« RegExp Tool | Main | Love WebKit, Hate Safari »

erik's weblog







Computed vs Cascaded Style

Most JS frameworks and libraries I've seen have a function similar to this:

```
function getStyle(el, prop) {
  if (document.defaultView && document.defaultView.getComputedStyle) {
    return document.defaultView.getComputedStyle(el, null)[prop];
} else if (el.currentStyle) {
    return el.currentStyle[prop];
} else {
    return el.style[prop];
}
```

So what is wrong with this you might ask? Lets add some background and then we can get back to that question.

Computed Style

Computed style represents the actual computed value of a style in absolute units. For example '100px' (for width) or 'left' (for text-align). The computed value is never 'auto', 'inherit', '50%', 'smaller', '1.5em' etc. These values are relative to something and the browser needs to convert these values to absolute values first.

Cascaded Style

Cascaded style is the real style value that is applied to an element when the style value is set to 'inherit'. Inherit means that it should get its value from the first ancestor that has a non 'inherit' value. A few CSS properties have their value set to 'inherit' by default. Most common of these are probably 'color' and 'font-size'. If you change the font size of a paragraph you want the links in the paragraph to change their font size as well.

Inline Style

Inline style is the style value you have added to the style attribute on you element or the style properties you have set in script using element.style.propertyName.

Example

<style>

December 2006 November 2006 September 2006 August 2006 May 2006 **April 2006** March 2006 February 2006 January 2006 December 2005 November 2005 October 2005 September 2005 August 2005 July 2005 June 2005 May 2005 **April 2005** February 2005 January 2005 December 2004 November 2004 October 2004 September 2004 August 2004 July 2004 June 2004 May 2004 **April 2004** March 2004 February 2004 January 2004 December 2003 November 2003 October 2003 September 2003 August 2003

Tags

July 2003

.NET (3)
Adobe (1)
Ajax (3)
ASP.NET (1)
Atlas (1)
Bindows (8)

```
div {
   width: 500px;
   font-size: 12px;
}

p {
   width: 50%;
   font-size: lem;
}

a {
   font-size: inherit;
}

</style>

<div>
Some text with <a id=a href=#>a link</a>.
More <b style="font-size:150%">text</b>
</div></pi>
```

Computed Style Cascaded Style Inline Style

```
p width "250px" "50%" ""
p fontSize "12px" "1em" ""
a fontSize "12px" "1em" ""
b fontSize "18px" "150%" "150%"
```

Browser Support

All browsers except IE has a way to get the computed style. The way to do this is to use document.defaultView.getComputedStyle. Only IE has a way to get the cascaded style. This is done using element.currentStyle.

All browsers supports getting the inline style (even IE4) but inline style is not that interesting because it is empty unless you already set it, and in that case you don't really need to query it.

Conclusion

Back to our earlier question. What is wrong with our getStyle function? It should be pretty obvious now. It will give very different and sometimes unexpected results depending both on browsers and on the value set by the page author and even worse, set by user style sheets.

So how do we solve this? We stop supporting IE of course... seriously, with the market share IE has that is not an option. The best solution is to remove functions like these from shared code and instead add more specific functions. One can for example, often calculate the computed value based on the cascaded value and the ancestors. A good example of that is 'visibility'. Computing that

עטן <u>פעעטוווע</u> **DHTML** (12) Friends (4) Google (16) Internet Explorer (16) JavaScript (49) Microsoft (7) Mozilla (29) Music (9) Opera (2) Personal (39) Politics (3) Program Languages (3) Rant (19) Safari (3) Skateboard (1) Software (29) Uncategorized (10) Web (87) WebKit (3)

Links

Aaron Boodman
Alex Russel
Bindows
Bindows Blog
Dan Pupius
Dean Edwards
Dylan Schiemann
Emil A Eklund
WebFX

Meta

Log in Contact WordPress

Entries (RSS) | Comments (RSS)

is pretty easy. If the value is 'inherit' check the parents until a non inherit value is found. For things like left, width etc you will be better of using offsetLeft, offsetWitdh etc. There are a lot of cases and sometimes it is just not possible (or requires iterative testing with different absolute sized elements).

Comments

Override style

There is one more style type that is worth mentioning and that is the override style. This is called document.defaultView.getOverrideStyle in the W3C DOM and element.runtimeStyle in IE. It allows you to override the style property in such a way that it has (a) highest priority and (b) not serialized from the DOM when you do things like cssText and innerHTML. getOverrideStyle is not supported in Gecko so it is still only useful to IE specific code.

It is a bit strange to me that the W3C DOM never added a way to get the cascaded style. The usecases for this are few but they do exists.

What is even more strange is that MS did not realize the need for a way to get the compouted style back in the days when they believed in Trident as a platform.

This entry was posted on Friday, July 27th, 2007 at 18:54 and is filed under <u>DHTML</u>, <u>Internet Explorer</u>, <u>Mozilla</u>, <u>Rant</u>. You can follow any responses to this entry through the <u>RSS 2.0</u> feed. You can <u>leave a response</u> , or <u>trackback</u> from your own site.

24 Responses to "Computed vs Cascaded Style"

1. <u>Dean Edwards</u> Says: <u>July 27th, 2007 at 19:44</u>

```
This works on IE6 and IE7:

var PIXEL = /^\d+(px)?$/i;
function getPixelValue(element, value) {
  if (PIXEL.test(value)) return parseInt(value);
  var style = element.style.left;
  var runtimeStyle = element.runtimeStyle.left;
  element.runtimeStyle.left = element.currentStyle.left;
  element.style.left = value || 0;
  value = element.style.pixelLeft;
  element.style.left = style;
  element.runtimeStyle.left = runtimeStyle;
  return value;
}
```



2. yn Says: July 28th, 2007 at 01:50

I disagree. I find currentStyle much more useful than getComputedStyle and would have preferred non-IE browsers to implement it instead of computed styles (ideally both).

There is always a way to calculate the pixel value using the tools browsers provide. But in FF today, there simply NO WAY to know whether a width value is set in percentages, inches or pixels if the value had been set outside of the style attribute! This is a pretty unbelievable shortcoming.

The offset* and client* functions provide nearly everything you may want in terms of getting the pixel values, and the difference between them helps you calculate padding/border values. In fact they're so useful that I've rarely seen anyone opt for computed style in such situations, even if the target is FF-only.

3. <u>Erik Arvidsson</u> Says: July 28th, 2007 at 13:39

Dean: That seems to work pretty well in some cases. Which cases? It seems like it would only work for properties that have pixel* equivalent (left, top, right, bottom, width and height).

yn: How would you compute the font size given the following:

```
* { font-size: 12pt; }
```

or the background color given:

* { background-color: Highlight; }

4. All in a days work... Says: July 28th, 2007 at 18:26

[...] Computed vs Cascaded Style The offset* and client* functions provide nearly everything you want for getting the pixel values, and the difference between them helps you calculate padding/border values. I've rarely seen anyone opt for computed style in such situations, even for FF. [...]

5. yn Says:

July 29th, 2007 at 05:52

Erik: It's not about finding cases where offset* and client* won't be enough, they certainly exist. It's about what's more useful to have if you had to choose one. Why would you need the pixel value of a font set in points? I really can't think of such a scenario. If you want the height of the line in pixels, which would make more sense, that's easy to do.

On the other hand, I can easily think of multiple scenarios requiring the actual cascaded properties and I've been constantly bumping against these cases, as mentioned above.

Btw, to answer your specific questions:



```
function getRealBackgroundColor(el) {
  var oRG=document.body.createTextRange();
  oRG.moveToElementText(el);
  var iClr=oRG.queryCommandValue("BackColor");
  return "rgb("+(iClr & 0xFF)+","+((iClr & 0xFF00)>>8)+","+((iClr & 0xFF0000)>>16)+")";
}
```

Your other question was already answered by Dean, his function can take any value in any unit (except % in some cases) and return it in pixels, it's not limited to .left, he's just using it to do the conversion. so getPixelValue(document.body,"12pt") will give you the pixel value for your font under your setup.

Again, I'm not saying anything everywhere can be converted to an absolute value, I'm just saying that currentStyle is a lot more useful.

6. <u>Erik Arvidsson</u> Says:
July 30th, 2007 at 08:52

yn: Thanks, the queryCommandValue is a great example how to do this.

7. Papandreou Says:
August 3rd, 2007 at 07:21

@yn: Your getRealBackgroundColor function is really interesting. For different reasons, I've been looking for a way to get the rrggbb values corresponding to a user's CSS system colors, but I actually gave up a while ago.

However, I can only get your code to work in IE. Any chance you know a cross-browser way of accomplishing this?

8. yn Says: August 3rd, 2007 at 21:17

Papandreou: In any other browser it's trivial to get the real background color, via getComputedStyle. IE is the only one that makes this a bit of a challenge.

To create a cross-browser function simply check if the getComputedStyle method exists (FF, Opera, etc.), and if so use it, otherwise see if the createTextRange method exists (IE), and if so use it.

A *real* challenge would be to get the cascaded value ("red", "highlight", etc.) set in external CSS in Firefox.

BTW, Erik: Your article is mistaken. Opera supports .currentStyle properly, allowing you to get the cascaded style (from the article: "Only IE has a way to get the cascaded style.").

9. <u>Links for 8/6/07 [my NetNewsWire tabs]</u> Says: August 6th, 2007 at 07:38

[...] erik's weblog » Blog Archive » Computed vs Cascaded Style [...]



10. <u>Dimitri Glazkov</u> Says: August 29th, 2007 at 06:21

Another interesting tidbit is that computed style does not take into account margin collapse. For instance, the marginTop value reported does not necessarily represent the actual margin value in effect. Quick example:

Yo

Using just default styles, the computed style on body will be 8, but it's not very useful, because the effective margin is 21 (body margin collapsed in favor of h1).

11. <u>Dimitri Glazkov</u> Says: August 29th, 2007 at 06:23

Sorry, I should've escaped the markup:

<body>
<div>
<h1>Yo</h1>
</div>
</body>

12. Melikoth Says:

February 15th, 2008 at 06:49

For some reason when I use this it returns "auto" for top and left values. The object I am interrogating has no styling applied, but I was hopinh it would tell me where it was located.

13. xErath Says:

March 26th, 2008 at 04:53

Why all the trouble of defining document.defaultView.document.defaultView.document.defaultView.document.defaultView.document.defaultView.?

use .getComputedStyle directly

14. <u>Michel</u> Says: September 23rd, 2008 at 00:14

Stylesheets are class. It reduces size doe this is a page of search engines thus can better spiders. What a shame I only think that the Internet Explorer and Firefox that partially distinct process.

15. <u>Things you should know when not using a JS library « Lea Verou</u> Says: February 21st, 2009 at 17:23

[...] ones (for instance, it might return ems for a property that was specified in ems, and not pixels). Dean Edwards has thought a very clever hack to workaround this and didn't even blog about it (it's



simply a comment in a blog post of Erik [...]

16. Bill Says:

March 5th, 2009 at 00:27

@Dean Edwards: the code provided does not work properly under the following conditions in IE7 (I haven't tried it in IE6).

Create a CSS rule body { font-size: 62.5%; }

Somewhere on a page, put some text in the body of a SPAN tag.

Your code will return 625px. NOT cool! (2)

17. <u>Rob</u> Says:
March 15th, 2009 at 09:33

I had a similar problem the other day requiring the computed style in px not the currentStyle or style value and ran into the limitations of Deans hack when working with styles using %. The following function seems to correctly convert the style to px and ems and also handles styles in % and child elements where the parent is in % but the child is set to "auto". It returns an object with the style value (computed/current), size in px, sixe in em, size 1em equates to in px on that element. May not be perfect but seems to work at the moment until told otherwise

http://www.strictly-software.com/CSSStyleObject.asp

18. <u>Tomek</u> Says: August 17th, 2009 at 02:02

Very nice solution.

But there is a little bug in the hack by Dean Edwards.

I had a case where floating point numbers had to pass this hack.

It return false this was logical ive modified this hack to

if (!/^\d+(.\d*)+(px)?\$/i.test(ret)){

// do something good 🚇

have fun i hope i could help someone

- 19. <u>Dean Edwards: Convert any colour value to hex in MSIE</u> Says: October 22nd, 2009 at 10:22
 - [...] this method has been around for a while. I posted this adaptation because it does not suffer some of the [...]
- 20. 超酷带纹理网页滚动条效果 « 酷码资源-帮助你收集一些常用、经典、不易记的代码与



<u>技巧 PHP代码 ASP代码 精选JS分</u> <u>技巧</u> ❷ Says:	代码 WORDPRESS 网站SEO Flash焦点厂告 JS焦点厂告 CSS
November 2nd, 2009 at 22:02	
[] dealing with computed vs. cascaded browsers inconsistencies (that's another []	
21. <u>Convert any colour value to hex in MSIE « DesignerLinks</u> Says: <u>March 1st, 2010 at 16:18</u>	
[] this method has been aroun some of the []	d for a while. I posted this adaptation because it does not suffer
22. Lindsey Says: November 2nd, 2010 at 22:55	
systems and hacking via code computer (2) I saw something in could help me figure out if it was virusbut I am willing to laugh a	anyone that doesn't have the capability of learning computer or html might just be suckers, I really hope you haven't hacked my my files that deeply disturbed me and I was hoping someone is a practical joke or really a virus. Because it's not funny if it is a at the cleverly of putting something in the encoding of your "virus" so love me, hate me whatever
~Lindsey	
23. <u>Bitkisel Ürünler</u> Says: December 16th, 2010 at 02:41	
!'wOw'! Thank you very too muc	h
24. <u>altın çilek</u> Says: <u>January 24th, 2011 at 08:21</u>	
For some reason when I use this it returns "auto" for top and left values. The object I am interrogating has no styling applied, but I was hopinh it would tell me where it was located.	
Leave a Reply	
Name (required):	
Email (required):	
Website:	



