Get Computed Style in JavaScript

The computed style of an element incorporates external CSS (via stylesheets), inline styles and styles prescribed via JavaScript. It's most commonly used to get the default style of an element if it's not specifically assigned via styles, i.e. the default style that the browser gives an element. It's a very undocumented part of JavaScript engines, and, of course, there are multiple implementations. For more details, read this blog post.

For my Linkurious project, I needed to get the computed style of an element in a cross browser way. I came up with an implementation that works in every version of every browser (within reason, the C64 is not supported), and is easy to use.

I'll leave the details of how it works as an exercise to the reader. I will say that this is a good example of how to use a closure to improve the efficiency of a function. Notice that the cross-browser garbage is only executed once, not on each call.

```
01  //must be executed on or after the domready event, since it (may) require
    document.body to be non-null
02  var getComputedStyle = function() {
03    var func = null;
04    if (document.defaultView && document.defaultView.getComputedStyle) {
05      func = document.defaultView.getComputedStyle;
06    } else if (typeof(document.body.currentStyle) !== "undefined") {
07      func = function(element, anything) {
08        return element["currentStyle"];
09      };
10    }
11
12    return function(element, style) {
13      return func(element, null)[style];
14    }
15  }();
16
17  //and use like so:
18  getComputedStyle(document.getElementById("foo"), "display"); //might return
    "inline-block"
```

Leave a Reply

Name, required

Email (will not be published), required

Website (if present)

Submit

Meta

Search   Find

PDFmyURL.com