

Peer-review of assignment 5 for *INF3331-jostbr*

Reviewer 1, chriswko, chriswko@student.matnat.uio.no

Reviewer 2, jorundfj, jorundfj@student.matnat.uio.no

Reviewer 3, danielosen, danieleo@student.matnat.uio.no

November 13, 2016

1 Review

python 3.4.2 Red hat 4.4.7 (ifi computer)

General feedback

Would have liked to see a better handling of overlap cases, and a more compact regex. The assignment is solved as the assignment specified. Good use of docstrings and we appreciate the added ReadMe file and the example files to make it easy to test your code.

Assignment 5.1: Syntax highlighting

The highlighter is working as expected and it's easy to figure out how the input arguments are supposed to look with the help of the help command.

The code is very well documented which makes it easy to read and understand. Good variable names and overall structure, not much to complain about here. Would have liked a way to deal with overlapping as strings gets canceled and doesn't finish highlighting if there's another regex in the middle of it, same happens with comments. To handle this you could save the indexes of the regexes with the help of `re.finditer` and use these to insert the formatting.

Assignment 5.2: Python syntax

The regex seems a bit unnecessarily complicated, you don't need to add 3 different regexes for ifs for example (such corner cases are usually solved by making a more robust algorithm for the highlighter). Other than that your regexes seem to work as intended and even if you added some of them together there are enough to fill the quota. There are no specific colors for arithmetic operators `" + / - * = "`, and no numbers (integers, floats or scientific), and there are several cases where your regex expressions do not appear to produce expected behaviour such as:

- coloring `"if"` in `"if("` (appear to be assuming there is whitespace right after `f` character)
- coloring `"except"` in `"except:"` (appear to be assuming there is whitespace right after `t` character)
- coloring certain cases where one is using assignments with strings (overlapping as mentioned above)
- coloring multiple assignments (tuples) `" var1,var2 = "string1","string2" "`

Assignment 5.3: Syntax for your favorite language

C

There are a lot of errors in the highlighting here, but it looks like the fault lies mainly with the highlighter itself and not the regexes you made. Overlap is handled poorly and it creates a lot of errors in the highlighting.

Like before, it would be nice to have regex for numbers, logical- and arithmetic operators. However, more important are keywords in C such as types (`int, float, enum, void, char, vector, string`) and type modifiers (`long, double, short`), access modifiers (`private, public, const, virtual`) and reference types (`& address, * pointers`) as these are fairly important in languages such as C.

Understanding some of your regular expressions is difficult, to say the least (though this is somewhat expected because regex is not really suitable for this task), but testing them on sites such as regex101.com might be useful to you (if

you haven't already done so). Since there are so many cases with errors, let's only look at one (with very simple code).

```
int main(){  
    return 0;  
}
```

Your regular expression for catching function definitions like this simply does not work. Presumably you would want to capture the "main" part. To do something like this you might try to capture lines of the following grammatical structure:

"type modifier/access" "type" "scope/membership" "functionname" "(args)" "{".

If you take advantage of group capturing, this will be relatively straightforward (you ignore anything but the last group, which you design to contain the function name). What has to be considered is of course that this expression is fairly general, and a lot of times several of the grammatical components will be missing. In this case, taking advantage of the syntax rules of C is very beneficial, because it gives you exactly what is valid to write in these cases. It might be a lot of work to consider all the possible combinations though. The general idea is that you want to make sure at least a "type" is there followed somewhere along the line with "functionname" "(" "anything" ")" and finally "{", where you require "type" so you don't accidentally color "for(...)" expressions.

Assignment 5.4: Syntax for your second favorite language

fortran

The regex seems to work fine and you highlight a lot of the syntax. We have never used fortran so we're not quite sure what's important to highlight, but the things you added work so we're satisfied.

Assignment 5.5: superdiff

Well documented and structured code. It works as expected and we liked that it writes out with colors directly. Would have liked a different color choice for non modified lines as it's hard to read on a white terminal. Easy to understand variable names and not an overly complicated code.

Assignment 5.6: Coloring diff

Everything works here, not much to say.