

Window.getComputedStyle()



[查看所有贡献者](#)

摘要

`Window.getComputedStyle()` 方法在应用活动样式表并解析这些值可能包含的任何基本计算后，给出一个元素的所有CSS属性的值。

语法

```
var style = window.getComputedStyle(element, [pseudoElt]);
```

element

用于获取计算样式的 [Element](#)

pseudoElt

可选

指定一个要匹配的伪元素的字符串。必须对普通元素省略（或null）。

📌 注意：在Gecko2.0 (Firefox 4 / Thunderbird 3.3 / SeaMonkey 2.1)之前版本，参数pseudoElt是必要的。如果为null，则不指定其他主要浏览器必须指定此参数。Gecko已经更改为匹配其他浏览器的行为。。

返回的样式是一个 *活的* [CSSStyleDeclaration](#) 对象，当元素的样式更改时,它会自动更新本身。

示例

```
1 var elem1 = document.getElementById("elemId");
2 var style = window.getComputedStyle(elem1, null);
3
4 // 它等价于
5 // var style = document.defaultView.getComputedStyle(elem1, null);
```

```
1 <style>
2   #elem-container{
3     position: absolute;
4     left:    100px;
5     top:     200px;
6     height:  100px;
7   }
8 </style>
9
10 <div id="elem-container">dummy</div>
11 <div id="output"></div>
12
13 <script>
14   function getTheStyle(){
15     var elem = document.getElementById("elem-container");
16     var theCSSprop = window.getComputedStyle(elem,null).getPropertyValue("height");
17     document.getElementById("output").innerHTML = theCSSprop;
18   }
19   getTheStyle();
20 </script>
```

```
1 function dumpComputedStyles(elem,prop) {
2
3   var cs = window.getComputedStyle(elem,null);
```

```

4   if (prop) {
5       dump("    "+prop+" : "+cs.getPropertyValue(prop)+"\n");
6       return;
7   }
8   var len = cs.length;
9   for (var i=0;i<len;i++) {
10
11       var style = cs[i];
12       dump("    "+style+" : "+cs.getPropertyValue(style)+"\n");
13   }
14
15 }

```

描述

返回的对象与从元素的 `style` 属性返回的对象具有相同的类型;然而，两个对象具有不同的目的。从`getComputedStyle`返回的对象是只读的，可以用于检查元素的样式（包括由一个`<style>`元素或一个外部样式表设置的那些样式）。`elt.style`对象应用于在特定元素上设置样式。

第一个参数必须是Element对象(传递一个非节点元素，如一个`#text`节点，将会抛出一个错误)。从Gecko 1.9.2（Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0）开始，现在返回的一个在URL周围有引号的URL值，像这样：`url("http://foo.com/bar.jpg")`。

默认视图（defaultView）

在许多在线的演示代码中，`getComputedStyle`是通过`document.defaultView`对象来调用的。大部分情况下，这是不需要的，因为可以直接通过`window`对象调用。但有一种情况，你必须需要使用`defaultView`，那是在firefox3.6上访问子框架内的样式(`iframe`)

与伪元素一起使用

`getComputedStyle` 可以从**伪元素**拉取样式信息 (比如, `::after`, `::before`, `::marker`, `::line-marker`—查看 [详情](#) 这里).

```

1  <style>
   h3::after {

```

```
2         content: "rocks!";
3     }
4 </style>
5
6 <h3>generated content</h3>
7
8 <script>
9     let h3 = document.querySelector('h3'),
10     result = getComputedStyle(h3, '::after').content;
11     alert(`the generated content is: ${result}`);
12     console.log(`the generated content is: ${result}`);
13     // the generated content is: "rocks!"
14 </script>
15
```

备注

getComputedStyle 的返回值是 **resolved values**, 通常跟CSS2.1中的**computed values**是相同的值。但对于一些旧的属性, 比如width, height, padding 它们的值又为 **used values**。最初, CSS2.0定义的计算值Computed values 就是属性的最终值。但是CSS2.1 重新定义了 computed values 为布局前的值, used values布局后的值。布局前与布局后的区别是, width 或者 height的 百分比可以代表元素的宽度, 在布局后, 会被像素值替换。

在某些情况下, 通过浏览器会特意返回不准确的值。特别是在避免CSS 浏览历史泄露的安全问题, 比如, 浏览者看过某个网站, 它的链接通常会变成蓝色带下划线的链接, 通过判断链接的颜色 (getComputedStyle(node, null).color) 是否为蓝色, 就会泄露用户的浏览历史, 所以浏览器会特意返回不准确的值, 保护用户隐私。可以了解更多关于css安全的链接<http://blog.mozilla.com/security/2010/03/31/plugging-the-css-history-leak/> 和 <http://hacks.mozilla.org/2010/03/privacy-related-changes-coming-to-css-vistited/>

在CSS3 的动画中, Firefox 的getComputedStyle会返回原始的属性值, 但在Webkit的浏览器中会返回最终属性值。

在Firefox中, 属性值为auto的会直接返回 used value, 而不是 'auto'. 比如, 你在设定了一个元素的css为height:30px; top: auto; bottom:0; 它的父元素的height:100px; 那么top的Computed value 为 'auto', used value 为 '70px' = 100px - 30px;

浏览器兼容

Desktop	Mobile				
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	(Yes)	(Yes)	9	(Yes)	(Yes)
pseudo-element support	(Yes)	(Yes)	11	未实现	(Yes)

详述

- [DOM Level 2 Style: getComputedStyle](#)
- [CSS Object Model specification](#)

还可以看看

- `window.getDefaultComputedStyle`
- `resolved value`
- <http://stackoverflow.com/questions/15195209/how-to-get-font-size-in-html>
- `getPropertyValue` ? http://www.w3schools.com/jsref/jsref_getcomputedstyle.asp
- <https://www.w3.org/TR/css-content-3/>
- <https://www.w3.org/TR/cssom-1/>
- https://developer.mozilla.org/en-US/docs/Web/CSS/used_value
- <https://developer.mozilla.org/en-US/docs/Web/API/Window/getDefaultComputedStyle>
- <https://developer.mozilla.org/en-US/docs/Web/API/CSSStyleDeclaration>