



CSSStyleSheet.insertRule()

English ▼

The `CSSStyleSheet.insertRule()` method inserts a new CSS rule into the current style sheet, with some restrictions.

Note: Although `insertRule()` is exclusively a method of `CSSStyleSheet`, it actually inserts the rule into `CSSStyleSheet.cssRules` — its internal `CSSRuleList`.

Syntax

```
stylesheet.insertRule(rule [, index])
```

Parameters

rule

A `DOMString` containing the rule to be inserted. What the inserted rule must contain depends on its type:

- **For rule-sets**, both a selector and a style declaration.
- **For at-rules**, both an at-identifier and the rule content.

index | Optional

A positive integer less than or equal to `stylesheet.cssRules.length`, representing the newly inserted rule's position in `CSSStyleSheet.cssRules`. The

default is `0`. (In older implementations, this was required. See [Browser compatibility](#) for details.)

Return value

The newly inserted rule's index within the stylesheet's rule-list.

Restrictions

CSS has some intuitive and not-so-intuitive restrictions affecting where rules can be inserted. Violating them will likely raise a `DOMException`.

- If `index > CSSRuleList.length`, the method aborts with an `IndexSizeError`.
- If `rule` cannot be inserted at `index 0` due to some CSS constraint, the method aborts with a `HierarchyRequestError`.
- If more than one rule is given in the `rule` parameter, the method aborts with a `SyntaxError`.
- If trying to insert an `@import` at-rule after a style rule, the method aborts with a `HierarchyRequestError`.
- If `rule` is `@namespace` and the rule-list has more than just `@import` at-rules and/or `@namespace` at-rules, the method aborts with an `InvalidStateError`.

Examples

Inserting a new rule

This snippet pushes a new rule onto the top of my stylesheet.

```
1 | myStyle.insertRule('#blanc { color: white }', 0);
```

Function to add a stylesheet rule

```

/**
 * Add a stylesheet rule to the document (it may be better practice
 * to dynamically change classes, so style information can be kept in
 * genuine stylesheets and avoid adding extra elements to the DOM).
 * Note that an array is needed for declarations and rules since ECMAScr
 * not guarantee a predictable object iteration order, and since CSS is
 * order-dependent.
 * @param {Array} rules Accepts an array of JSON-encoded declarations
 * @example
addStylesheetRules([
  ['h2', // Also accepts a second argument as an array of arrays instead
    ['color', 'red'],
    ['background-color', 'green', true] // 'true' for !important rules
  ],
  ['.myClass',
    ['background-color', 'yellow']
  ]
]);
*/
function addStylesheetRules (rules) {
  var styleEl = document.createElement('style');

  // Append <style> element to <head>
  document.head.appendChild(styleEl);

  // Grab style element's sheet
  var styleSheet = styleEl.sheet;

  for (var i = 0; i < rules.length; i++) {
    var j = 1,
        rule = rules[i],
        selector = rule[0],
        propStr = '';
    // If the second argument of a rule is an array of arrays, correct o
    if (Array.isArray(rule[1][0])) {
      rule = rule[1];
      j = 0;
    }

    for (var pl = rule.length; j < pl; j++) {
      var prop = rule[j];
      propStr += prop[0] + ':' + prop[1] + (prop[2] ? ' !important' : '

```

```

    // Insert CSS Rule
    styleSheet.insertRule(selector + '{' + propStr + '}', styleSheet.css
  }
}

```

Polyfill

The below polyfill will correct the input of the arguments of `insertRule()` to standardize them in Internet Explorer 5–8. It supplements `insertRule()` with a function that separates the selector from the rules before sending the arguments to the default native `insertRule()`.

```

1  (function(Sheet_proto){
2      var originalInsertRule = Sheet_proto.insertRule;
3
4      if (originalInsertRule.length === 2){ // 2 mandatory arguments:
5          Sheet_proto.insertRule = function(selectorAndRule){
6              // First, separate the selector from the rule
7              a: for (var i=0, Len=selectorAndRule.length, isEscaped=0, ne
8                  newCharCode = selectorAndRule.charCodeAt(i);
9                  if (!isEscaped && (newCharCode === 123)) { // 123 = "{" .ch
10                     // Secondly, find the last closing bracket
11                     var openBracketPos = i, closeBracketPos = -1;
12
13                     for (; i !== Len; ++i) {
14                         newCharCode = selectorAndRule.charCodeAt(i);
15                         if (!isEscaped && (newCharCode === 125)) { // 125 = "}"
16                             closeBracketPos = i;
17                         }
18                         isEscaped ^= newCharCode===92?1:isEscaped; // 92 = "\\"
19                     }
20
21                     if (closeBracketPos === -1) break a; // No closing brack
22                     /*else*/ return originalInsertRule.call(
23                         this, // the sheet to be changed
24                         selectorAndRule.substring(0, openBracketPos), // The s
25                         selectorAndRule.substring(closeBracketPos), // The rul


```

```

26         arguments[3] // The insert index
27     );
28 }
29
30     // Works by if the char code is a backslash, then isEscape
31     // gets flipped (XOR-ed by 1), and if it is not a backslas
32     // then isEscaped gets XORed by itself, zeroing it
33     isEscaped ^= newCharCode===92?1:isEscaped; // 92 = "\\".ch
34 }
35 // Else, there is no unescaped bracket
36 return originalInsertRule.call(this, selectorAndRule, "", ar
37 };
38 }
39 })(CSSStyleSheet.prototype);

```

Specifications

| Specification | Status | Comment |
|---|---|---|
| CSS Object Model (CSSOM) The definition of 'CSSStyleSheet.insertRule' in that specification. |  WD Working Draft | No change from Document Object Model (DOM) Level 2 Style Specification. |
| Document Object Model (DOM) Level 2 Style Specification The definition of 'CSSStyleSheet.insertRule' in that specification. | Obsolete | Initial definition |

Browser compatibility

[Update compatibility data on GitHub](#)

insertRule()

Yes

| | |
|--------------------------|-----|
| Edge | 12 |
| Firefox | 1 |
| IE | 9 |
| Opera | Yes |
| Safari | Yes |
| WebView Android | Yes |
| Chrome Android | Yes |
| Firefox Android | 4 |
| Opera Android | Yes |
| Safari iOS | Yes |
| Samsung Internet Android | Yes |

`index` is optional

| | |
|--------------------------|-----|
| Chrome | 60 |
| Edge | ≤79 |
| Firefox | 55 |
| IE | No |
| Opera | 47 |
| Safari | ? |
| WebView Android | 60 |
| Chrome Android | 60 |
| Firefox Android | ? |
| Opera Android | 44 |
| Safari iOS | ? |
| Samsung Internet Android | 8.0 |

What are we missing?



Full support

No support



Compatibility unknown

Legacy browser support

To support Internet Explorer 8 and below, use: `addRule(selector, rule [, index]);`. **Example:** `addRule('pre', 'font: 14px verdana');` // add rule at end

Also note the non-standard `removeRule()` and `.rules` instead of `deleteRule()` and `.cssRules`, respectively.

See also

- `CSSStyleSheet.deleteRule`
- Cross-Browser CSS-rules ordering (CSS1)
- Quirksmode - CSS

Last modified: Mar 23, 2019, by MDN contributors

Syntax

Examples

Polyfill

Specifications

Browser compatibility

See also

Related Topics

CSS Object Model

cssStyleSheet

CSSStyleSheet

▼ Properties

`cssRules`
`ownerRule`
`rules`

▼ Methods

`addRule()`
`deleteRule()`
`insertRule()`
`removeRule()`

▼ Inheritance:

`StyleSheet`

▼ Related pages for CSSOM

`AnimationEvent`
`CSS`
`CSSConditionRule`
`CSSGroupingRule`
`CSSKeyframeRule`
`CSSKeyframesRule`
`CSSMediaRule`
`CSSNamespaceRule`
`CSSPageRule`
`CSSRule`
`CSSRuleList`
`CSSStyleDeclaration`
`CSSStyleRule`
`CSSSupportsRule`
`CssTextDeclaration`

CaretPosition

LinkStyle

MediaQueryList

MediaQueryListListener

Screen

StyleSheet

StyleSheetList

TransitionEvent



Learn the best of web development

Get the latest and greatest from MDN delivered straight to your inbox.

Sign up now