How to get CSS styles from an element with JavaScript

Updated on October 15, 2022

▶ **IN THIS ARTICLE** 👇

In JavaScript, sometimes, you might want to retrieve CSS styles applied to an element through inline styles or external style sheets. There are multiple ways to do this, depending on whether you fetch inline styles or rendered styles.

Getting Inline Styles

In vanilla JavaScript, the DOM style property is used to get the styles applied to an HTML element using the `` `style` `` attribute.

Let us say we have the following HTML element:

```
<button id="clickMe" style="color: red; font-weight: bold;">Click Me</button>
```

The following example shows how to get the style information from the above example:

```
const btn = document.getElementById('clickMe')

// pint color & font weight properties
console.log(btn.style.color) // red
console.log(btn.style.fontWeight) // bold
```

However, the `style` property *only* works for the inline styles defined using the `style` attribute. If you try to access a property not defined as an inline style rule, let us say the `backgroundColor`, a `null` value will be returned:

```
console.log(btn.style.backgroundColor) // null
```

The `style` property can not be used to retrieve style information from elsewhere, such as the style rules defined using the `<style>` elements or external style sheets.

Getting Rendered Styles

To get the actual CSS property values used to render an HTML element, you can use the `getComputedStyle()` method. This method works for everything: inline styles, external style sheets, and browser defaults.

Let us say you have the following embedded `<style>` element that defines style rules for the above button:

```
<style>
    button {
        background-color: yellow;
        text-align: center;
        height: 20px;
        width: 100px;
    }
</style>
```

The `getComputedStyle()` method is always called on the `window` object with the element as a parameter and returns an object of properties:

```
const btn = document.getElementById('clickMe')
```

```
// get all rendered styles
const styles = window.getComputedStyle(btn)
```

Now you can access any CSS property like you'd have done with the `style` property. For example, to access the `background-color`, you can do the following:

```
const backgroundColor = styles.backgroundColor // rgb(255, 255, 0)
```

To get the rendered height and width of the element regardless of whether or not it was defined, use the following code:

```
const height = styles.height // 20px
const width = styles.width // 100px
```

Alternatively, you can use the `getPropertyValue()` method on the styles object to access a CSS property. It accepts the *actual* name of the CSS property and not the one used for JavaScript:

```
const fontWeight = styles.getPropertyValue('font-weight') // 700
```

> **Note:** The value `700` is equivalent to `bold` for the CSS property `font-weight`. Similarly, `rgb(255, 255, 0)` is just an RGB value which is the same as `yellow`.

The `getComputedStyle()` method is used to get the actual CSS properties used by the browser to render the element. It works in all modern and old browsers, including IE 9 and higher.

Finally, the `getComputedStyle()` method only works for getting styles. You can not use it to set a specific style to an HTML element. To set CSS properties, you should always use the DOM style property, as explained in an earlier article.

Getting Styles from Pseudo Elements

CSS pseudo-elements are extremely useful to style parts of an element without additional HTML elements.

To get style information from pseudo-elements, you pass in the name of the pseudo-element as a second argument to the `getComputedStyle()` method.

Let us say we have the following `<p>` element:

```html
<p class="tip">Learn JavaScrit for free!</p>
```

Here is the CSS `::first-letter` pseudo element applied to the above paragraph:

```css
.tip::first-letter {
  color: green;
  font-size: 22px;
}
```

To retrieve the color property of the `.tip::first-letter`, you should use the following JavaScript code snippet:

```javascript
const tip = document.querySelector('.tip')

// retrieve all styles
const styles = window.getComputedStyle(tip, '::first-letter')

// get color
console.log(styles.color) // rgb(0, 128, 0)
```

Summary

In this article, we looked at three ways to get CSS styles from an element using JavaScript.

You can use:

1. The DOM style property to retrieve inline styles applied to an element using the `` `style` `` attribute.
2. The `` `window.getComputedStyle()` `` method to retrieve computed styles applied to an element through `` `<style>` `` elements and external style sheets. To get pseudo-elements CSS styles, you need to pass the name of the pseudo-element as a second parameter.

✌ **Like this article? Follow me on Twitter and LinkedIn. You can also subscribe to RSS Feed.**

#JavaScript

You might also like...

- Remove trailing zeros from a number in JavaScript
- Parse float with 2 decimal places in JavaScript
- Format a number to 2 decimal places in JavaScript
- Round a number to 2 decimal places in JavaScript
- How to remove the last character from a string in JavaScript
- How to get the last N characters of a string in JavaScript

Share it ⟶