

30 Days Of JavaScript: Manipulating DOM Object




LinkedIn



Follow @asabeneh

Author: [Asabeneh Yetayeh](#)

January, 2020

[<< Day 21](#) | [Day 23 >>](#) Thirty Days Of JavaScript

- [Day 22](#)
 - [DOM\(Document Object Model\)-Day 2](#)
 - [Creating an Element](#)
 - [Creating elements](#)
 - [Appending child to a parent element](#)
 - [Removing a child element from a parent node](#)
 - [Exercises](#)
 - [Exercises: Level 1](#)
 - [Exercises: Level 2](#)
 - [Exercises: Level 3](#)

Day 22

DOM(Document Object Model)-Day 2

Creating an Element

To create an HTML element we use tag name. Creating an HTML element using JavaScript is very simple and straight forward. We use the method `document.createElement()`. The method takes an HTML element tag name as a string parameter.

```
// syntax
document.createElement('tagname')
```

```
<!DOCTYPE html>
<html>
```

```
<head>
  <title>Document Object Model:30 Days Of JavaScript</title>
</head>
```

```
<body>

  <script>
    let title = document.createElement('h1')
    title.className = 'title'
    title.style.fontSize = '24px'
    title.textContent = 'Creating HTML element DOM Day 2'

    console.log(title)
  </script>
</body>

</html>
```

Creating elements

To create multiple elements we should use loop. Using loop we can create as many HTML elements as we want. After we create the element we can assign value to the different properties of the HTML object.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Document Object Model:30 Days Of JavaScript</title>
  </head>

  <body>

    <script>
      let title
      for (let i = 0; i < 3; i++) {
        title = document.createElement('h1')
        title.className = 'title'
        title.style.fontSize = '24px'
        title.textContent = i
        console.log(title)
      }
    </script>
  </body>

</html>
```

Appending child to a parent element

To see a created element on the HTML document we should append it to the parent as a child element. We can access the HTML document body using *document.body*. The *document.body* support the *appendChild()* method. See the example below.

```
<!DOCTYPE html>
<html>

<head>
  <title>Document Object Model:30 Days Of JavaScript</title>
</head>

<body>

  <script>
    // creating multiple elements and appending to parent element
    let title
    for (let i = 0; i < 3; i++) {
      title = document.createElement('h1')
      title.className = 'title'
      title.style.fontSize = '24px'
      title.textContent = i
      document.body.appendChild(title)
    }
  </script>
</body>
</html>
```

Removing a child element from a parent node

After creating an HTML, we may want to remove element or elements and we can use the *removeChild()* method.

Example:

```
<!DOCTYPE html>
<html>

<head>
  <title>Document Object Model:30 Days Of JavaScript</title>
</head>

<body>
  <h1>Removing child Node</h1>
  <h2>Asabeneh Yetayeh challenges in 2020</h2>
  <ul>
    <li>30DaysOfPython Challenge Done</li>
    <li>30DaysOfJavaScript Challenge Done</li>
    <li>30DaysOfReact Challenge Coming</li>
    <li>30DaysOfFullStack Challenge Coming</li>
    <li>30DaysOfDataAnalysis Challenge Coming</li>
    <li>30DaysOfReactNative Challenge Coming</li>
    <li>30DaysOfMachineLearning Challenge Coming</li>
  </ul>
```

```

<script>
  const ul = document.querySelector('ul')
  const lists = document.querySelectorAll('li')
  for (const list of lists) {
    ul.removeChild(list)
  }
</script>
</body>

</html>

```

As we have seen in the previous section there is a better way to eliminate all the inner HTML elements or the children of a parent element using the method *innerHTML* properties.

```

<!DOCTYPE html>
<html>

<head>
  <title>Document Object Model:30 Days Of JavaScript</title>
</head>

<body>
  <h1>Removing child Node</h1>
  <h2>Asabeneh Yetayeh challenges in 2020</h2>
  <ul>
    <li>30DaysOfPython Challenge Done</li>
    <li>30DaysOfJavaScript Challenge Done</li>
    <li>30DaysOfReact Challenge Coming</li>
    <li>30DaysOfFullStack Challenge Coming</li>
    <li>30DaysOfDataAnalysis Challenge Coming</li>
    <li>30DaysOfReactNative Challenge Coming</li>
    <li>30DaysOfMachineLearning Challenge Coming</li>
  </ul>

  <script>
    const ul = document.querySelector('ul')
    ul.innerHTML = ''
  </script>
</body>

</html>

```

The above snippet of code cleared all the child elements.

🟡 You are so special, you are progressing everyday. Now, you knew how to destroy a created DOM element when it is needed. You learned DOM and now you have the capability to build and develop

applications. You are left with only eight days to your way to greatness. Now do some exercises for your brain and for your muscle.

Exercises

Exercises: Level 1

1. Create a div container on HTML document and create 100 to 100 numbers dynamically and append to the container div.
 - Even numbers background is green
 - Odd numbers background is yellow
 - Prime numbers background is red

 Number Generator


Exercises: Level 2


1. Use the countries array to display all the countries. See the design

 World Countries List

Exercises: Level 3

Check the requirement of this project from both images(jpg and gif). All the data and CSS has been implemented using JavaScript only. The data is found on starter folder project_3. The drop down button has been created using *details* HTML element.

 Challenge Information

 Challenge Information

🎉 CONGRATULATIONS ! 🎉

<< [Day 21](#) | [Day 23](#) >>