

# Forum:[Hack Forums](#)

Thread: [ReterBatch Editor | A Batch Windows Script Text Editor | C# | 0.1](#)

**Skofos** - July 18, 2019, 5:42 pm

so you know im so bored so i make this tool. just for fun right.  
basically why i made this? cuz [this section](#) so quite if you're Batch Script coder please use this  
and i will be happy for this :D  
this project was uploaded on GitHub so take it easy

Spoiler ([Click to View](#))

[Here the Source Code](#)

[Here The Release Version](#)

Donate me BTC if you're rich: 3GBbVr59Z6FG31DJMymqN6WZVs82YDAEx7

[Attention Required! | Cloudflare](#)

Credits:

[SharpDevelop](#) - Text Editor

**gigajew** - July 18, 2019, 8:53 pm

You should add some syntax highlighting to it

**Skofos** - July 18, 2019, 9:08 pm

(Jul 18, 2019 08:53 PM)gigajew Wrote: [\[CITE\]](#) You should add some syntax highlighting to it  
yea i was thinking to add syntax highlighting but idk how to code it lol

**DNA** - July 23, 2019, 9:54 pm

(Jul 18, 2019 09:08 PM)Skofos Wrote: [\[CITE\]](#) yea i was thinking to add syntax highlighting but idk  
how to code it lol

You can do this by using String.Contains and/or parsing in a constant loop

**Skofos** - July 25, 2019, 8:11 pm

(Jul 23, 2019 09:54 PM)Jelly Wrote: [\[CITE\]](#) You can do this by using String.Contains and/or parsing  
in a constant loop

k maybe next time ill update this shit code editor XD

**DNA** - July 25, 2019, 8:26 pm

(Jul 25, 2019 08:11 PM)Skofos Wrote: [\[CITE\]](#) k maybe next time ill update this shit code editor XD  
I'm gonna check it out after I finish my universal bot, looks good dude

**Skofos** - July 25, 2019, 9:15 pm

(Jul 25, 2019 08:26 PM)Jelly Wrote: [\[CITE\]](#) I'm gonna check it out after I finish my universal bot,  
looks good dude

nice to see another programmer in this forum ?

**DNA** - July 25, 2019, 9:38 pm

(Jul 25, 2019 09:15 PM)Skofos Wrote: [\[CITE\]](#) nice to see another programmer in this forum ?  
The more the merrier, we all have knowledge of things that others may not.

As my signature says aha

**Fhack** - July 25, 2019, 11:12 pm

(Jul 18, 2019 09:08 PM)Skofos Wrote: [\[CITE\]](#) yea i was thinking to add syntax highlighting but idk how to code it lol  
Just use scintilla.net lib it's open source !

**Skofos** - July 26, 2019, 2:34 am

(Jul 25, 2019 09:38 PM)Jelly Wrote: [\[CITE\]](#) The more the merrier, we all have knowledge of things that others may not.

As my signature says aha  
lol, nice words

(Jul 25, 2019 11:12 PM)Fhack Wrote: [\[CITE\]](#) Just use scintilla.net lib it's open source !  
thank you for sharing this lib. but i have no time to update it XD

**X-Warz** - August 4, 2019, 1:05 am

nice project that you made! keep it up

**Skofos** - August 8, 2019, 8:35 pm

(Aug 4, 2019 01:05 AM)X-Warz Wrote: [\[CITE\]](#) nice project that you made! keep it up  
thank you man

**0xDEAD10CC** - August 10, 2019, 3:49 pm

(Jul 23, 2019 09:54 PM)Jelly Wrote: [\[CITE\]](#) You can do this by using String.Contains and/or parsing in a constant loop

This would be the naive way to do it. Language syntax is much more than just word parsing. Context plays a role in the highlighting as well, so this task is usually done by lexical parsing/analysis. You might also want to consider Regex if you're not willing to do all of that manually; this may solve basic language syntax highlighting problems as a minimum viable product approach.

**DNA** - August 10, 2019, 5:30 pm

(Aug 10, 2019 03:49 PM)0xDEAD10CC Wrote: [\[CITE\]](#) This would be the naive way to do it. Language syntax is much more than just word parsing. Context plays a role in the highlighting as well, so this task is usually done by lexical parsing/analysis. You might also want to consider Regex if you're not willing to do all of that manually; this may solve basic language syntax highlighting problems as a minimum viable product approach.

Regex is probably slow and inefficient in this case and I was only mentioning Contains and Parsing in order to pick out particulars, i.e variables and other expressions being highlighted differently

**0xDEAD10CC** - August 10, 2019, 5:33 pm

(Aug 10, 2019 05:30 PM)Jelly Wrote: [\[CITE\]](#) Regex is probably slow and inefficient in this case and I was only mentioning Contains and Parsing in order to pick out particulars, i.e variables and other expressions being highlighted differently

Slow by what measure? Who writes Mb's worth of batch scripting? It'll be far more versatile than what you've suggested. Additionally SublimeText uses Regex for their language definitions for the syntax highlighting.

Contains() also won't be based on word boundaries, meaning it will catch keywords within other text which is probably not what you want, so Regex is already a more appropriate suggestion. Your

suggestion was flawed from the start.

The only real way to achieve this though is through lexical analysis.

**DNA** - August 11, 2019, 2:25 am

(Aug 10, 2019 05:33 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Slow by what measure? Who writes Mb's worth of batch scripting? It'll be far more versatile than what you've suggested. Additionally SublimeText uses Regex for their language definitions for the syntax highlighting.

Contains() also won't be based on word boundaries, meaning it will catch keywords within other text which is probably not what you want, so Regex is already a more appropriate suggestion. Your suggestion was flawed from the start.

The only real way to achieve this though is through lexical analysis.

Well for example, I would be converting this to handle other formats, not just batch scripts, otherwise it wouldn't be of much use because as you mentioned, who writes Mb's worth of batch scripts

Contains will assist in catching parts of the phrases but not entirely, which is why it would assist with syntaxes, you would obviously need to further define it.

However if you're sole purpose is to use your editor for one thing only, creating small scripts, then I guess Regex is appropriate.

**0xDEAD10CC** - August 11, 2019, 1:35 pm

(Aug 11, 2019 02:25 AM)Jelly Wrote: [\[CITE\]](#) Well for example, I would be converting this to handle other formats, not just batch scripts, otherwise it wouldn't be of much use because as you mentioned, who writes Mb's worth of batch scripts

Contains will assist in catching parts of the phrases but not entirely, which is why it would assist with syntaxes, you would obviously need to further define it.

However if you're sole purpose is to use your editor for one thing only, creating small scripts, then I guess Regex is appropriate.

>> Contains will assist in catching parts of the phrases but not entirely, which is why it would assist with syntaxes, you would obviously need to further define it.

I don't see what you mean here. Contains() will match "int" for instance in all of the following:

Quote:var interpret;

Quote:var x = "interpret";

Quote:func interpret(x, y, z) { }

etc...

This is why Contains() is a terrible idea. It returns a boolean so you don't even know the position of where the text was found to do anything with it either.

SublimeText has no problem using regex for syntax highlighting larger files. Regex would be fine for more than one syntax because syntax definitions for multiple languages are never defined all in the same context. Language definitions for any syntax highlighting method are all separate for each language. (i.e. SublimeText has a tmlanguage file I believe for every language IIRC.)

Example:

There's regex everywhere in these files.

**DNA** - August 11, 2019, 3:16 pm

(Aug 11, 2019 01:35 PM)0xDEAD10CC Wrote: [\[CITE\]](#) >> Contains will assist in catching parts of the phrases but not entirely, which is why it would assist with syntaxes, you would obviously need to further define it.

I don't see what you mean here. Contains() will match "int" for instance in all of the following:  
etc...

This is why Contains() is a terrible idea. It returns a boolean so you don't even know the position of where the text was found to do anything with it either.

SublimeText has no problem using regex for syntax highlighting larger files. Regex would be fine for more than one syntax because syntax definitions for multiple languages are never defined all in the same context. Language definitions for any syntax highlighting method are all separate for each language. (i.e. SublimeText has a tmlanguage file I believe for every language IIRC.)

Example:

There's regex everywhere in these files.

Exactly, using Contains to first tell if the syntax contains what you are looking for and if so, parse it..

I.e. Logically speaking, If the string contains the string "int" and it is a phrase of its own, then check this prior to parsing

```
Code bool containsInt = (line[x].Contains("int")) ? true : false;  
if (containsInt){  
//Etc etc  
}
```

Also it depends on what you are using it for, for example If I was to use this for JS, I may not use regex, due to the amount of variables I would need to take into consideration.

As you can see from your screenshot, regex can be messy as well

**0xDEAD10CC** - August 11, 2019, 5:56 pm

(Aug 11, 2019 03:16 PM)Jelly Wrote: [\[CITE\]](#) Exactly, using Contains to first tell if the syntax contains what you are looking for and if so, parse it..

I.e. Logically speaking, If the string contains the string "int" and it is a phrase of its own, then check this prior to parsing

```
Code bool containsInt = (line[x].Contains("int")) ? true : false;  
if (containsInt){  
//Etc etc  
}
```

Also it depends on what you are using it for, for example If I was to use this for NodeJS, I may not use regex, due to the amount of variables I would need to take into consideration.

As you can see from your screenshot, regex can be messy as well

... ? Now why would you do that over something which returns useful information like IndexOf()?

Contains() is a very silly approach.

Variables don't have anything to do with syntax, identifiers are the important subject, and context; none of which you can do in an effective manner whatsoever with Contains().

Not sure why you use ternary here:

Code (line[x].Contains("int")) ? true : false

Contains() already returns a bool - none of that makes sense, including the strategy. Regex is a dynamic approach, try doing what that Regex is doing with Contains() and lets see how much more messy the implementation would be... "Messy" is a relative term here, and without any comparison, saying something is "messy" is not useful.

NodeJs is also not a language, it's just a Javascript environment, which means it's still Javascript; SublimeText uses Regex for this too.

**DNA** - August 11, 2019, 6:44 pm

(Aug 11, 2019 05:56 PM)0xDEAD10CC Wrote: [\[CITE\]](#) ... ? Now why would you do that over something which returns useful information like IndexOf()? Contains() is a very silly approach.

Variables don't have anything to do with syntax, identifiers are the important subject, and context; none of which you can do in an effective manner whatsoever with Contains().

Not sure why you use ternary here:

Code (line[x].Contains("int")) ? true : false

Contains() already returns a bool - none of that makes sense, including the strategy. Regex is a dynamic approach, try doing what that Regex is doing with Contains() and lets see how much more messy the implementation would be... "Messy" is a relative term here, and without any comparison, saying something is "messy" is not useful.

NodeJs is also not a language, it's just a Javascript environment, which means it's still Javascript; SublimeText uses Regex for this too.

Variables have everything to do with syntax, you can't have a syntax without a variable.. it was simply an example of a predecessor to determining if, what or how the condition is.

That statement was returning a bool for if it contains the string int in the line, of course it makes sense.. especially considering you can add the ampersand operands adding other checks to it..

I said NodeJS as I responded to you after just waking up, had four hours sleep, I'll edit the post

Sublime is a terrible example to use here as it only furthers my point that regex is slow and inefficient.

**0xDEAD10CC** - August 11, 2019, 7:04 pm

(Aug 11, 2019 06:44 PM)Jelly Wrote: [\[CITE\]](#) Variables have everything to do with syntax, you can't have a syntax without a variable.. it was simply an example of a predecessor to determining if, what or how the condition is.

That statement was returning a bool for if it contains the string int in the line, of course it makes sense.. especially considering you can add the ampersand operands adding other checks to it..

I said NodeJS as I responded to you after just waking up, had four hours sleep, I'll edit the post

Sublime is a terrible example to use here as it only furthers my point that regex is slow and

inefficient.

Yes you can:

```
Code public delegate void MyDelegate();
```

^ This isn't a variable. And neither are these:

```
Code namespace XYZ {
```

```
}
```

```
Code class XYZ {
```

```
}
```

etc...

These are all still syntax. Syntax definitions are even for non-programming languages that don't contain any variables too, so that's not even a valid statement under all circumstances.

You simply don't know what you're saying. Have you ever actually written anything like this or done anything with lexical parsing or analysis? It seems less and less likely with each of your suggestions. The bool makes absolutely no sense, provided that there is an `IndexOf()` which returns -1 if not found, or `>= 0` if found (which would be your bool), but also return information about where it was found in the same call. What would the checks do? If you check for "int" in a line of code, `Contains()` is entirely useless because it just tells you that it found that text, it doesn't tell you whether it was part of a string, part of an identifier, or whether it was the identifier, hence why lexical analysis and parsing out tokens is the real world approach.

What are you going to do with `Contains()`? -- check every identifier then validate one by one whether it is actually an identifier and not part of a string or function or variable name? You don't even realize how silly that is...

Quote: Sublime is a terrible example to use here as it only furthers my point that regex is slow and inefficient.

In what way? As I said, the highlighting in SublimeText is faster than VisualStudio on my machines. Where do you draw your conclusions from? :/ I suggest you do some reading.

**DNA** - August 11, 2019, 7:25 pm

(Aug 11, 2019 07:04 PM)0xDEAD10CC Wrote: [CITE](#) Yes you can:

```
Code public delegate void MyDelegate();
```

^ This isn't a variable. And neither are these:

```
Code namespace XYZ {
```

```
}
```

```
Code class XYZ {
```

```
}
```

etc...

These are all still syntax. Syntax definitions are even for non-programming languages that don't contain any variables too, so that's not even a valid statement under all circumstances.

You simply don't know what you're saying. Have you ever actually written anything like this or done anything with lexical parsing or analysis? It seems less and less likely with each of your suggestions. The bool makes absolutely no sense, provided that there is an `IndexOf()` which returns -1 if not found, or `>= 0` if found (which would be your bool), but also return information about where it was

found in the same call. What would the checks do? If you check for "int" in a line of code, Contains() is entirely useless because it just tells you that it found that text, it doesn't tell you whether it was part of a string, part of an identifier, or whether it was the identifier, hence why lexical analysis and parsing out tokens is the real world approach.

What are you going to do with Contains()? -- check every identifier then validate one by one whether it is actually an identifier and not part of a string or function or variable name? You don't even realize how silly that is...

In what way? As I said, the highlighting in SublimeText is faster than VisualStudio on my machines. Where do you draw your conclusions from? :/ I suggest you do some reading.

Ok champ, I don't know what I'm saying, except that's not what a syntax is, a syntax comprises of everything from methods to the parameters that follow, to each individual equation you write and each declaration you wrote for it, but I'm not going to nitpick, I'm on a mobile and I don't care much for being a keyboard warrior.

The above Boolean does make sense, as I mentioned you can append a statement, meaning it's not redundant and IndexOf wouldn't yield the same results

I.e.

```
(line[x].Contains("int")&&theOtherVariable) ? true : false;
```

You keep comparing apples and oranges however, but as mentioned, I'm not one to argue so I'm going to drop it here.

Good luck

0xDEAD10CC - August 11, 2019, 8:58 pm

(Aug 11, 2019 07:25 PM)Jelly Wrote: [\[CITE\]](#) Ok champ, I don't know what I'm saying, except that's not what a syntax is, a syntax comprises of everything from methods to the parameters that follow, to each individual equation you write and each declaration you wrote for it, but I'm not going to nitpick, I'm on a mobile and I don't care much for being a keyboard warrior.

The above Boolean does make sense, as I mentioned you can append a statement, meaning it's not redundant and IndexOf wouldn't yield the same results

I.e.

```
(line[x].Contains("int")&&theOtherVariable) ? true : false;
```

You keep comparing apples and oranges however, but as mentioned, I'm not one to argue so I'm going to drop it here.

Good luck

[https://en.wikipedia.org/wiki/Syntax\\_\(programming\\_languages\)](https://en.wikipedia.org/wiki/Syntax_(programming_languages))

Quote:In computer science, the syntax of a computer language is the set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in that language.[1] This applies both to programming languages, where the document represents source code, and to markup languages, where the document represents data.

An empty namespace is still valid syntax, I have no idea where you get your ideas from but they are wrong. Show me references that say otherwise.

It's quite evident you don't know what you're talking about at all; this is basic computer terminology that you can't quite grasp. Syntax has nothing to do with variables, methods, and/or parameters

because it's much broader than that. It's just a set of rules that defines how something is correctly structured textually, nothing more. There isn't even a requirement for a syntax to be construed as only applying to programming languages either. Therefore, it doesn't need to include variables or methods or parameters, and those components may not even exist in the first place for some markup languages. (This is proof for why a statement like "Variables have everything to do with syntax, you can't have a syntax without a variable.." is 100% false.)

Contains() and IndexOf() != -1 would yield the exact same results btw, and these are also fundamental string functions that most beginners should know... And once again, you use the ternary operator in a completely redundant way. Explain how this:

Code (line[x].Contains("int")&&theOtherVariable) ? true : false;

Solves a syntax highlighting task? Until then you should do some further reading and look into lexical analysis as I've explained throughout this thread ( [https://en.wikipedia.org/wiki/Lexical\\_analysis](https://en.wikipedia.org/wiki/Lexical_analysis)). I would love to see how someone would implement an efficient syntax highlighter like that, or even one that works at all for even a few simple languages.

There's enough evidence in this thread alone to prove that you don't have much programming experience, not sure why you're arguing with someone who has been programming for the 15+ years... Reading through your posts here's another example:

(Jul 25, 2019 11:40 AM)Jelly Wrote: [\[CITE\]](#) I've had bad experiences with while loops, to the point where I've had to hard boot my PC.

The problem would have never been with the while loop, it would've been with the code you wrote if that was the issue. To blame a language construct for your computer reboot requirement is ridiculously senseless enough, and shows a lack of critical thinking and problem solving skills.

DNA - August 11, 2019, 9:24 pm

(Aug 11, 2019 08:58 PM)0xDEAD10CC Wrote: [\[CITE\]](#) [https://en.wikipedia.org/wiki/Syntax\\_\(programming\\_languages\)](https://en.wikipedia.org/wiki/Syntax_(programming_languages))

An empty namespace is still valid syntax, I have no idea where you get your ideas from but they are wrong. Show me references that say otherwise.

It's quite evident you don't know what you're talking about at all; this is basic computer terminology that you can't quite grasp. Syntax has nothing to do with variables, methods, and/or parameters because it's much broader than that. It's just a set of rules that defines how something is correctly structured textually, nothing more. There isn't even a requirement for a syntax to be construed as only applying to programming languages either. Therefore, it doesn't need to include variables or methods or parameters, and those components may not even exist in the first place for some markup languages. (This is proof for why a statement like "Variables have everything to do with syntax, you can't have a syntax without a variable.." is 100% false.)

Contains() and IndexOf() != -1 would yield the exact same results btw, and these are also fundamental string functions that most beginners should know... And once again, you use the ternary operator in a completely redundant way. Explain how this:

Code (line[x].Contains("int")&&theOtherVariable) ? true : false;

Solves a syntax highlighting task? Until then you should do some further reading and look into lexical analysis as I've explained throughout this thread ( [https://en.wikipedia.org/wiki/Lexical\\_analysis](https://en.wikipedia.org/wiki/Lexical_analysis)). I would love to see how someone would implement an efficient syntax highlighter like that, or even one that works at all for even a few simple languages.

There's enough evidence in this thread alone to prove that you don't have much programming experience, not sure why you're arguing with someone who has been programming for the 15+ years... Reading through your posts here's another example:

The problem would have never been with the while loop, it would've been with the code you wrote if that was the issue. To blame a language construct for your computer reboot requirement is ridiculously senseless enough, and shows a lack of critical thinking and problem solving skills.

For someone who claims to have 15 years of experience, you seem to lack an understanding of anything of what was said and would rather engage in mindless debates comparing apples and oranges.

You're more focused on creating a debate over something that the other party can't be fucked debating in.

At this point I'd be more inclined to think you have less than 5 years experience because you seem to be ignorant to the fact that there will always be multiple and more efficient ways to do things .

Oh noes, you pulled up a completely unrelated comment about how I got stuck in an infinite loop, do go on about how this makes my 14 years of experience anything less than what it is because I prefer not to use dangerous code when doing complex equations.

Link all the Wikipedia articles you want, just goes to show you're more interested in engaging in pointless debates and citing irrelevant information.

As mentioned, I'm finished with this debate as I don't wish you trying to debate something that wouldn't be as efficient and you wouldn't have as much control as you would by using a regex.

Go create another alt account and post a "tutorial" subject on regex like you have before. Seems like you're carrying on your [little debate](#) again, I'm really not interested.

For the last time, done here.  
Have a good day

0xDEAD10CC - August 11, 2019, 10:05 pm

(Aug 11, 2019 09:24 PM)Jelly Wrote: [\[CITE\]](#) For someone who claims to have 15 years of experience, you seem to lack an understanding of anything of what was said and would rather engage in mindless debates comparing apples and oranges.

You're more focused on creating a debate over something that the other party can't be fugged debating in.

At this point I'd be more inclined to think you have less than 5 years experience because you seem to be ignorant to the fact that there will always be multiple and more efficient ways to do things .

Oh noes, you pulled up a completely unrelated comment about how I got stuck in an infinite loop, do go on about how this makes my 14 years of experience anything less than what it is because I prefer not to use dangerous code when doing complex equations.

Link all the Wikipedia articles you want, just goes to show you're more interested in engaging in pointless debates and citing irrelevant information.

As mentioned, I'm finished with this debate as I don't wish you trying to debate something that wouldn't be as efficient and you wouldn't have as much control as you would by using a regex.

Where is my lack of understanding exactly? I have just proved that the majority of what you have

just said is all wrong.

1. Where are your references
2. How is that link irrelevant? (Can you even read?)
3. And what does comprehension have to do with programming anyways, even if what you claim is true? ?

Your reasoning skills are on par with the average preschool kid.

If you truly believe that `Contains()` is efficient then most people here already have the capability to correctly assume that you don't know what you're saying... You might not even understand the relevance due to your clear lack of knowledge on this subject, but you're just babbling nonsense. This was a true debate on a real topic, and lexical analysis is essentially the de facto standard when it comes to parsing syntax for any reason. Do your own research if you don't believe me because that's something you really need to do, otherwise go spread your poor information elsewhere because nobody is getting anywhere with `Contains()` for this task lol.

People with even 2 years experience would have more sense than to blame a while loop for a computer problem. I've been a member of this forum for longer than you've even known about programming most likely. ./ Yes, while loops are incredibly dangerous... ? 14 years and you've learned absolutely nothing, that should be more embarrassing in itself to be honest if it's indicative of your learning pace.

**DNA** - August 12, 2019, 2:51 am

(Aug 11, 2019 10:05 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Incoherent rambling if a child reduced to avoid recaptcha

No, your assumptions should be embarrassing.

You're assuming what you say is correct because I don't bother giving you a novel response on my phone.

You've done nothing but waste your time by explaining to someone, who has been programming for 14 years, what your definition of sh\*t is.

I could read the condescension in between, but again, since I didn't give you the time of day to read your novels, I certainly wouldn't this time.

I shouldn't need to dumb these things down for you, if you cannot even understand what a syntax is, what point is there of even correcting you, it's a waste of time.

Irony is you say you say I am babbling, yet you carry on an argument that 1: Would never be brought to fruition without you, and 2: I'm simply not interested in  
Are you that desperate for human contact?

"Your reasoning skills are on par with the average preschool kid."

Yea, You are definitely a child and it's visible by how you conduct yourself, you turn something as simple as a comment about how to determine a predecessor to an conversation where one person just doesn't care and the other is trying to hard, if that was too hard for you to follow, you're the latter.

Again, good luck.

**0xDEAD10CC** - August 12, 2019, 9:49 am

(Aug 12, 2019 02:51 AM)Jelly Wrote: [\[CITE\]](#) No, your assumptions should be embarrassing.

You're assuming what you say is correct because I don't bother giving you a novel response on my

phone.

You've done nothing but waste your time by explaining to someone, who has been programming for 14 years, what your definition of sh\*t is.

I could read the condescension in between, but again, since I didn't give you the time of day to read your novels, I certainly wouldn't this time.

I shouldn't need to dumb these things down for you, if you cannot even understand what a syntax is, what point is there of even correcting you, it's a waste of time.

Irony is you say you say I am babbling, yet you carry on an argument that 1: Would never be brought to fruition without you, and 2: I'm simply not interested in  
Are you that desperate for human contact?

"Your reasoning skills are on par with the average preschool kid."

Yea, You are definitely a child and it's visible by how you conduct yourself, you turn something as simple as a comment about how to determine a predecessor to an conversation where one person just doesn't care and the other is trying to hard, if that was too hard for you to follow, you're the latter.

Again, good luck.

>> You're assuming what you say is correct because I don't bother giving you a novel response on my phone.

Lol ? .. If that's the conclusion that you draw from it, you're not that smart at all. The information is out there, you just don't want to search for it, which is why I've given you all the information you would need to see that your statements are incorrect. The fact that you still can't see it makes me think you're incapable of understanding, or you're in denial of facts that disprove your opinions, which is ridiculous.

I see no response to the segments of my posts that are relevant to the thread, which is a clear indication that you have nothing to say because you know you are wrong.

Skids these days. :/

edit: Once again, no variables or methods here, or parameters... Still syntax:

I don't know how much more obvious things can get. If equivocation is your tactic to get people to believe that you actually even have at least a couple years experience programming, then you're only fooling yourself.

**MavenGradle** - August 13, 2019, 9:00 am

Mr. Hexadecimal 10 dead creditcards makes some very valid arguments (I don't know why you keep 10 dead creditcards around but hey, better than 10 dead toenails). Contains would be a pretty safe way to get yourself in serious trouble. Imagine the sheer length of the if statements that would be required, not to mention the performance loss. Can it be done using Contains? Of course it can. But you would need more than just contains because you need to have the position at which a certain element is present in order to highlight it. Which brings us back to indexOf. Granted, 0xDEAD10CC's argument for int checking can be mitigated by checking the length of the input. Again, this is highly error prone stuff, definitely you'd need thousands of lines of code to effectively go where you want it to go so in a realistic setting you would not use the Contains method to write your syntax/code highlighter.

Furthermore, [I would implore you to take a look at the inner code of the Contains method \(can be found in System\)](#). You will find that at the end it calls IndexOf. Peculiar. It almost seems like @0xDEAD10CC is trying to tell you how to properly implement a code/syntax highlighter (well, not 'properly', but feasibly). What he failed to mention however, is that Contains uses an ordinal comparison method, which means it will check case sensitivity. Now the question arises as to how in the seven seas you're going to use string.Contains for a language such as SQL, Fortran or ABAP.

In any case, I highly recommend all parties to take a look at ANTLR:  
<https://www.antlr.org/>

And of course, someone has already written an excellent library capable of syntax/code highlighting and even completion called AvalonEdit:

<https://github.com/icsharpcode/AvalonEdit/blob/master/ICSharpCode.AvalonEdit/Highlighting/HighlightingEngine.cs>

HighLightingRule being of course the class that contains the regular expression that contains the pattern and a color property that will be used to effectively highlight said pattern in the control:

<https://github.com/icsharpcode/AvalonEdit/blob/master/ICSharpCode.AvalonEdit/Highlighting/HighlightingRule.cs>

There's more to this than just these two classes but please by all means go take a look:  
<https://github.com/icsharpcode/AvalonEdit>

Now I hope we can all continue living happy lives without having the need to call each other kids. You better grow some thick skin as a programmer because people will always offer you better suggestions to complete a certain task. The way in which you respond to said improvement really tells volumes about what kind of person you are. Hence, grow some thick skin, do what has to be done and do it properly.

**JUSTIN** - August 13, 2019, 9:37 am

using string contains for syntax highlighting is so logically flawed that i fail to understand how an argument even occurred here

**MavenGradle** - August 13, 2019, 9:41 am

(Aug 13, 2019 09:37 AM)JUSTIN Wrote: [\[CITE\]](#) using string contains for syntax highlighting is so logically flawed that i fail to understand how an argument even occurred here

I can think of a couple of reasons: Lack of sleep, alcoholism, lack of experience in the programming field, Xanax, getting hit in the head.

No offense to the person who suggested it, we all make mistakes and we're all hoping you learn something from it.

**0xDEAD10CC** - August 13, 2019, 8:39 pm

(Aug 13, 2019 09:00 AM)MavenGradle Wrote: [\[CITE\]](#) Mr. Hexadecimal 10 dead creditcards makes some very valid arguments (I don't know why you keep 10 dead creditcards around but hey, better than 10 dead toenails). Contains would be a pretty safe way to get yourself in serious trouble. Imagine the sheer length of the if statements that would be required, not to mention the performance loss. Can it be done using Contains? Of course it can. But you would need more than just contains because you need to have the position at which a certain element is present in order to highlight it. Which brings us back to indexOf. Granted, 0xDEAD10CC's argument for int checking can be mitigated by checking the length of the input. Again, this is highly error prone stuff, definitely you'd

need thousands of lines of code to effectively go where you want it to go so in a realistic setting you would not use the Contains method to write your syntax/code highlighter.

Furthermore, [I would implore you to take a look at the inner code of the Contains method \(can be found in System\)](#). You will find that at the end it calls IndexOf. Peculiar. It almost seems like @0xDEAD10CC is trying to tell you how to properly implement a code/syntax highlighter (well, not 'properly', but feasibly). What he failed to mention however, is that Contains uses an ordinal comparison method, which means it will check case sensitivity. Now the question arises as to how in the seven seas you're going to use string.Contains for a language such as SQL, Fortran or ABAP. Not really necessary - the default overload where the comparison is not specified, matches the comparison used by Contains() (ordinal). The comparison I provided didn't even show the parameters though since string comparers weren't the main point that I was trying to illustrate; only how to check if something is "contained" within a string like the Contains() method does, since he said IndexOf() couldn't solve the same task. He moaned about the detail in my posts so I attempted to keep it as on point as I could, yet he couldn't comprehend the information apparently. If he was ambitious enough to look on MSDN, he would have found this information if he checked the other overloads or read the Remarks section of any of these methods.

Clearly he just doesn't like being wrong, the argument stems from a stubborn ego somewhere out there.

**MavenGradle** - August 13, 2019, 9:03 pm

(Aug 13, 2019 08:39 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Not really necessary - the default overload where the comparison is not specified, matches the comparison used by Contains() (ordinal). The comparison I provided didn't even show the parameters though since string comparers weren't the main point that I was trying to illustrate; only how to check if something is "contained" within a string like the Contains() method does, since he said IndexOf() couldn't solve the same task. He moaned about the detail in my posts so I attempted to keep it as on point as I could, yet he couldn't comprehend the information apparently. If he was ambitious enough to look on MSDN, he would have found this information if he checked the other overloads or read the Remarks section of any of these methods.

Clearly he just doesn't like being wrong, the argument stems from a stubborn ego somewhere out there.

It is really necessary to point this out when using these functions for syntax highlighting in languages such as SQL. However, I do get your point entirely and your reasons for pressing through. I do hope Jelly can learn something from all of this, which is why we're here (developers with 300+ years of experience \*wink wink\*).

Edit: This next part is entirely philosophical and should not be taken as -the- truth for everyone; For developers it is absolutely alright to be wrong. This is the only way a lot of us learn and see our mistakes. The only thing that's hard to break is the pride we have of the quality of our code (or the lack thereof in some cases). I, for instance, like being wrong. This means I get to learn something new and improve my skills further. We might think we know everything, but do we, really?

**0xDEAD10CC** - August 13, 2019, 9:06 pm

(Aug 13, 2019 09:03 PM)MavenGradle Wrote: [\[CITE\]](#) It is really necessary to point this out when using these functions for syntax highlighting in languages such as SQL. However, I do get your point entirely and your reasons for pressing through. I do hope Jelly can learn something from all of this, which is why we're here (developers with 300+ years of experience \*wink wink\*).

Edit: This next part is entirely philosophical and should not be taken as -the- truth for everyone; For developers it is absolutely alright to be wrong. This is the only way a lot of us learn and see our

mistakes. The only thing that's hard to break is the pride we have of the quality of our code (or the lack thereof in some cases). I, for instance, like being wrong. This means I get to learn something new and improve my skills further. We might think we know everything, but do we, really?

My point was never about using these functions for syntax highlighting. In the context of what he wrote, the information I provided should've been sufficient enough. All I had to prove was that you can use `IndexOf()` to achieve what `Contains()` does. Obviously since `IndexOf()` is a fundamental function for text parsing though, it would be included in the solution, but wouldn't be the the solution itself to syntax highlighting or `Contains()` for that matter.

On the topic of this thread -- I've been trying to paraphrase the importance of lexical analysis this whole time. Programmers who write compilers would understand what I'm talking about.

**MavenGradle** - August 13, 2019, 9:15 pm

(Aug 13, 2019 09:06 PM)0xDEAD10CC Wrote: [\[CITE\]](#) My point was never about using these functions for syntax highlighting, so in the context of what he wrote, the information I provided should've been sufficient enough. All I had to prove was that you can use `IndexOf()` to achieve what `Contains()` does.

On the topic of this thread -- I've been trying to paraphrase the importance of lexical analysis this whole time. Programmers who write compilers would understand what I'm talking about.

You did absolutely that. Don't get me wrong, I think communication here is going a bit off track. We both agree that lexical analysis provides the correct answer for how it should be performed. One of the first things a compiler does is lexical analysis, which is why you are referencing how only programmers who write compilers understand what you say, yet I do not write compilers and I understand perfectly. A basic understanding of what is required from a compiler or what a compiler does should be enough.

I am merely adding to the argument on the `Contains` vs `IndexOf` method and using either of those for syntax/code highlighting. We could go on all night regarding these linguistic oddities, so let's keep it at that ? .

**0xDEAD10CC** - August 13, 2019, 9:18 pm

(Aug 13, 2019 09:15 PM)MavenGradle Wrote: [\[CITE\]](#) You did absolutely that. Don't get me wrong, I think communication here is going a bit off track. We both agree that lexical analysis provides the correct answer for how it should be performed. One of the first things a compiler does is lexical analysis, which is why you are referencing how only programmers who write compilers understand what you say, yet I do not write compilers and I understand perfectly.

I am merely adding to the argument on the `Contains` vs `IndexOf` method and using either of those for syntax/code highlighting. We could go on all night regarding these linguistic oddities, so let's keep it at that ? .

I understand then -- OP would truly be better off with a third party library though if he's never done anything similar, unless he's up for the challenge. There's more understanding involved on this topic though than just being able to write code; however, there is lots of material on the subject however.

**MavenGradle** - August 13, 2019, 9:28 pm

(Aug 13, 2019 09:18 PM)0xDEAD10CC Wrote: [\[CITE\]](#) I understand then -- OP would truly be better off with a third party library though if he's never done anything similar, unless he's up for the challenge. There's more understanding involved on this topic though than just being able to write code; however, there is lots of material on the subject however.

Quite the challenge indeed. I feel like this topic should have its own thread where interested and like-minded people could discuss said material. - For example the two AI bots that started talking in their own language. Very fascinating. I cannot help but wonder how the logic behind those bots

works - Talking in a completely new language and listening to and executing commands. An achievement man should be proud of.

**0xDEAD10CC** - August 13, 2019, 9:42 pm

(Aug 13, 2019 09:28 PM)MavenGradle Wrote: [\[CITE\]](#) Quite the challenge indeed. I feel like this topic should have its own thread where interested and like-minded people could discuss said material. - For example the two AI bots that started talking in their own language. Very fascinating. I cannot help but wonder how the logic behind those bots works - Talking in a completely new language and listening to and executing commands. An achievement man should be proud of.

It would've been through a collection of probably very complex ANN's that deal with each layer of the processing to get to custom I/O from the data.

**riu41776** - August 19, 2019, 3:52 am

(Jul 23, 2019 09:54 PM)Jelly Wrote: [\[CITE\]](#) You can do this by using String.Contains and/or parsing in a constant loop

Constant loop not good, especially on main thread which will just stuck the entire app. I would suggest the parsing to be inside the TextChanged event handler of the ui control.

**DNA** - August 19, 2019, 4:01 am

(Aug 13, 2019 08:39 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Not really necessary - the default overload where the comparison is not specified, matches the comparison used by Contains() (ordinal). The comparison I provided didn't even show the parameters though since string comparers weren't the main point that I was trying to illustrate; only how to check if something is "contained" within a string like the Contains() method does, since he said IndexOf() couldn't solve the same task. He moaned about the detail in my posts so I attempted to keep it as on point as I could, yet he couldn't comprehend the information apparently. If he was ambitious enough to look on MSDN, he would have found this information if he checked the other overloads or read the Remarks section of any of these methods.

Clearly he just doesn't like being wrong, the argument stems from a stubborn ego somewhere out there.

You just failed to take in any information I was saying and focused on only arguing something that literally makes milliseconds of difference, I was merely suggesting a way to define it further then using regex and dictionary solutions which I am not particularly a fan of.

I didn't want to stretch a conversation out like this, so I ended it and went back to finishing what I was focusing on at the time.

The sheer fact you tried to belittle me, was enough for me to end the conversation there as it was clearly not going anywhere. I didn't say IndexOf couldn't resolve the issue, I wasn't saying that Contains is a derivative of that function and the argument was moot.

The only argument was that regex was somewhat better and I didn't care much about that argument as I was more inclined to believe you were the one posting that "Tutorial" that I linked.

Anyhow, I thought I'd check back in since someone quoted me.

(Aug 19, 2019 03:52 AM)riu41776 Wrote: [\[CITE\]](#) Constant loop not good, especially on main thread which will just stuck the entire app. I would suggest the parsing to be inside the TextChanged event handler of the ui control.

You would of course have that loop triggered on and off by events, you shouldn't constantly loop without triggers.

**0xDEAD10CC** - August 19, 2019, 1:03 pm

(Aug 19, 2019 04:01 AM)Jelly Wrote: [\[CITE\]](#) You just failed to take in any information I was saying and focused on only arguing something that literally makes milliseconds of difference, I was merely suggesting a way to define it further then using regex and dictionary solutions which I am not particularly a fan of.

I didn't want to stretch a conversation out like this, so I ended it and went back to finishing what I was focusing on at the time.

The sheer fact you tried to belittle me, was enough for me to end the conversation there as it was clearly not going anywhere. I didn't say IndexOf couldn't resolve the issue, I wasn't saying that Contains is a derivative of that function and the argument was moot.

The only argument was that regex was somewhat better and I didn't care much about that argument as I was more inclined to believe you were the one posting that "Tutorial" that I linked.

Anyhow, I thought I'd check back in since someone quoted me.

You don't even know what the solution is here clearly, and as I said before, everything you've said is not without equivocation; simply because you don't know what you're talking about. Now you're suggesting Regex, dictionaries, and Contains() all together (with the constant loop from your previous post)? All I see is ever-changing statements about what you meant the post before that sways in the direction of everything I've been saying from the start against what you initially said... And yet I'm the one who has "failed to take in any information"? ?

You should explain what you're even talking about and show how this is even a valid strategy to begin with before saying that I've failed to take in information. I've ignored 99.9% of it because it is useless information, doesn't make any sense, and won't help OP develop code that achieves this task in a good way. I already know this, so there's no point in entertaining the idea of any of your suggestions here.

"IndexOf couldn't resolve the issue, I wasn't saying that Contains is a derivative of that function": Not even close to what you said before, because you definitely said that "IndexOf wouldn't yield the same results". The post is still in this thread if you've already forgotten. It was another member who mentioned that Contains() was implemented with IndexOf(), not you.

You haven't been on this forum for nearly as long as I have, I've responded within many Regex threads with bad information, and would never post a random thread like that without context (and never have), so why would anybody with a functional brain conclude that it was I who posted that thread?

(Jul 23, 2019 09:54 PM)Jelly Wrote: [\[CITE\]](#) You can do this by using String.Contains and/or parsing in a constant loop

Let's take this and add it to the previous context with the older post of yours I referenced -- I wonder WHY you think loops are bad, you don't understand when to use them or how to use them properly. A constant/continuous loop isn't even close to how it's implemented in the real world (anywhere). Why would an event to trigger a while loop help? If an event triggers the "constant loop" what condition stops the "constant loop"? You don't understand how these things work so you shouldn't be part of the discussion here.

You lack basic thinking skills, and humility here. Prove me wrong, or stop posting random senseless information. Is there anything you post that isn't arbitrary? :/

**devvariety** - August 19, 2019, 1:26 pm

The most mind blowing fact is that this argument has been on-going for over 1 week, crazy!

**0xDEAD10CC** - August 19, 2019, 1:44 pm

(Aug 19, 2019 01:26 PM)devvariety Wrote: [\[CITE\]](#) The most mind blowing fact is that this argument has been on-going for over 1 week, crazy!

This commentary is not necessary. I'm still waiting on people to post some code instead of arbitrary recommendations on how they think these things work.

**DNA** - August 19, 2019, 4:54 pm

(Aug 19, 2019 01:44 PM)0xDEAD10CC Wrote: [\[CITE\]](#) This commentary is not necessary. I'm still waiting on people to post some code instead of arbitrary recommendations on how they think these things work.

That is what you have been doing this entire time!

Hypocrisy is thy name.

When you're worth the time of day, I'll reply again. Clearly you want to reignite this and I'm not interested, and haven't been since you started your tirade.

Not everyone sits on their arse writing novels to feed their superiority complex.

Go write another 20 paragraph long novel that no one will end up reading.

Ironic how you skipped out the part of my statement calling you out for being a troll, isn't it?

I have my doubts about your, you lack any professionalism which makes me believe you don't have the experience you claim to have.

You're too busy bashing on everyone else's integrity to try and feel superior.

We're done here, go feed your ego elsewhere.

(Aug 19, 2019 01:26 PM)devvariety Wrote: [\[CITE\]](#) The most mind blowing fact is that this argument has been on-going for over 1 week, crazy!

He's arguing with himself because I haven't been interested since page 2.

I didn't even read his paragraph he left just above. He isn't worth the time

A week later and he's still arguing with himself about something that makes so little difference it legit doesn't matter ?

**Mogwai** - August 20, 2019, 12:20 am

(Aug 19, 2019 04:54 PM)Jelly Wrote: [\[CITE\]](#) That is what you have been doing this entire time!

Hypocrisy is thy name.

When you're worth the time of day, I'll reply again. Clearly you want to reignite this and I'm not interested, and haven't been since you started your tirade.

Not everyone sits on their arse writing novels to feed their superiority complex.

Go write another 20 paragraph long novel that no one will end up reading.

Ironic how you skipped out the part of my statement calling you out for being a troll, isn't it?

I have my doubts about your, you lack any professionalism which makes me believe you don't have the experience you claim to have.

You're too busy bashing on everyone else's integrity to try and feel superior.

We're done here, go feed your ego elsewhere.

He's arguing with himself because I haven't been interested since page 2.

I didn't even read his paragraph he left just above. He isn't worth the time

A week later and he's still arguing with himself about something that makes so little difference it legit doesn't matter ?

I wouldn't bother replying much more man, I've read this entire thing and this guy is an idiot and clearly trolling you by carrying out an argument you clearly did not want to get involved in.

Let sleeping dogs lie

**Venusaur** - August 20, 2019, 12:30 am

(Aug 19, 2019 01:44 PM)0xDEAD10CC Wrote: [\[CITE\]](#) This commentary is not necessary. I'm still waiting on people to post some code instead of arbitrary recommendations on how they think these things work.

yes it is because it shows how much you are blowing this up for no reason

(Aug 13, 2019 08:39 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Clearly he just doesn't like being wrong, the argument stems from a stubborn ego somewhere out there.

i think you dont like being wrong because you keep carrying on like a skid

**Washi** - August 20, 2019, 3:30 am

There is absolutely no reason to call `String.Contains` prior to a `String.IndexOf` call. `String.IndexOf` obtains the same information as `String.Contains` (whether a substring exists in the given input), and even more (where this substring actually is actually located in the input string). It would be doing twice the work that you actually need to do.

For syntax highlighting, the underlying task is to figure out which characters in the input string form tokens of terminals that exist in the programming language (keywords, brackets, literals, identifiers etc.). This means that we're not interested in just the existence of a terminal, but actually that we need to know the location of it as well. As 0xDEAD10CC correctly pointed out, lexical analysis is the way to go if you want 100% accuracy on which tokens exist where in the given input file. Doing a `String.Contains` is just not going to cut it, and if you're going to use `IndexOf` anyway, then by the previous argument it is useless to call `Contains`.

Now onto the debate on whether `IndexOf` is actually suitable. You can probably use it, but it gets hairy really quickly. Aside from the example given by 0xDEAD10CC (the example that tries to highlight the keyword `int`), syntax highlighting using `String.IndexOf` will also run into problems such as dealing with strings, especially if the language allows escaping characters, such as the following:

PHP Code

```
<?php
```

```
string a = "Hello, world!";
```

```
string b = "Shakespeare's famous quote is: \"To be, or not to be, that is the question.\"";
```

```
string c = "C:\\Users\\Washi\\Desktop\\";
```

If you'd start by looking for the occurrence of two subsequent `"` characters using `String.IndexOf`, it would work for the first example, but you'll run into problems with the second one, as the next quote is actually escaped. You could then maybe check if the previous character was a `"` instead, but then you'll run into more problems in the third example, because now the slash before the last quote is actually escaped instead of the quote itself! You can imagine this continuing to blow up even more

with edge cases, to the point that using regular expressions or a proper lexing algorithm is just so much more easier, less error-prone, and potentially faster (as it's compiled to a minimal finite state machine). In fact, a lot of lexers are actually implemented using a regular expression for each token, combined into one single state machine.

On a side note: For even further advanced syntax highlighting, it is not even enough to just do lexical analysis. Most, if not all programming languages are not regular, meaning you won't be able to capture everything using just regular expressions or finite state automata (that consume one character at a time). Visual Studio uses Roslyn to syntax highlight their code to also distinguish identifiers between variables, methods, classes, interfaces and so forth, and give them a different colour each. To achieve this, you enter the world of parsers and type checkers, which is far more involved than just lexical analysis.

(Aug 19, 2019 04:54 PM)Jelly Wrote: [\[CITE\]](#) I didn't even read his paragraph he left just above. He isn't worth the time

He might not be worth your time, but we as a forum kind of care about the information that is being shared. Whether you like it or not, if someone posts incorrect information, it will be corrected by someone else. As mentioned before, it is totally fine to be wrong at times. Everyone is. However, if you cannot accept being wrong at times, then you might want to reconsider posting in the future.

(Aug 20, 2019 12:20 AM)Mogwai Wrote: [\[CITE\]](#) I wouldn't bother replying much more man, I've read this entire thing and this guy is an idiot and clearly trolling you by carrying out an argument you clearly did not want to get involved in.

Let sleeping dogs lie

(Aug 20, 2019 12:30 AM)Jaws Wrote: [\[CITE\]](#) yes it is because it shows how much you are blowing this up for no reason

i think you dont like being wrong because you keep carrying on like a skid  
That is funny, because 0xDEAD10CC is right. Who's the idiot now? ?  
A for effort though.

**DNA** - August 20, 2019, 4:10 am

(Aug 20, 2019 03:30 AM)Washi Wrote: [\[CITE\]](#) There is absolutely no reason to call `String.Contains` prior to a `String.IndexOf` call. `String.IndexOf` obtains the same information as `String.Contains` (whether a substring exists in the given input), and even more (where this substring actually is actually located in the input string). It would be doing twice the work that you actually need to do.

For syntax highlighting, the underlying task is to figure out which characters in the input string form tokens of terminals that exist in the programming language (keywords, brackets, literals, identifiers etc.). This means that we're not interested in just the existence of a terminal, but actually that we need to know the location of it as well. As 0xDEAD10CC correctly pointed out, lexical analysis is the way to go if you want 100% accuracy on which tokens exist where in the given input file. Doing a `String.Contains` is just not going to cut it, and if you're going to use `IndexOf` anyway, then by the previous argument it is useless to call `Contains`.

Now onto the debate on whether `IndexOf` is actually suitable. You can probably use it, but it gets hairy really quickly. Aside from the example given by 0xDEAD10CC (the example that tries to highlight the keyword `int`), syntax highlighting using `String.IndexOf` will also run into problems such as dealing with strings, especially if the language allows escaping characters, such as the following:

PHP Code  
`<?php  
string a = "Hello, world!";`

```
string b = "Shakespeare's famous quote is: \"To be, or not to be, that is the question.\"";  
string c = "C:\\Users\\Washi\\Desktop\\";
```

If you'd start by looking for the occurrence of two subsequent " characters using `String.IndexOf`, it would work for the first example, but you'll run into problems with the second one, as the next quote is actually escaped. You could then maybe check if the previous character was a "\" instead, but then you'll run into more problems in the third example, because now the slash before the last quote is actually escaped instead of the quote itself! You can imagine this continuing to blow up even more with edge cases, to the point that using regular expressions or a proper lexing algorithm is just so much more easier, less error-prone, and potentially faster (as it's compiled to a minimal finite state machine). In fact, a lot of lexers are actually implemented using a regular expression for each token, combined into one single state machine.

On a side note: For even further advanced syntax highlighting, it is not even enough to just do lexical analysis. Most, if not all programming languages are not regular, meaning you won't be able to capture everything using just regular expressions or finite state automata (that consume one character at a time). Visual Studio uses Roslyn to syntax highlight their code to also distinguish identifiers between variables, methods, classes, interfaces and so forth, and give them a different colour each. To achieve this, you enter the world of parsers and type checkers, which is far more involved than just lexical analysis.

He might not be worth your time, but we as a forum kind of care about the information that is being shared. Whether you like it or not, if someone posts incorrect information, it will be corrected by someone else. As mentioned before, it is totally fine to be wrong at times. Everyone is. However, if you cannot accept being wrong at times, then you might want to reconsider posting in the future.

That is funny, because 0xDEAD10CC is right. Who's the idiot now? ?

A for effort though.

The one who carries on an argument about something so insignificant is the idiot, clearly I wasn't interested in the discussion, hence why I didn't bother putting in any input.

You quoted me with the same input that I don't care to have an argument about.

While you both are arguing about something that makes so little difference that it's measured in milliseconds, I've completed 3 projects.

You legit had confirmed what I said in your above statement, who's the idiot now? ;)

**Washi** - August 20, 2019, 5:06 am

(Aug 20, 2019 04:10 AM)Jelly Wrote: [\[CITE\]](#) The one who carries on an argument about something so insignificant is the idiot, clearly I wasn't interested in the discussion, hence why I didn't bother putting in any input.

You quoted me with the same input that I don't care to have an argument about.

While you both are arguing about something that makes so little difference that it's measured in milliseconds, I've completed 3 projects.

You legit had confirmed what I said in your above statement, who's the idiot now? ;)

First of all, I never called you an idiot. Secondly, spreading incorrect information is more significant than you might think.

I responded because your first argument that you brought to the table was "Regex is probably slow and inefficient in this case", implying you do care for performance.

If I understood you correctly (as the terms you use seem a bit inconsistent at times in your posts), you state that it is more efficient to first check if a line of code might contain some kind of keyword, and if so, then parse that line to make sure. I just made an argument that this is not more efficient, as you're doing at least twice the work that you need (First traverse the string to check if a substring exist, then traverse the string again to verify whether that substring is actually part of a keyword or some other token, both operations can be done in  $O(n)$ ). This would explode even more if you did this for every terminal in the programming language you're targeting.

Now you're basically just saying: "I don't care what you think". I cannot force you to do so, and I won't. But what you should realize is that you're replying to a thread in a programming section. If you don't want anyone providing counter-arguments to your points, then you might have chosen the wrong place to share your thoughts.

Congratulations on finishing three projects though. That's pretty quick in just one week.

**DNA** - August 20, 2019, 5:17 am

(Aug 20, 2019 05:06 AM)Washi Wrote: [\[CITE\]](#) First of all, I never called you an idiot. Secondly, spreading incorrect information is more significant than you might think.

I responded because your first argument that you brought to the table was "Regex is probably slow and inefficient in this case", implying you do care for performance.

If I understood you correctly (as the terms you use seem a bit inconsistent at times in your posts), you state that it is more efficient to first check if a line of code might contain some kind of keyword, and if so, then parse that line to make sure. I just made an argument that this is not more efficient, as you're doing at least twice the work that you need (First traverse the string to check if a substring exist, then traverse the string again to verify whether that substring is actually part of a keyword or some other token, both operations can be done in  $O(n)$ ). This would explode even more if you did this for every terminal in the programming language you're targeting.

Now you're basically just saying: "I don't care what you think". I cannot force you to do so, and I won't. But what you should realize is that you're replying to a thread in a programming section. If you don't want anyone providing counter-arguments to your points, then you might have chosen the wrong place to share your thoughts.

Congratulations on finishing three projects though. That's pretty quick in just one week.

I understand this, which is why I suggested it only for minimalist langs. You're right, I do care for performance, but I just don't care enough to argue the point for it.

Yes It's not more efficient (and as you say, blows up fast) but it is more effective imho and as you say like I said, you don't simply rely on an IndexOf/Contains statement to determine the result, it requires refinement.

In fact, it's not that I don't care to discuss what's being said, it's that I don't care to discuss it with someone who is being derogatory, so I may have misinterpreted what you said.

This is what I mean, I would rather finish what I need to do rather than discuss semantics on something as silly as a string function.

Got enough on my mind, hence why I didn't really bother with this topic

MavenGradle - August 20, 2019, 9:56 am

(Aug 20, 2019 12:20 AM)Mogwai Wrote: [\[CITE\]](#) I wouldn't bother replying much more man, I've read this entire thing and this guy is an idiot and clearly trolling you by carrying out an argument you clearly did not want to get involved in.

Let sleeping dogs lie

Excuse me but 0xDEAD10CC is offering solid and sound advice. He is by no means an idiot. A word which is quite clearly used here in this context to provoke the man. This entire name calling thing should STOP. Both from you, 0xDEAD10CC and everyone else. I apologize if I occasionally sprout out similar nonsense of curse words but they are entirely not necessary in a world where code is nothing but logical thinking and applying the proper principles and using the right functions for the job at hand. As said before, REGEX is not the right way to do things but very much a way to get to where you want to go in a feasible and economically viable way.

(Aug 20, 2019 12:30 AM)Jaws Wrote: [\[CITE\]](#) yes it is because it shows how much you are blowing this up for no reason

i think you dont like being wrong because you keep carrying on like a skid

Wrong? Excuse me once again but like I said above, 0xDEAD10CC has not said anything WRONG in terms of coding advice except for some minor things here and there to which we can all agree, but his point still stands (come at me 0x. Just messing with you). You should read the entire thread in order to get the right idea.

Now I do agree that 0xDEAD10CC could perhaps improve the way he handles his way of communicating his advice and tips but that does not mean the man is wrong. Granted I am not the best judge of character due to my short lifespan on this forum but 0xDEAD10CC is someone, who like me, likes to go into details which others might overlook. He is simply going through said details like a good nitpicker would and nitpicking on code is in my opinion a thing you should very much value because optimizations in any way is always good practice. Always.

Both of you are providing nothing to this argument except picking sides which is an emotional thing that is simply not about programming. Emotions have nothing to with programming, this is a field of logic, not imaginary sides to whatever conflict it is you think is going on here. I'm sure both 0xDEAD10CC and Jelly would be perfectly capable of drinking a beer together if they so choose, at least if they can get over their name calling differences. Can we please move on with this conversation in an applicable, professional and informative way? This question is very much for everyone here.