



# JavaScript `getComputedStyle()` Method

**Summary:** in this tutorial, you will learn how to use the JavaScript `getComputedStyle()` method to get the computed CSS properties of an element.

## Introduction to JavaScript `getComputedStyle()` method

The `getComputedStyle()` is a method of the `window` object, which returns an object that contains the computed style of an element:

```
let style = window.getComputedStyle(element [,pseudoElement]);
```

The `getComputedStyle()` method accepts two arguments:

- `element` is the element that you want to return the computed styles. If you pass another node type e.g., Text node, the method will throw an error.
- `pseudoElement` specifies the pseudo-element to match. It defaults to `null`.

For example, if you want to get the computed value of all the CSS properties of a link with the hover state, you can pass the following arguments to the `getComputedStyle()` method:

```
let link = document.querySelector('a');  
let style = getComputedStyle(link,':hover');  
console.log(style);
```

Note that `window` is the global object, therefore, you can omit it when calling the `getComputedStyle()` method.

The `getComputedStyle()` method returns a live style object which is an instance of the `CSSStyleDeclaration` object. It means that the computed styles are automatically updated when

the styles of the element are changed.

## JavaScript getComputedStyle() examples

Let's take some examples of using the `getComputedStyle()` method.

### 1) Simple getComputedStyle() method example

Consider the following example:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS getComputedStyle() Demo</title>
  <style type="text/css">
    .message {
      background-color: #fff3d4;
      border: solid 1px #f6b73c;
      padding: 20px;
      color: black;
    }
  </style>
</head>
<body>

  <p class="message" style="color:red">
    This is a JS getComputedStyle() Demo!
  </p>

  <script>
    let message = document.querySelector('.message');
    let style = getComputedStyle(message);

    console.log('color:', style.color);
    console.log('background color:', style.backgroundColor);
  </script>
```

```
</body>
</html>
```

Output:

```
color: rgb(255, 0, 0)
background color: rgb(255, 243, 212)
```

Note that we mix all CSS and JavaScript with HTML for the demonstration purpose. In practice, you should separate them into different files.

How it works:

- First, define CSS rules for the `message` class in the head section of the HTML file. The text color is black.
- Second, declare a paragraph element whose text color is red as defined in the inline style. This rule will override the one defined in the head section.
- Third, use the `getComputedStyle()` method to get all the computed styles of the paragraph element. The `color` property is `red` indicated in the Console window ( `rgb(255, 0, 0)` ) as expected.

## 2) The `getComputedStyle()` method for pseudo-elements example

The following example uses the `getComputedStyle()` method to pull style information from a pseudo-element:

```
<html>
<head>
  <title>JavaScript getComputedStyle() Demo</title>
  <style>
    body {
      font: arial, sans-serif;
      font-size: 1em;
      line-height: 1.6;
```

```
    }

    p::first-letter {
        font-size: 1.5em;
        font-weight: normal
    }
</style>
</head>
<body>
    <p id='main'>JavaScript getComputedStyle() Demo for pseudo-elements</p>
    <script>
        let p = document.getElementById('main');
        let style = getComputedStyle(p, '::first-letter');
        console.log(style.fontSize);
    </script>
</body>
</html>
```

Output:

24px

How it works:

- First, define CSS rules for the first letter of any paragraph element in the head section of the HTML file.
- Then, use the `getComputedStyle()` method to pull the computed style of the pseudo-element. The font size of the first letter of the paragraph with the id is 24px.

## Summary

- The `getComputedStyle()` is a method of the `window` object.
- The `getComputedStyle()` method returns an object that contains the computed style of an element.

## Quiz