Why can't you say in a script block?

devblogs.microsoft.com/oldnewthing/20060605-00

June 5, 2006

Raymond Chen

Because it ends the script block, of course. Duh, what's so hard about that?

Because if you have script that generates script, you'll find yourself caught out if you're not careful. For example, you can't say

```
document.write("<SCRIPT>blahblah</SCRIPT>");
```

in a script block because the HTML parser will see the `</SCRIPT>` and conclude that your script block is over. In other words, the script block extends as far as the highlighted section below:

```
<SCRIPT>
document.write("<SCRIPT>blahblah</SCRIPT>");
</SCRIPT><!-- mismatched tag -->
```

The parser doesn't understand "quoted strings" or "comments" or anything like that. It just looks for the nine characters "<", "/", "S", "C", "R", "I", "P", "T", and ">". When it sees them, it decides that the script block is over and returns to HTML parsing.

Why doesn't the parser understand quoted string?

Well, in order to parse quoted strings, you have to be able to parse comments:

```
<SCRIPT>
/* unmatched quotation mark " ignored since it's in a comment */
</SCRIPT><!-- you might expect this to end the script block -->
```

But every language has a different comment syntax. JScript uses `/* ... */` and `//`, Visual Basic uses `'`, perl uses `#`, and so on. And even if you got comments figured out, you also would need to know how to parse quoted strings. Perl, for example, has a very large vocabulary for expressing quoted strings, from the simple `"..."` and `'...'` to the idiosyncratic `qq:...:`. And I lied about the JScript comment and quotation syntax; it's actually more complicated than I suggested:

```
/"//"</SCRIPT>is this inside or outside quotes?
```

That first quotation mark is itself quoted and does not count as a "beginning of quoted string" marker. And the `//` sequence is not a comment marker. The first slash in the `//` sequence ends the regular expression, and the second is a division operator.

It would be unreasonable to expect the HTML parser to be able to understand every language both present and future. (At least not until clairvoyance has been perfected.)

```
<SCRIPT>
'is this a quoted string?'</SCRIPT>
Is this inside or outside the script block?
'<SCRIPT>' is this a new script block
or the continuation of the previous one?
</SCRIPT>
```

One "solution" would be to require all languages to conform to one of a fixed number of quotation and comment syntaxes. Nevermind that not even JScript conforms to the basic syntax, as we saw above, thanks to the complicated quotation rules implied by regular expression shorthand. And do you really want all HTML parsers to understand perl?

Another "solution" would be to have the language processor do the parsing and tell the HTML parser where the `</SCRIPT>` tag is. This has its own problems, however. First, it means that the HTML parser would still have to load the language parser even for `DEFER` script blocks, which sort of defeats one of the purposes of `DEFER`. Even worse, it means that a web page that used a language that the system didn't support would become unparseable:

```
<SCRIPT LANG="unknown-language">
Lorem ipsum dolor sit amet,
...
```

If a language parser were required to locate the end of the script block, it would be impossible to parse past this point.

So how do you work around this aspect of HTML parsing? You have to find an alternate way of expressing the string you want. Typically, this is done by breaking in up into two strings that you then reassemble:

```
document.write("<SCRIPT>blahblah</SCRI"+"PT>");
```

Raymond Chen

Follow