

# JavaScript Style

**Summary:** in this tutorial, you will learn how to use the style property to manipulate the inline style of the HTML elements.

## Setting inline styles

To set the inline style of an element, you use the `style` property of that element:

```
element.style
```

The `style` property returns the read-only `CSSStyleDeclaration` object that contains a list of CSS properties. For example, to set the color of an element to `red`, you use the following code:

```
element.style.color = 'red';
```

If the CSS property contains hyphens ( - ) for example `-webkit-text-stroke` you can use the array-like notation ( `[]` ) to access the property:

```
element.style['-webkit-text-stroke'] = 'unset';
```

The following table shows the common CSS properties:

CSS	JavaScript
background	background
background-attachment	backgroundAttachment
background-color	backgroundColor

<b>CSS</b>	<b>JavaScript</b>
background-image	backgroundImage
background-position	backgroundPosition
background-repeat	backgroundRepeat
border	border
border-bottom	borderBottom
border-bottom-color	borderBottomColor
border-bottom-style	borderBottomStyle
border-bottom-width	borderBottomWidth
border-color	borderColor
border-left	borderLeft
border-left-color	borderLeftColor
border-left-style	borderLeftStyle
border-left-width	borderLeftWidth
border-right	borderRight
border-right-color	borderRightColor
border-right-style	borderRightStyle
border-right-width	borderRightWidth
border-style	borderStyle
border-top	borderTop

<b>CSS</b>	<b>JavaScript</b>
border-top-color	borderTopColor
border-top-style	borderTopStyle
border-top-width	borderTopWidth
border-width	borderWidth
clear	clear
clip	clip
color	color
cursor	cursor
display	display
filter	filter
float	cssFloat
font	font
font-family	fontFamily
font-size	fontSize
font-variant	fontVariant
font-weight	fontWeight
height	height
left	left
letter-spacing	letterSpacing

<b>CSS</b>	<b>JavaScript</b>
line-height	lineHeight
list-style	listStyle
list-style-image	listStyleImage
list-style-position	listStylePosition
list-style-type	listStyleType
margin	margin
margin-bottom	marginBottom
margin-left	marginLeft
margin-right	marginRight
margin-top	marginTop
overflow	overflow
padding	padding
padding-bottom	paddingBottom
padding-left	paddingLeft
padding-right	paddingRight
padding-top	paddingTop
page-break-after	pageBreakAfter
page-break-before	pageBreakBefore
position	position

CSS	JavaScript
stroke-dasharray	strokeDasharray
stroke-dashoffset	strokeDashoffset
stroke-width	strokeWidth
text-align	textAlign
text-decoration	textDecoration
text-indent	textIndent
text-transform	textTransform
top	top
vertical-align	verticalAlign
visibility	visibility
width	width
z-index	zIndex

To completely override the existing inline style, you set the `cssText` property of the `style` object. For example:

```
element.style.cssText = 'color:red;background-color:yellow';
```

Or you can use the `setAttribute()` method:

```
element.setAttribute('style','color:red;background-color:yellow');
```

Once setting the inline style, you can modify one or more CSS properties:

```
element.style.color = 'blue';
```

If you do not want to overwrite the existing CSS properties completely, you can concatenate the new CSS property to the `cssText` as follows:

```
element.style.cssText += 'color:red;background-color:yellow';
```

In this case, the `+=` operator appends the new style string to the existing one.

The following `css()` helper function is used to set multiple styles for an element from an object of key-value pairs:

```
function css(e, styles) {  
  for (const property in styles)  
    e.style[property] = styles[property];  
}
```

You can use this `css()` function to set multiple styles for an element with the id `#content` as follows:

```
let content = document.querySelector('#content');  
css(content, { background: 'yellow', border: 'solid 1px red' });
```

The following example uses the `style` object to set the CSS properties of a paragraph with the id `content` :

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>JS Style Demo</title>  
  </head>  
  <body>  
    <p id="content">JavaScript Setting Style Demo!</p>  
    <script>
```

```
    let p = document.querySelector('#content');  
    p.style.color = 'red';  
    p.style.fontWeight = 'bold';  
  </script>  
</body>  
</html>
```

How it works:

- First, select the paragraph element whose id is `content` by using the `querySelector()` method.
- Then, set the color and font-weight properties of the paragraph by setting the `color` and `fontWeight` properties of the `style` object.

## Getting inline styles

The `style` property returns the inline styles of an element. It is not very useful in practice because the style property doesn't return the rules from elsewhere e.g., styles from an external style sheet.

To get all styles applied to an element, you should use the `window.getComputedStyle()` method.

## Summary

- Use the properties of `element.style` object to set the inline CSS properties for the HTML element.

## Quiz