

# 获取元素CSS值之getComputedStyle方法熟悉

这篇文章发布于 2012年05月8日, 星期二, 23:31, 归类于 [JS API](#)。阅读 272947 次, 今日 33 次 [83 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>

本文地址: <http://www.zhangxinxu.com/wordpress/?p=2378>

## 一、碎碎念~前言

我们都用过jQuery的 `CSS()` 方法, 其底层运作就应用了 `getComputedStyle` 以及 `getPropertyValue` 方法。

对于那些只想混口饭吃的人来讲, 晓得 `CSS()` 如何使用就足够了。对于希望在JS道路上越走越远的来说, 简单了解一些JS库底层实现对自己的学习很有帮助。可能谈不上信手拈来的使用, 至少对创造一些创意新颖的新技术拓宽了思路。

jQuery为何受欢迎, 其中原因之一就是方法名称比较短。好比打架一样, 块头大的潜意识认为厉害, 就不由得心生畏惧, 退避三舍; 小个子(或村妇, 小孩)嘛, 自然以为是软豆腐, 愿意接受作为对手。



因此, 当看到类似 `getComputedStyle` 这么长的方法名的时候, 我们身体里那个懒惰小人突然就醒了: 哎哟哟, 这东西, 就跟放心食品一样, 几年才见一回。看上去又蛮复杂, 想到晚上还要跟妹子聊天。这东西, 就让它从旁边过吧, 反正不会掉块肉的。

网上不是有这么个段子嘛: 可怕的是, 比你聪明的人比你更勤奋。那么, 这里的“勤奋”的差异体现在什么地方呢? 就拿这个 `getComputedStyle` 举例: 懒惰青年是看一下, 立马像看见凤姐一样视线离开; 普通青年是看一下, 发现自己不了解, 百一下或谷一下, 熟悉了解之; 勤奋青年是不仅了解, 还抽出时间实践之(如做个简单demo测试), 熟悉方法的特异性, 验证一些观点正确性。

您可以按照你现在的心理状态看看你是哪类青年: 如果此时, 您已经对本文的内容没有兴趣了(当然, 您的工作与JS关系亲密), 您是……(你懂的); 如果您看完本文内容, 发现, 还是有不少收获, 心中回味下, 然后ctrl+w去其他地方觅食, 不再回来, 那您是普通青年; 如果您看完本文, 然后对所讲的一些内容作了额外的测试, 例如IE9浏览器是否支持测试等, 并通过评论形式指出可能的错误, 不得不说您是勤奋青年, 假以时日, 必有一番技术作为。从我文章的些评论来看, 有不少同行就是这样的勤奋人儿。

的, 好久没写文章, 话又多了, 打住, 进入正题。

## 二、getComputedStyle是?

`getComputedStyle` 是一个可以获取当前元素所有最终使用的CSS属性值。返回的是一个CSS样式声明对象([object CSSStyleDeclaration])，只读。

`getComputedStyle()` gives the final used values of all the CSS properties of an element.

语法如下:

```
var style = window.getComputedStyle("元素", "伪类");
```

例如:

```
var dom = document.getElementById("test"),
    style = window.getComputedStyle(dom, ":after");
```

就两个参数，大家都懂中文的，没什么好说的。只是额外提示下: Gecko 2.0 (Firefox 4 / Thunderbird 3.3 / SeaMonkey 2.1) 之前，第二个参数“伪类”是必需的（如果不是伪类，设置为 `null`），不过现在嘛，不是必需参数了。

## 三、getComputedStyle与style的区别

我们使用 `element.style` 也可以获取元素的CSS样式声明对象，但是其与 `getComputedStyle` 方法还有一些差异的。

### 1. 只读与可写

正如上面提到的 `getComputedStyle` 方法是只读的，只能获取样式，不能设置；而 `element.style` 能读能写，能屈能伸。

### 2. 获取的对象范围

`getComputedStyle` 方法获取的是最终应用在元素上的所有CSS属性对象（即使没有CSS代码，也会把默认的祖宗八代都显示出来）；而 `element.style` 只能获取元素 `style` 属性中的CSS样式。因此对于一个光秃秃的元素 `<p>`，`getComputedStyle` 方法返回对象中 `length` 属性值（如果有）就是 `190+`（据我测试FF:192, IE9:195, Chrome:253, 不同环境结果可能有差异），而 `element.style` 就是 `0`。

## 四、getComputedStyle与defaultView

如果我们查看jQuery源代码，会发现，其 `css()` 方法实现不是使用的 `window.getComputedStyle` 而是 `document.defaultView.getComputedStyle`，啥？这是怎么回事？

```
cssHeight = [ "Top", "Bottom" ],

// cache check for defaultView.getComputedStyle
getComputedStyle = document.defaultView && document.defaultView.getComputedStyle,
// normalize float css property
styleFloat = jQuery.support.cssFloat ? "cssFloat" : "styleFloat",
fcamelCase = function( all, letter ) {
    return letter + allInnerCase();
}
```

张鑫旭-鑫空间-鑫生活  
http://www.zhangxinxu.com

实际上，使用 `defaultView` 基本上是没有必要的，`getComputedStyle` 本身就存在 `window` 对象之中。根据DennisHall的说法，使用 `defaultView` 可能一是人们不太乐意在window上专门写个东西，二是让API在Java中也可用（这我不懂，忘指点~~）。

不过有个特殊情况，在FireFox3.6上不使用 `defaultView` 方法就搞不定的，就是访问框架(frame)的样式

.

## 五、getComputedStyle兼容性

对于桌面设备：

Chrome
Firefox (Gecko)
Internet Explorer
Opera
Safari
基本支持
✓
✓
9
✓
✓
伪类元素支持
✓
✓
✗
✗
✓

对于手机设备：

Android
Firefox Mobile (Gecko)
IE Mobile
Opera Mobile
Safari Mobile
基本支持
✓
✓
WP7 Mango
✓
✓
伪元素支持
?
?
✗
?
?

上面打问号的表示没有测试，是否兼容不知。如果您方便测试，欢迎将测试结果告知，这里将及时更新，并附上您的姓名，以谢您做的贡献。

我们先把注意力放在桌面设备上，可以看到，`getComputedStyle` 方法IE6~8是不支持的，得，背了半

天的媳妇，发现是孙悟空变的——郁闷了。不急，IE自有自己的一套东西。

## 六、getComputedStyle与currentStyle

`currentStyle` 是IE浏览器自娱自乐的一个属性，其与 `element.style` 可以说是近亲，至少在使用形式上类似，`element.currentStyle`，差别在于 `element.currentStyle` 返回的是元素当前应用的最终CSS属性值（包括外链CSS文件，页面中嵌入的 `<style>` 属性等）。

因此，从作用上将，`getComputedStyle` 方法与 `currentStyle` 属性走的很近，形式上则 `style` 与 `currentStyle` 走的近。不过，`currentStyle` 属性貌似不支持伪类样式获取，这是与 `getComputedStyle` 方法的差异，也是jQuery `css()` 方法无法体现的一点。

//zxx: 如果你只知jQuery `css()`方法，你是不会知道伪类样式也是可以获取的，虽然部分浏览器不支持。

例如，我们要获取一个元素的高度，可以类似下面的代码：

```
alert((element.currentStyle? element.currentStyle : window.getComputedStyle(element, null)).height);
```

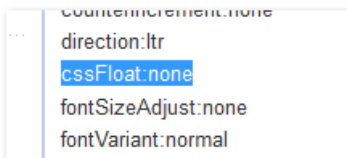
您可以狠狠地点击这里：[使用getComputedStyle和currentStyle 获取元素高度demo](#)

结果FireFox下显示 `24px` (经过计算了), 而IE浏览器下则是CSS中的 `2em` 属性值：

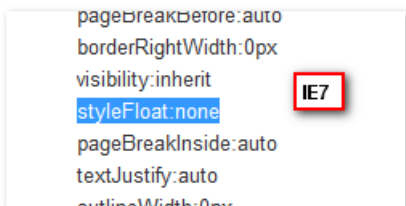


`getComputedStyle` 方法与 `currentStyle` 属性其他具体差异还有很多，我以一个普通按钮做元素，遍历了其中靠谱的属性名和属性值，您可以狠狠地点击这里：[getComputedStyle和currentStyle属性展示demo](#)

仔细对比查看，我们可以看到不少差异，例如浮动属性，FireFox浏览器下是这个(`cssFloat`):



IE7浏览器下则是 `styleFloat`：



而IE9浏览器下则是 `cssFloat` 和 `styleFloat` 都有。

等其他N多差异。

## 七、getPropertyValue方法

`getPropertyValue` 方法可以获取CSS样式申明对象上的属性值（直接属性名称），例如：

```
window.getComputedStyle(element, null).getPropertyValue("float");
```

如果我们不使用 `getPropertyValue` 方法，直接使用键值访问，其实也是可以的。但是，比如这里的 `float`，如果使用键值访问，则不能直接使用 `getComputedStyle(element, null).float`，而应该是 `cssFloat` 与 `styleFloat`，自然需要浏览器判断了，比较折腾！

使用 `getPropertyValue` 方法不必可以驼峰书写形式（不支持驼峰写法），例如：`style.getPropertyValue("border-top-left-radius")`；

更新于2018-05-30

`getPropertyValue` 方法可以用来获取CSS原生变量值，如下示意：

```
const sidebarElement = document.querySelector('.sidebar');

// cssStyles包含sidebarElement元素的样式
const cssStyles = getComputedStyle(sidebarElement);

// 获取 --left-pos CSS 变量值
const cssVal = String(cssStyles.getPropertyValue('--left-pos')).trim();

// 输出cssVal
// 输出变量值是: 100px
console.log(cssVal);
```

设置CSS变量可以使用 `setProperty()` 方法，例如：

```
sidebarElement.style.setProperty('--left-pos', '200px');
```

兼容性

`getPropertyValue` 方法IE9+以及其他现代浏览器都支持，见下表：

Chrome
Firefox (Gecko)
Internet Explorer
Opera
Safari
基本支持


9



OK，一涉及到兼容性问题(IE6-8肿么办)，感觉头开始微微作痛了～，不急，IE自由一套自己的套路，就是 `getAttribute` 方法。

## 八、getPropertyValue和getAttribute

在老的IE浏览器（包括最新的），`getAttribute` 方法提供了与 `getPropertyValue` 方法类似的功能，可以访问CSS样式对象的属性。用法与 `getPropertyValue` 类似：

```
style.getAttribute("float");
```

注意到没，使用 `getAttribute` 方法也不需要 `cssFloat` 与 `styleFloat` 的怪异写法与兼容性处理。不过，还是有一点差异的，就是属性名需要驼峰写法，如下：

```
style.getAttribute("backgroundColor");
```

如果不考虑IE6浏览器，貌似也是可以这么写：

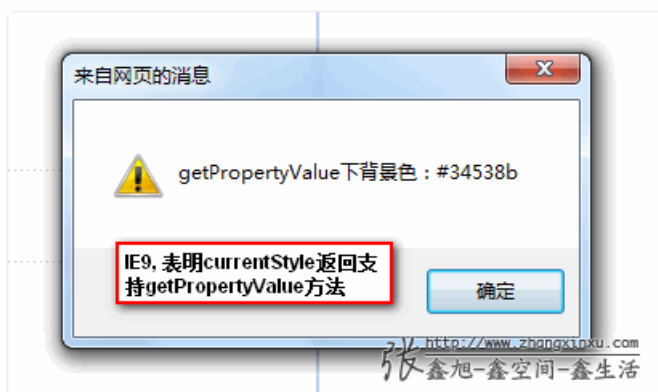
```
style.getAttribute("background-color");
```

实例才是王道，您可以狠狠地点击这里：[getPropertyValue和getAttribute获取背景色demo](#)

结果FireFox下一如既往的rgb颜色返回(Chrome也是返回rgb颜色)：



对于IE9浏览器，虽然应用的是 `currentStyle`，但是从结果上来讲，`currentStyle` 返回的对象是完全支持 `getPropertyValue` 方法的。



## 九、getPropertyValue和getPropertyCSSValue

从长相上看 `getPropertyCSSValue` 与 `getPropertyValue` 是近亲，但实际上，`getPropertyCSSValue` 要顽劣的多。

`getPropertyCSSValue` 方法返回一个CSS最初值(CSSPrimitiveValue)对象(width, height, left, ...)或CSS值列表(CSSValueList)对象(background-color, font-size, ...)，这取决于 `style` 属性值的类型。在某些特别的style属性下，其返回的是自定义对象。该自定义对象继承于CSSValue对象（就是上面所说的 `getComputedStyle` 以及 `currentStyle` 返回对象）。

`getPropertyCSSValue` 方法兼容性不好，IE9浏览器不支持，Opera浏览器也不支持（实际支持，只是老是抛出异常）。而且，虽然Firefox中，`style` 对象支持 `getPropertyCSSValue` 方法，但总是返回 `null`。因此，目前来讲，`getPropertyCSSValue` 方法可以先不闻不问。

## 十、补充~结语

有了jQuery等优秀库，我们有熟悉底层的 `getComputedStyle` 方法的必要吗？

实际上，本文一直没有深入展开 `getComputedStyle` 方法一个很重要的，类似 `css()` 方法没有的功能——获取伪类元素样式。但从这一点上将，熟悉 `getComputedStyle` 方法有必要。

下一篇文章，我就将介绍如何实现 `getComputedStyle` 方法在伪类元素上的特异功能，实现CSS3 media queries下一些JS交互，及实际应用。

well, 开篇已经啰哩吧嗦不少内容了，这里就不再讲废话了。行文匆忙，文中难免有表述不准确之处，欢迎指正。欢迎补充，感谢阅读，希望本文的内容能够对您的学习有所帮助。

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。



« 近期折腾：使用JS以及HTML5实现鑫情动画

伪类+js实现CSS3 media queries跨界准确判断 »

### 猜你喜欢

- 小tips: 如何HTML标签和JS中设置CSS3 var变量
- style标签下的CSS代码的显示与实时编辑
- 伪类+js实现CSS3 media queries跨界准确判断
- 内容loading加载后高度变化CSS3 transition体验优化
- Canvas中颜色过渡动画效果的实现
- CSS :visited伪类选择器隐秘往事回忆录
- JS护航下CSS3效果的渐进增强使用
- ieBetter.js-让IE6-IE8拥有IE9+,Chrome等浏览器特性
- IE9对CSS3的支持情况概述
- CSS3选择器:nth-child和:nth-of-type之间的差异
- 我熟知的三种三栏网页宽度自适应布局方法

分享到：        0

标签： CSS伪类, currentStyle, getAttribute, getComputedStyle, getPropertyValue, ie9, style, 兼容性

### 发表评论（目前83条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

## « 先前评论

### 1. 海天鹰说道：

2018年11月8日 00:18

a:visited 伪类怎么获取？

[回复](#)



#### 张鑫旭说道：

2018年11月9日 20:20

安全限制，无法获取。可参见“CSS :visited伪类选择器隐秘往事回忆录”这篇文章：<https://www.zhangxinxu.com/wordpress/2018/10/css-visited-pseudo-class/>

[回复](#)



### 2. 我是谁说道：

2018年08月27日 20:17

说唠叨的，说废话的，你们可以按照自己的语言自己整理啊，别人辛辛苦苦免费做知识，被批的一文不值，真是闲的蛋疼。

[回复](#)



### 3. 小狼子说道：

2018年07月24日 14:48

恕我直言，博主，有些东西直接给个表列出来，再说明一下，比叨叨一大堆好很多，讲话直，勿怪

[回复](#)



### 4. 笑军生说道：

2018年03月23日 10:35

旭哥文章真幽默又有料，看着看着就学到了，谢谢

[回复](#)



### 5. 张旭说道：

2018年01月28日 21:16

getAttribute不是元素节点的方法吗，刚才自己试了一下，发现

window.getComputedStyle(hh).getAttribute('id')

报错，根本没有这个方法啊

[回复](#)





none说道:

2018年01月31日 19:08

1. `getAttribute()` 是DOM方法
2. `Element.getAttribute()` 是这样调用的
3. 返回指定属性名的属性值

```
var oDiv = document.querySelector(".ele").getAttribute('style');
```

[回复](#)



6. monkeyfly说道:

2018年01月26日 22:02

```
console.log(getComputedStyle(iEl).display);
```

```
if(getComputedStyle(iEl).display == "none"){  
  iEl.style.display = "inline-block";  
}else{  
  iEl.style.display = "none";  
}
```

断点调试发现一个奇怪的事情，第一次点击显示i元素之后，用`getComputedStyle()`去获取i元素的`display`值为“none”；但是第二次继续点击后，再用`getComputedStyle()`去获取i元素的`display`得到的是“block”，而不是“inline-block”。但是我设置的就是“inline-block”，为什么会得到“block”呢？求大神解答。

[回复](#)

张旭说道:

2018年01月28日 21:25

单独测试这段代码没问题，只测试了chrome和IE，都是inline-block啊

[回复](#)



7. guowei001说道:

2017年08月16日 11:08

有没有手机版的博主，方便在移动学习

[回复](#)

cca1000说道:

2017年11月22日 13:07

同样的地址 手机打开就可以了。

[回复](#)



8. guowei001说道:

2017年08月16日 11:07

学习了，真心不错

[回复](#)



9. bydraw说道:

2017年03月26日 11:58

有个疑问

如果某个css属性比如width默认没有设置

`getComputedStyle`返回出来的会是窗口宽度

(chrome下)

[回复](#)



fXy-during 说道:

2017年07月20日 21:51

你是块级元素才会出现这种问题吧

[回复](#)



10. Ittplus 说道:

2017年01月24日 15:43

我好像是普通青年！捂脸！把demo做了一遍，发现chrome和ff下2em显示高度为26.6563px。小白去查资料了！向大大比哈特！

[回复](#)



hello 说道:

2017年03月24日 13:50

请教一下，什么原因导致出现浮点数？如何处理？万分感谢！

[回复](#)



夏说道:

2017年04月3日 21:55

我之前看到一个人特意说计算机的浮点数，并且测试了一下，发现计算机对于小数的判断并不是很准，所以要做四舍五入处理

[回复](#)



11. 山高水远 说道:

2016年11月8日 17:48

给你赞一个，项目中遇到问题都会来看一下解决方法，getComputedStyle的功能很强大，虽然有现成的库如jQuery可以使用，当不想用这个库的时候怎么用原生的方法解决是个问题，今天碰到了，获取元素宽度，公司的大神说了这个 getBoundingClientRect 这个方法，我自己印象中还有getComputedStyle，都可以解决我的问题

[回复](#)



## « 先前评论

### 最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

### 今日热门

- » [常见的CSS图形绘制合集](#) <sup>(193)</sup>
- » [未来必热: SVG Sprite技术介绍](#) <sup>(120)</sup>
- » [粉丝群第1期CSS小测点评与答疑](#) <sup>(115)</sup>
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) <sup>(93)</sup>
- » [让所有浏览器支持HTML5 video视频标签](#) <sup>(86)</sup>
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) <sup>(82)</sup>
- » [CSS3下的147个颜色名称及对应颜色值](#) <sup>(80)</sup>
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) <sup>(76)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(76)</sup>
- » [小tips: 纯CSS实现打字动画效果](#) <sup>(76)</sup>

## 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看，for..in和for..of在那里吵架！](#) <sup>(60)</sup>
- » [是时候好好安利下LuLu UI框架了！](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

## 猜你喜欢

- [小tips: 如何HTML标签和JS中设置CSS3 var变量](#)
- [style标签下的CSS代码的显示与实时编辑](#)
- [伪类+js实现CSS3 media queries跨界准确判断](#)
- [内容loading加载后高度变化CSS3 transition体验优化](#)
- [Canvas中颜色过渡动画效果的实现](#)
- [CSS :visited伪类选择器隐秘往事回忆录](#)
- [JS护航下CSS3效果的渐进增强使用](#)
- [ieBetter.js-让IE6-IE8拥有IE9+,Chrome等浏览器特性](#)
- [IE9对CSS3的支持情况概述](#)
- [CSS3选择器:nth-child和:nth-of-type之间的差异](#)
- [我熟知的三种三栏网页宽度自适应布局方法](#)