Command line parsers look at what you typed, not what what you typed looks like

devblogs.microsoft.com/oldnewthing/20090707-00

July 7, 2009

Raymond Chen

Command line parsers are stricter than human beings. The computer cares about what you typed. If you type something else that superficially resembles what you want, but is not actually what you want, then you won't get what you want, even though it sure looks like what you want.

I covered a special case of this topic earlier when I described smart quotes as the hidden scourge of text meant for computer consumption. You think you typed a quotation mark, but the editor secretly "improves" it from U+0022 to U+201C. Then you paste the text into a command line and you get strange output because the command line parser uses U+0022 as the quotation mark, not characters like U+201C which physically resemble U+0022 but aren't. A human being would say, "Yeah, that's a quotation mark, we'll let it slide." A command line parser knows what it wants and accepts no substitutes.

Even though my original examples are for command line use, the principle applies elsewhere, and I'm going to use CSS selectors as my example, because the command line programs used in my original examples are internal Microsoft tools, and explaining their command line syntax would distract from the message. On the other hand, everybody can look up CSS selector syntax. (There is an opportunity for a snarky comment here, but I'm going to withhold the pre-emptive snarky comment in the hopes that my readers are mature enough to not go for the cheap drive-by insult. I am almost always disappointed by the result, but I try again every so often to see if the environment has changed.)

Here is an imaginary question/answer session inspired by actual events:

> **From**: Xxxx Xxxxx
>
> How do I make a CSS rule apply only to elements with a specific tag and a specific class? Specifically, I want a rule to apply only to elements of the form `<SPAN CLASS="Name">`.

> **From**: Yyyyy Yyyyyy
>
> You use `SPAN.Name`.

> **From**: Xxxx Xxxxx
>
> I tried that, but it doesn't work.
>
> `SPAN. Name { color: red; }`

This person became one of <u>those people who are not interested in why something works but are only interested in the magic phrase they need to type</u>. Whether or not you believe that the "just tell me what to type" mindset is acceptable, you have to concede that if you choose to adopt that mindset, *you need to actually type what people tell you to type*.

In this case, the person decided that the punctuation rules for CSS are the same as the punctuation rules for English, and since English allows (and even encourages) a space after a period, then CSS should also allow it. "I mean, it looks so much better on the screen with a space after the period." (I bet these are the same people who <u>move cities around on a map in order to produce a more visually-pleasing arrangement</u>.)

Or maybe the person decided that the punctuation rules for CSS were the same as C++. C++ allows spaces on either side of the field selector operator. If C++ allows it, then so too should CSS.

Of course, neither case is true. CSS has its own rules for parsing, and in CSS, spaces are significant.

What I found even more mind-boggling was another question/answer session. I'm going to use the same question and answer for illustrative purposes, but in reality, the question was different.

> **From**: Wwww Wwwww
>
> How do I make a CSS rule apply only to elements with a specific tag and a specific class? Specifically, I want a rule to apply only to elements of the form `<SPAN CLASS="Name">`.

> **From**: Yyyyy Yyyyyy
>
> You use `SPAN.Name`.

> **From**: Wwww Wwwww
>
> I tried that, but it doesn't work.
>
> `.Name SPAN { color: red; }`

I don't know what possessed this second person to take the instructions and then *scramble the pieces* on the assumption that, hey, the order of the pieces of the selector aren't important, as long as they're all there, right?