Regular expressions and the dreaded *? operator

devblogs.microsoft.com/oldnewthing/20040325-00

March 25, 2004

Raymond Chen

The regular expression *? operator means “Match as few characters as necessary to make this pattern succeed.” But look at what happens when you mix it up a bit: `".*?"` This pattern matches a quoted string containing no embedded quotes. This works because the first quotation mark starts the string, the .*? gobbles up everything in between, and then the second quotation mark eats the close-quote. (Note how this differs from `".*"`, which uses a greedy match. This time, the .* operator is perfectly happy to gobble up quotation marks, as long as it leaves one to match the second quotation mark in the pattern.) Okay, great, now let’s make a small change to the above pattern: `".*?">` All I did was stick a > at the end of the pattern. This would therefore match a quoted string (containing no quotation marks) followed by a > character, right? Wrong. There’s nothing in .*? that says “no quotation marks allowed”. It just says “Don’t match more than you need to.” But there are strings where it needs to match a quotation mark. Consider:

`"`	`hello"world`	`"`	`>`
`"`	`.*?`	`"`	`>`

Notice that here, the .*? pattern matched the inner quotation mark because that was the only way to make the pattern match successfully. (“I wouldn’t have done it, but you forced me!”) Even smart people make this mistake. If you really don’t want quotation marks to match the .*? then you need to say so. `"[^"]*">` This means “Match a quotation mark, then zero or more characters that aren’t quotation marks, then another quotation mark, and then a greater-than.”

[Raymond is currently on vacation; this message was pre-recorded.]

Raymond Chen

Follow