Task documentation SYN: Syntax highlighting in PHP 5 for IPP 2016/2017
Name and surname: Martin Ivančo
Login: xivanc03

Task overview

The task was to create a PHP script that searches for regular expressions in text and highlights them using html markup tags.

Implementation parts

Argument processing

The script begins with checking arguments. If the arguments are valid, global argument variables are set accordingly. In the case of invalid arguments, or use of a forbidden combination of them, the script exits with an error. For this I wrote an auxiliary function called `argCheck().`

Loading input

Provided that an input file has been specified, it is opened and loaded into variable `$input`. If script fails to open this file, it exits with an error. If the user hasn't provided any input file, the script gets input by reading a line from standard input.

Format file processing

First, the script checks if a format file name has been given, and tries to open it. If no file has been specified, or the script is unable to open it, no formatting (except if the argument `--br` has been used – see section "Adding line breaks") will be done on the input. However, this is not considered as an error, and the script continues. Given that the format file has been successfully opened, it is then read line by line, while each line is processed with function `parseFormatEntry()`. This function divides the line to a search word (a regular expression to be highlighted) and a list of formatting commands for this search word. The search word is modified, so that it matches the PCRE standard using makeSearchWord(). It then executes the search phase.

Search phase

In this part, begin and end marks (HTML tags) are created according to a formatting command using `makeMarks()` function. After that, `findAll()` is called, which finds all the instances of the search word in the input, and adds the marks to a separate array `$positions` according to the position and length of each instance. This whole process is done for each formatting command in the format file.

Adding line breaks

If the `--br` argument was used, the script calls the `addLineBreaks()` function, which is a simplified version of `findAll()` function, searching for all `'\n'` symbols in file, and adding `< /br>` marks to the `$positions` array accordingly.

Highlighting

When the script has gathered all the formatting marks in the `$positions` array, it calls the function `highlight()`, which generates output from `$input` and `$positions`. It is saved in the `$output` variable.

Exporting output

If an output file has been specified, it is opened and the `$output` is written into it. If the script fails to open the file, it exits with an error. When there has not been an output file specified, the `$output` is printed to standard output.

Conclusion

Although this was the first time I was working with PHP, it didn't take me long to understand the basics of this language. Working on this project I got to know some of the useful standard PHP functions, such as `preg_match_all()` for finding regular expressions. Overall, I found this project very useful.