

# Peer-review of INF3331 assignment 5 for *yosiefht*

Jonathan P. Stang, jonathps, jonathps@ulrik.uio.no  
Thach Khoi Pham, thachkp, thachkp@ifi.uio.no  
Thor K. Høgås, tmhogaas, tmhogaas@ifi.uio.no

November 13, 2016

## 1 Review

I've used a Linux workstation at Ifi (Institutt for Informatikk) to review your assignment. I compiled your code using Python 3.5.2 with Anaconda 4.2.0 installed. The main reviewer was Jonathan Stang.

### General feedback

You have solved the exercises as expected. The main issue I personally would've liked to see, was a well written Readme.txt. It was not clear which programming language you were choosing as favorite languages. This could've been explained in the Readme.txt. Also, the Readme.txt would've been much more helpful if it contained precise command line arguments to run all the different exercises.

### Assignment 5.1: Syntax highlighting

First off, your code works as expected. It prints out the desired colored text to the terminal, according to the syntax and theme file.

- Comments on code structure in the script highlighter.py. The first two comments are general, and should apply to all your future python scripts.

- First off, I would've liked to see you test whether the python interpreter runs this script as the main program or as a module in another program. The way to implement this would be:

```
1 if __name__ == "__main__": # Add this line before calling your function(s).
2     if len(sys.argv) == 4:
3         syntaxFile = sys.argv[1]
4         themeFile = sys.argv[2]
5         sourceFile_to_colour = sys.argv[3]
6         readFile(syntaxFile, themeFile, sourceFile_to_colour)
7     else:
8         sys.stdout.write("provide correct file inputs:" + '\n')
9         sys.stdout.write("python3 highlighter.py syntaxfile themefile sourcefile_to_colour")
```

- Secondly, I would've liked a Python Docstring. This is standard procedure in programming languages such as Python, Java, and C. It is basically a section that explains each function and/or class. In Python, you can write it above and outside or below and inside the function declaration. In my example below it is placed below and inside the function declaration, hence it is indented:

```
1 def foo_bar(foo, bar):
2     """ These first lines explains what your function foo_bar() does in plain language.
3     A reader should be able to understand what this function is doing by reading this docstring.
4
5     Args:
6         foo: The first parameter.
7         bar: The second parameter.
8
9     Returns:
10         The return value. string, array, double array, integer, ...
11
12     """
```

- Your function readFile() actually does two / three different procedures. Two are quite similar, but the third is not. Try to separate different tasks into different functions. The two similar procedures are the reading of the theme file and the syntax file. This could've been inside one function. Next, the reading and printout of the file\_to\_color could've been in a different function, called printFile(). I understand you make two dictionaries theme and syntax. Your code structure could therefore have the form:

```

1 def readFile(syntaxFile, themeFile, sourceFile):
2     """ Docstring explaining this function.
3     """
4
5     # Making of dictionaries syntax_dict and theme_dict
6     ...
7     # Call to printFile()
8     printFile(syntax_dict, theme_dict, sourceFile)
9
10 def printFile(syntax_dict, theme_dict, sourceFile):
11     """ Docstring explaining this function.
12     """
13     # Reading of sourceFile
14     ...
15     # Printing of sourceFile with colorings.
16     ...

```

## Assignment 5.2: Python syntax

The second and third regex commands in `python.syntax` works as expected. The first, however, which is the one on 'comments', only works on inline comments, not on docstrings. Docstrings can be able to be several lines long, which your regex does not take into account.

- Solution to docstring regex:

```

1 # For docstrings, instead of
2 """.*(?:$|\n)|''.*(?:$|\n)

```

```

1 # use
2 '{3}(\.|\n)*?'{3}|" {3}(\.|\n)*?'{3}

```

- I would say the regex line 'standard' cover 5-6 different python syntaxes. These could've been implemented separately, to give more of a color palette to your printout.
- There is actually a small error inside the line

```

1 "\b(def|for|while|else|from|import|if|elif|)\b": standard

```

which is the the character '|' at the end. The code runs when this is removed.

## Assignment 5.3: Syntax for your favorite language

The code runs. My problem is: Which language is this? I suppose it is Java, but you should comment on these things in the `Readme.txt`. As in Assignment 5.2 1, you have not implemented more than one regex for the whole of the Java syntaxes. This should've been done.

## Assignment 5.4: Syntax for your second favorite language

As in Assignment 5.3, I do not know which language you are talking about. The syntax runs, but seems limited, just as in 5.3.

## Assignment 5.5: superdiff

Congratulations on writing a working diff-algorithm. The standard algorithm for this is Longest Common Subsequence (video explanation). Which it doesn't look like you've been using. Unlike `highlighther.py`, you do actually have a Python Docstring for your function here. Good! But like `highlighther.py`, you fail to test whether the script is run as the main program or as a module in another program. This should be implemented.

Also, The last line lacks a newline character '\n'. This would make print to terminal more pretty.

## Assignment 5.6: Coloring diff

Aside from the fact you color purple instead of red and blue instead of green, everything works as expected.