

Lesson

Tuesday

# Introduction to Programming

## (/introduction-to-programming)

### / JavaScript and Web Browsers

## (/introduction-to-programming/javascript-and-web-browsers)

### / Accessing HTML Element Attributes and Properties in the DOM

Text

Cheat sheet

## Accessing DOM Element attributes

In this lesson we learned how to access HTML DOM elements to get and set the values of attributes:

- In the DOM, the HTML elements from our source code are transformed into HTML DOM element objects. Like all objects, an HTML DOM element is a collection of properties. These properties represent things like:
  - The HTML tag name.
  - The inner text of an element, if there is any. For example, a heading element will have inner text, but an image element won't.
  - Any attributes associated with the HTML element, including inline styles with the `style` attribute.

- We can access the attributes of a DOM element by accessing a property or by calling a method on it.

New terminology:

- **property accessor** is just a way to access an object property, and the one we know about is called **dot notation**.

## Examples

---

### Getting and Setting Attributes

We can get and set the value of the HTML DOM element attributes via properties. Here's an example with the `id` property.

```
> let h1 = document.querySelector("h1");
> h1;
<h1 id="specialHeader">Best Chocolate Chip Cookies</h1>
> h1.id;
"specialHeader"
> h1.id = "newId";
> h1;
<h1 id="newId">Best Chocolate Chip Cookies</h1>
```

Note that some property names change: the `class` attribute as a HTML DOM element property is called `className`.

### style

To add an inline style:

```
> h1.style.backgroundColor = "hotpink";  
"hotpink"  
> h1;  
<h1 id="specialHeader" style="background-color: hotpink;">Best Chocolate Chip Cookies</h1>
```

To remove an inline style, we have two options: setting the CSS property to "" or to null. However, this does not remove the attribute itself from the DOM element.

```
> h1.style.backgroundColor = null;  
> h1;  
<h1 id="specialHeader" style>Best Chocolate Chip Cookies</h1>
```

## innerText

```
> let h1 = document.querySelector("h1");  
> h1.innerText;  
"Best Chocolate Chip Cookies"  
> h1.innerText = "The Very Best Chocolate Chip Cookies";  
> h1;  
<h1 id="specialHeader">The Very Best Chocolate Chip Cookies</h1>
```

## tagName

```
> let h1 = document.querySelector("h1");
> h1.tagName;
"H1"
> let body = document.querySelector("body");
> body.tagName;
"BODY"
> let ul = document.querySelector("ul");
> ul.tagName;
"UL"
> let firstLi = document.querySelector("ul>li");
> firstLi.tagName.toLowerCase();
"li"
```

## Methods to Remove/Add/Check/Set Attributes

We can also use methods to get the value of, set the value of, check the existence of, and remove attributes:

- `Element.getAttribute()` — gets the value of an attribute of a DOM element by the name of the attribute.
- `Element.setAttribute()` — sets the value of an attribute on a DOM element by the name of the attribute and the specified new value.
- `Element.hasAttribute()` — returns a boolean based on whether or not a DOM element has an attribute by the name specified.
- `Element.removeAttribute()` — removes an attribute from a DOM element by the name of the attribute.

Let's check out some examples using our H1 element from the Cookie Recipe:

```
> let h1 = document.querySelector("h1");
> h1;
<h1 id="specialHeader">Best Chocolate Chip Cookies</h1>
> h1.getAttribute("id");
"specialHeader"
> h1.setAttribute("class", "coolStyles");
> h1;
<h1 id="specialHeader" class="coolStyles">Best Chocolate Ch
ip Cookies</h1>
> h1.hasAttribute("id");
true
> h1.hasAttribute("style");
false
> h1.removeAttribute("class");
> h1;
<h1 id="specialHeader">Best Chocolate Chip Cookies</h1>
```

[Previous \(/introduction-to-programming/javascript-and-web-browsers/javascript-s-global-object\)](#)

[Next \(/introduction-to-programming/javascript-and-web-browsers/understanding-web-apis-interfaces-object-types-and-inheritance\)](#)

Lesson 47 of 75

Last updated more than 3 months ago.

[disable dark mode](#)



Epicodus

(<http://www.epicodus.com>)

© 2023 Epicodus (<http://www.epicodus.com/>), Inc.