JS JavaScript TUTORIAL

JavaScript Siblings

Summary: in this tutorial, you will learn how to select the next siblings, previous siblings, and all siblings of an element.

Let’s say you have the following list of items:

```
<ul id="menu">
    <li>Home</li>
    <li>Products</li>
    <li class="current">Customer Support</li>
    <li>Careers</li>
    <li>Investors</li>
    <li>News</li>
    <li>About Us</li>
</ul>
```

Get next siblings

To get the next sibling of an element, you use the `nextElementSibling` attribute:

```
let nextSibling = currentNode.nextElementSibling;
```

The `nextElementSibling` returns `null` if the specified element is the last one in the list.

The following example uses the `nextElementSibling` property to get the next sibling of the list item that has the `current` class:

```
let current = document.querySelector('.current');
let nextSibling = current.nextElementSibling;

console.log(nextSibling);
```

Output:

```
<li>Careers</li>
```

In this example:

- First, select the list item whose class is `current` using the `querySelector()` .
- Second, get the next sibling of that list item using the `nextElementSibling` property.

To get all the next siblings of an element, you can use the following code:

```
let current = document.querySelector('.current');
let nextSibling = current.nextElementSibling;

while(nextSibling) {
    console.log(nextSibling);
    nextSibling = nextSibling.nextElementSibling;
}
```

Get previous siblings

To get the previous siblings of an element, you use the `previousElementSibling` attribute:

```
let current = document.querySelector('.current');
let prevSibling = currentNode.previousElementSibling;
```

The `previousElementSibling` property returns `null` if the current element is the first one in the list.

The following example uses the `previousElementSibling` property to get the previous siblings of the list item that has the `current` class:

```
let current = document.querySelector('.current');
let prevSiblings = current.previousElementSibling;
```

```
console.log(prevSiblings);
```

The following example selects all the previous siblings of the list item that have the `current` class:

```
let current = document.querySelector('.current');
let prevSibling = current.previousElementSibling;
while(prevSibling) {
    console.log(prevSibling);
    prevSibling = current.previousElementSibling;
}
```

Get all siblings of an element

To get all siblings of an element, we'll use the logic:

- First, select the element's parent whose siblings you want to find.
- Second, select the first child element of that parent element.
- Third, add the first element to an array of siblings.
- Fourth, select the next sibling of the first element.
- Finally, repeat the 3rd and 4th steps until no siblings are left. In case the sibling is the original element, skip the 3rd step.

The following function illustrates the steps:

```
let getSiblings = function (e) {
    // for collecting siblings
    let siblings = [];
    // if no parent, return no sibling
    if(!e.parentNode) {
        return siblings;
    }
    // first child of the parent node
    let sibling  = e.parentNode.firstChild;

    // collecting siblings
    while (sibling) {
        if (sibling.nodeType === 1 && sibling !== e) {
            siblings.push(sibling);
        }
        sibling = sibling.nextSibling;
    }
    return siblings;
};
```

Put it all together:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>JavaScript Siblings</title>
</head>
<body>
    <ul id="menu">
        <li>Home</li>
        <li>Products</li>
        <li class="current">Customer Support</li>
        <li>Careers</li>
        <li>Investors</li>
        <li>News</li>
        <li>About Us</li>
    </ul>

    <script>
        let getSiblings = function (e) {
            // for collecting siblings
            let siblings = [];
            // if no parent, return no sibling
            if(!e.parentNode) {
                return siblings;
            }
            // first child of the parent node
            let sibling  = e.parentNode.firstChild;
            // collecting siblings
            while (sibling) {
                if (sibling.nodeType === 1 && sibling !== e) {
                    siblings.push(sibling);
                }
                sibling = sibling.nextSibling;
            }
            return siblings;
        };

        let siblings = getSiblings(document.querySelector('.current'));
        siblingText = siblings.map(e => e.innerHTML);
        console.log(siblingText);
    </script>
</body>
</html>
```

Output:

```
["Home", "Products", "Careers", "Investors", "News", "About Us"]
```

Summary

- The `nextElementSibling` returns the next sibling of an element or `null` if the element is the last one in the list.
- The `previousElementSibling` returns the previous sibling of an element or `null` if the element is the first one in the list.
- To get all siblings of an element, you can use a helper function that utilizes the `nextElementSibling` property.

Quiz