



Creating, Removing & Replacing Elements | JavaScript Tutorial In Hindi #16

[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Creating, Removing & Replacing Elements | JavaScript Tutorial In Hindi #16

In the previous lecture, we explored a number of ways to select the HTML element. In this tutorial, we will study how to create new elements and manipulate existing elements on the page, such as removing elements from the page, replacing elements with new elements. This tutorial will help us master all the JavaScript techniques that we need to manipulate elements in the DOM.

createElement():-

The **document.createElement()** is a method used to create the HTML element. **createElement()**. Following is the syntax.

```
var element= document.createElement("name");
```

In the above syntax, the name is passed as a parameter. The name specifies the type of the created element. The document.createElement() returns the newly created element.

The following example uses the document.createElement() to create a new <div> element:

```
let div = document.createElement('div');
```

There is also another method, which is createTextNode(). This method creates a Text Node with the specified text. Use the createElement() method to create an element Node with the specified name. After the Text Node is created, use the *element.appendChild()* method to append it to an element.

```
var paragraph = document.createElement("P");  
var text = document.createTextNode("This is a paragraph.");
```

appendChild()-

The **appendChild()** method is used to create a text node as the node's last child. Appending in JavaScript is a way to insert content to the end of already existing elements. To append in Javascript, we use the `appendChild()` method.

Syntax:

```
node.appendChild( node )
```

Here is an example:

```
var paragraph = document.createElement("P");
var text = document.createTextNode("This is a paragraph.");
paragraph.appendChild(text);
```

Working with Attributes:-

The attributes are the special words used inside the start tag of an HTML element. JavaScript provides us with several methods for adding, removing or changing an HTML element's attribute. Following are the some methods to set attributes in HTML element.

setAttribute() :-

The `setAttribute()` method is used to set an attribute on the specified element. Using this method, a new attribute is added with the specified name and value. If the attribute already exists on the element, the value is updated. Here is an example:

In this example, we add a href attribute with a value of "https://codewithharry.com/" to an <a> element:

```
document.getElementById("myAnchor").setAttribute("href", "https://codewithharry.com/");
```

getAttribute() :-

The `getAttribute()` method is used to get the current value of a attribute on the specified element. If the attribute does not exist on the element, it will return null.

Here is an example:

In this example, we will get the value of the target attribute of an <a> element:

```
var h = document.getElementById("myAnchor").getAttribute("target");
```

hasAttribute()-

The `hasAttribute()` method is used to check an element has a specified attribute or not. The syntax is :

```
let result = element.hasAttribute(name);
```

In the argument, we specifies the name of the attribute that we want to check.

This method returns a Boolean value that indicates if the element has the specified attribute. If the element contains an attribute, it will return true; otherwise, it will return false. Here is an example:

In this example, we find that if the <button> element has an onclick attribute:

```
var h = document.getElementById("Btn").hasAttribute("onclick");
```

removeAttribute() :-

The removeAttribute() method is used to remove an attribute from the specified element. The difference between this method and the removeAttributeNode() method is that the removeAttributeNode() method removes the specified *Attr object*, while this method removes the attribute with the specified *name*. The result will be the same. Also this method has no return value, while the removeAttributeNode() method returns the removed attribute as an Attr object.

Remove the href attribute from an <a> element:

```
document.getElementById("myAnchor").removeAttribute("href");
```

replaceWith():-

The replaceWith() method replaces this ChildNode in the children list of its parent with a set of Node. Here is an example:

```
let element1 = document.getElementById("myid1 ");
let element2 = document.createElement("u");
let content = document.createTextNode("Added Content");
element2.appendChild(content);
element1.replaceWith(element2);
```

We are getting the element using its id. Then we are creating a new element which creates an underlined content and then we adds content to the newly created element. In next step we are adding the content to the element2. And finally we are replacing the old element, with the newly created element(old_elem.replaceWith(new_elem);).

replaceChild():-

The replaceChild() method replaces a child node with a new node. We can create a new node, or The new node could be an existing node in the document or the new node could be an existing node in the document. In the following example, we will replace an existing node sp2 with the new span element sp1:

```
parentDiv.replaceChild(sp1, sp2);
```

removeChild():-

This method removes a specified child node of the specified element. It will return the removed node as a Node object, or *null* if the node does not exist. Remember that the removed child node is no longer part of the DOM. In the following example, we will find out if a list has any child nodes. If so, remove its second child node.

```
let list = document.getElementById("myList");
if (list.hasChildNodes()) {
```

```
list.removeChild(list.childNodes[1]);  
}
```

Website.html code as described/written in the video

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <meta http-equiv="X-UA-Compatible" content="ie=edge">  
  <title>Document</title>  
</head>  
<body>  
  <div class="container">  
    <h1 id="heading"> Welcome to Code With Harry</h1>  
    <div id="myfirst" class="child red good" id="first">child 1  
  
      <ul class="this">  
        <li class="childul">this</li>  
        <li class="childul">is</li>  
        <li class="childul">a</li>  
        <li class="childul">list </li>  
        <li class="childul">of my dreams</li>  
      </ul>  
    </div>  
    <div class="child">child 2</div>  
    <div class="child red">child 3</div>  
    <div class="child">child 4</div>  
    <form action="none.html" method="post">  
      <a href="//codewithharry.com">Go to Code With Harry</a>  
      <br>  
      <br>  
      Search this website: <input type="text" name="Hello" id="">  
      <input type="button" value="submit">  
    </form>
```

```
</div>
<br>
<div class="no">this is a dummy div1</div>
<div class="no">this is a dummy div2</div>
<div class="no">this is a dummy div3</div>
</body>
<!-- <script src="js/tut12.js"></script> -->
<!-- <script src="js/tut14.js"></script> -->
<script src="js/tut15.js"></script>
</html>
```

JavaScript code as described/written in the video

```
console.log('this is tut 16');
let element = document.createElement('li');
let text = document.createTextNode('I am a text node');
element.appendChild(text)

// Add a class name to the li element
element.className = 'childul';
element.id = 'createdLi';
element.setAttribute('title', 'mytitle');
// element.innerText = '<b>Hello this is created by harry</b>';
// element.innerHTML = '<b>Hello this is created by harry</b>';

let ul = document.querySelector('ul.this');
ul.appendChild(element);
// console.log(ul)
// console.log(element)

let elem2 = document.createElement('h3');
elem2.id = 'elem2';
elem2.className = 'elem2';
let tnode = document.createTextNode('This is a created node for elem2');
elem2.appendChild(tnode);

element.replaceWith(elem2);
let myul = document.getElementById('myul');
```

[Copy](#)

```
myul.replaceChild(element, document.getElementById('fui'));
myul.removeChild(document.getElementById('lui'));
let pr = elem2.hasAttribute('href');
elem2.removeAttribute('id');
elem2.setAttribute('title', 'myelem2title');
console.log(elem2, pr);

// quick quiz
// create a heading element with text as "Go to CodeWithHarry" and create an a tag outside it with href = "https://www.codewithharry.c
```

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

