

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

Neringa Altanaite, neringaa, neringaa@uio.no
Nicolai August Hagen, nicolhag, nicolhag@uio.no
Vegard Liloe Svendsen, vegardls, vegardls@uio.no

November 10, 2016

1 Review - *to be filled out*

Python version and operating system used:

Mac OS X with Python 3.5.2

General feedback

In general we find your solution good. You have good documentation of your solution, and use good variable and function names. Some parts of your solution seem a bit complicated, but works as expected.

Some of the lines in your programs, especially docstrings are too long, and should preferably be split into multiple lines. It would also have been nice if you had supplied a README-file to help testing your code.

Assignment 5.1: Syntax highlighting

Very good and thorough documentation of your solution, and nice variable and function names. The docstrings are especially useful. Nice! In your program, it would have been nice to supply a -h option for command line usage for help on how to run your program.

The part where you extract the content of the regex-files and theme-files is a bit more complicated than necessary. For example, your function `getColorDictionary()` could have been re-written as:

```
1 syntax_file = (open(sys.argv[1])).read()
2 syntax_values = re.findall(r'(.+?):', syntax_file)
3 syntax_keys = re.findall(r'(: (.+))', syntax_file)
4 syntax = dict(zip(syntax_keys, syntax_values))
```

Here, the built-in `re.findall()`-function, along with group extractions in the regex makes a list of tuples - which again can be translated into a python dictionary. The same logic can also be applied to the regex-file.

In your solution, you do not handle overlapping of coloring - which makes it necessary for you to check that a potential highlighting is not inside another potential highlighting (e.g., in the `python.syntax`-file where you check that `import` do not come after a `#` for comments). Because of this, some of the regexes will not always be correct. For example, consider adding the following comment at line NO. 9 in your program:

```
# set to True if value is found
```

Programming parts not answered:

- Overlapping coloring

Assignment 5.2: Python syntax

You have colored 10 different pieces of Python code - good.

Nice, most of the regexes work as expected! You correctly color the `import`-statements, but here I also miss the three other possible "spellings" of `imports`:

```
import matplotlib.pyplot as plt
from matplotlib import colors
from matplotlib import colors as cs
```

This could have been corrected by adding **from** and **as** inside the regexes, also taking into account that they actually occur on the import-lines:

```
((?:^|\n)import\w*|(?:^|\n)from\w*|(?:^|(?:\w*)import(?:\w*))|as(?:= \w*\n))
```

Another problem with your solution is that "variables" will not always be colored correctly. Consider the code below:

```
grid[grid == 0] = max_escape_time
```

To support this, you could have used negative lookahead and/or negative lookbehind, checking that the character in from or in behind is not =. We find the following article useful: [lookarounds](#).

Assignment 5.3: Syntax for your favorite language

You have colored 10 different pieces of Java code - good.

When coloring different datatypes, we're missing the Java datatypes **short**, **long**, **float**, **char**, **byte**.

Also, your regex for **class** will also highlight inside e.g. variable-name. Consider adding the following line to your program:

```
private double classOfStudents = 0;
```

This can be handled by checking that **class** is at the beginning of a line, possibly "prefixed" by private or public.

Assignment 5.4: Syntax for your second favorite language

Not implemented.

Assignment 5.5: superdiff

Very well documented with comments and docstrings!

Nice, your program works as expected when first file is (A, B, C, D, E, F) and second file is (B, C, D, E, F, G). Very good! Some parts of the program seem a bit complicated, but works as expected when run. We would recommend avoiding the use of **while True**-loops and keeping the use of the **break**-keyword to a bare minimum, to increase readability of the code.

In your program, we are missing handling of erroneous input (e.g., missing files given to the program).

Assignment 5.6: Coloring diff

The `diff.syntax`-file works as expected - very good!