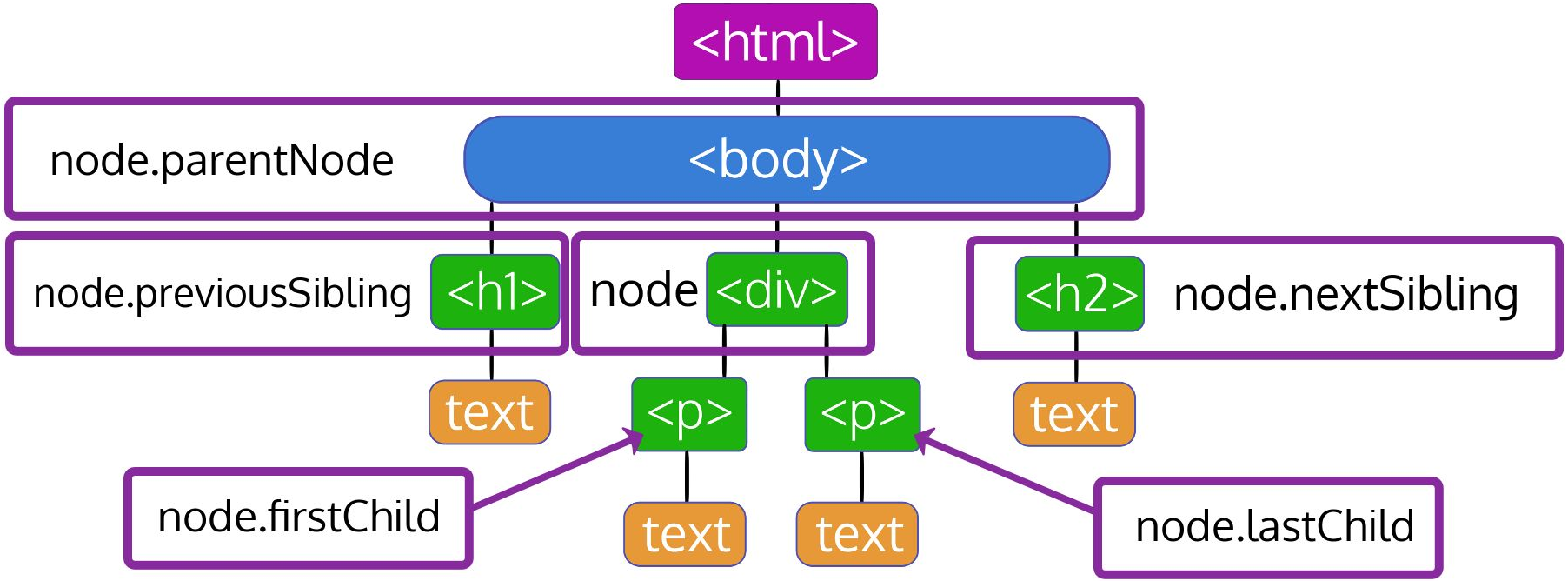
```
let node = document.body;
node = node.childNodes[1];
```



```
let node = document.body;
node = node.childNodes[1];
```



```
let node = document.body;
node = node.childNodes[1];
```



node-relationships.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>DOM Relationships Demo</title>
    <style>
        table, table * {
            border: 2px solid;
            padding: 10px;
        }
        td {
            background: palegreen;
        }
        body {
            font: 1.2em "Source Sans Pro", Arial, sans-serif;
        }
    </style>
</head>
<body>
    <h1>DOM Relationships Demo</h1>
    <p>
        Click on any element to see the DOM path to its node
    </p>
    <h2>LISTS & ANCHORS</h2>
    <ul>
        <li><a href="">Unordered Listitem Anchor</a></li>
```

node-relationships.html

```
</head>
<body>
    <h1>DOM Relationships Demo</h1>
    <p>
        Click on any element to see the DOM path to its node
    </p>
    <h2>LISTS & ANCHORS</h2>
    <ul>
        <li><a href="">Unordered Listitem Anchor</a></li>
        <li>Unordered Listitem 4
            <ol>
                <li><a href="">Nested Ordered Listitem Anchor</a></li>
                <li>Nested Ordered Listitem 2</li>
            </ol>
        </li>
    </ul>
    <h2>TABLE</h2>
    <table>
        <tbody>
            <tr>
                <td>Cell 1</td>
                <td>Cell 2</td>
                <td>Cell 3</td>
            </tr>
        </tbody>
    </table>
```

node-relationships.html

```
    </table>

    <script>
        function handleClick(event) {
            event.stopPropagation();
            let node = event.target;
            let thisPath = node.nodeName;
            while (node.parentNode) {
                node = node.parentNode;
                thisPath = node.nodeName + " > " + thisPath;
            }
            alert(thisPath);
        }

        function attachHandler(node) {
            if (node === null) {
                return;
            }
            node.onclick = handleClick;
            for (child of node.childNodes) {
                attachHandler(child);
            }
        }
        attachHandler(document.body);
    </script>
</body>
```

```
<script>
    function handleClick(event) {
        event.stopPropagation();
        let node = event.target;
        let thisPath = node.nodeName;
        while (node.parentNode) {
            node = node.parentNode;
            thisPath = node.nodeName + " > " + thisPath;
        }
        alert(thisPath);
    }

    function attachHandler(node) {
```

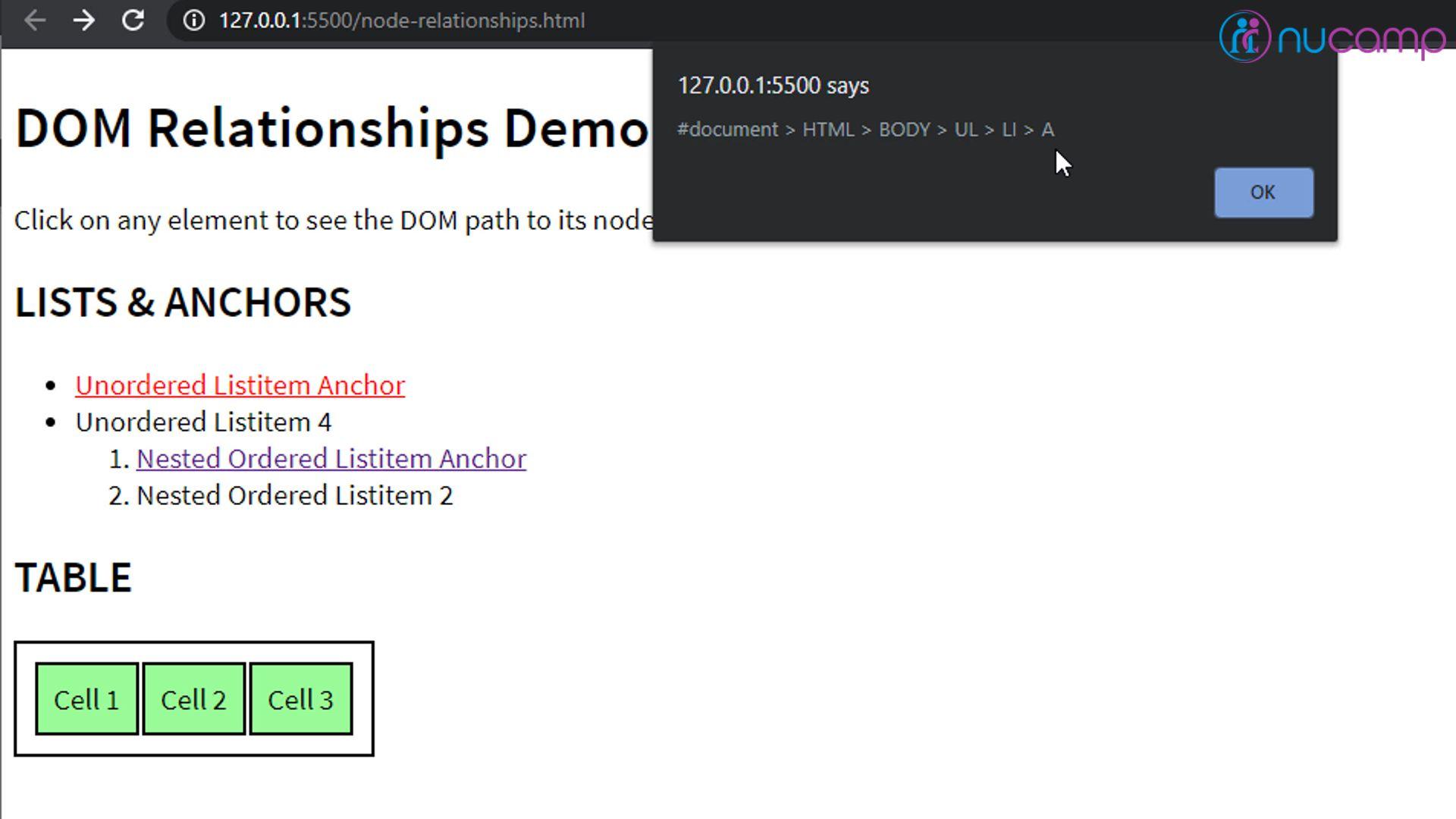
```
    }

    function attachHandler(node) {
        if (node === null) {
            return;
        }
        node.onclick = handleClick;
        for (child of node.childNodes) {
            attachHandler(child);
        }
    }
    attachHandler(document.body);
</script>
</body>
```

```
    }

    function attachHandler(node) {
        if (node === null) {
            return;
        }
        node.onclick =
        for (child of n                des) {
            attachHandler(child);
        }
    }
    attachHandler(document.body);
</script>
ody>
```

recursion


127.0.0.1:5500/node-relationships.html
DOM Relationships Demo
Click on any element to see the DOM path to its node
127.0.0.1:5500 says
#document > HTML > BODY > UL > LI > A
OK
LISTS & ANCHORS
Unordered Listitem Anchor
Unordered Listitem 4
Nested Ordered Listitem Anchor
Nested Ordered Listitem 2
TABLE
Cell 1
Cell 2
Cell 3

127.0.0.1:5500/node-relationships.html

127.0.0.1:5500 says

#document > HTML > BODY > H2

OK

DOM Relationships Demo

Click on any element to see the DOM path to its node

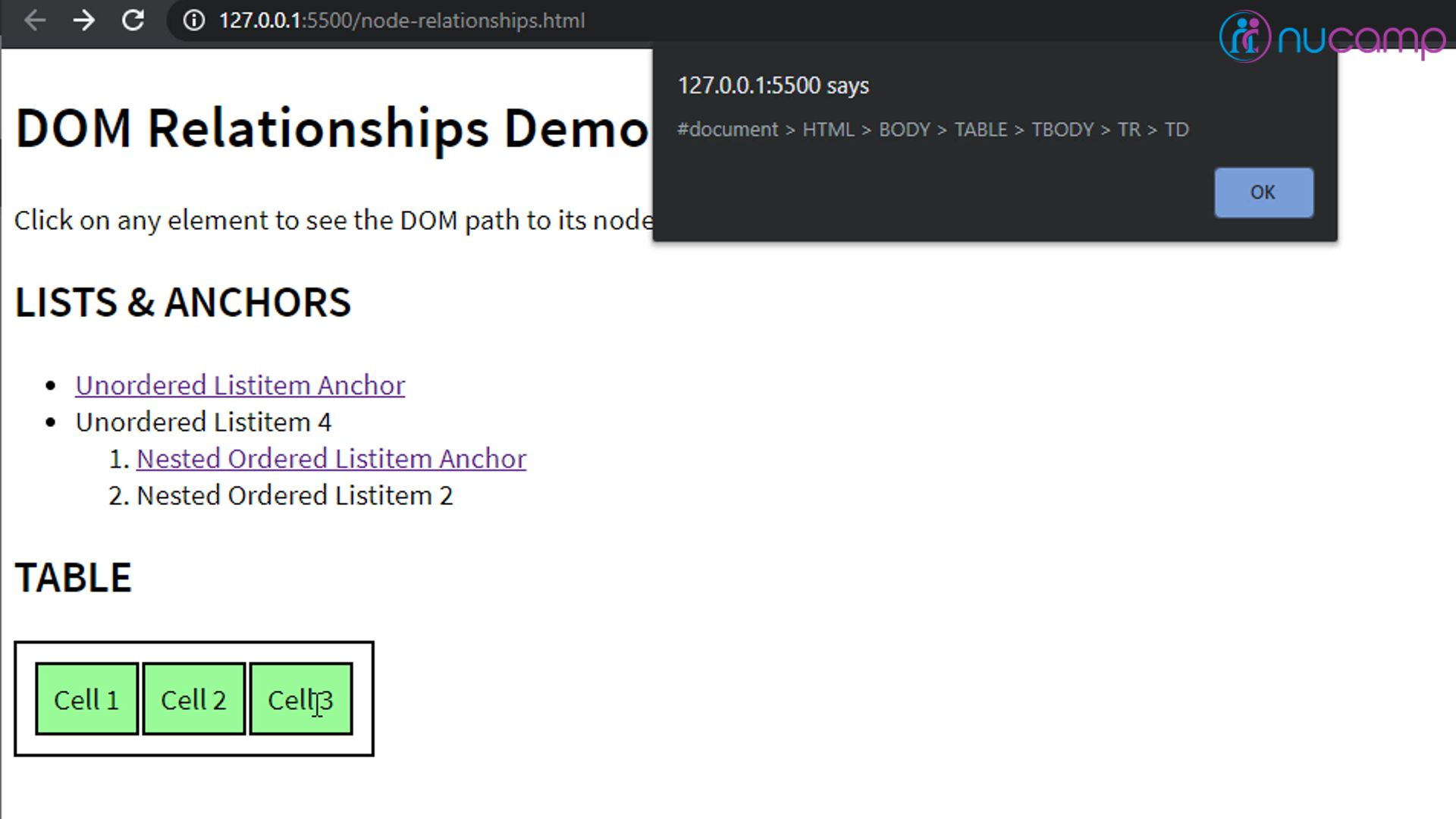
LISTS & ANCHORS

- Unordered Listitem Anchor
- Unordered Listitem 4
    1. Nested Ordered Listitem Anchor
    2. Nested Ordered Listitem 2

TABLE

Cell 1	Cell 2	Cell 3

127.0.0.1:5500/node-relationships.html



DOM Relationships Demo

Click on any element to see the DOM path to its node

LISTS & ANCHORS

- Unordered Listitem Anchor
- Unordered Listitem 4
    1. Nested Ordered Listitem Anchor
    2. Nested Ordered Listitem 2

TABLE

Cell 1	Cell 2	Cell 3