# Laboratory Tasks #2

## Exercise 1 - Prepare your environment.

I. Before we start, you need to have your environment configured.
   - A. Create new git repository on your GitHub account,
   - B. Clone that repository in htdocs directory
   - C. Add this pdf to repository and push into repository with „Task description" message.

## Exercise 2 - Simple function.

I. Create function that will print message „hello functions"

II. Execute your function

III. Commit and push your code to github repository

## Exercise 3 - Arguments to simple function.

I. Add argument „text" to your function.

II. Instead „hello functions" print text function argument

III. Execute your function with parameter value „param3t3r"

IV. Commit your changes and push them to github repository
   - A. Your changes will be visible in history, so don't skip that part!

## Exercise 4 - Repeat function operation.

I. Create repeatContent function, for execute your first function specific number of times
   - A. Add $count argument, that will execute your first function $count times
   - B. Make that argument optional, by giving them 1 as a default

II. Make a loop in repeatContent that will repeat your first function $count times

III. Execute your repeatContent function without parameter, and with 5

IV. Commit your changes and push them to github repository
   - A. Your changes will be visible in history, so don't skip that part!

## Exercise 5 - Change your branch.

I. In your repository crate branch called „calculator" and checkout them.

II. Create new file called calculator.php

III. Push your branch to remote repository
   - A. Your remote repository will in that moment not contains that branch, so you need to publish it

## Exercise 6 - Write basic math functions.

I. In your new file calculator.php create functions, that will have 2 parameters and return following values
   - A. Sum of 2 arguments

   B. Difference of 2 arguments
   C. Multiplication of 2 arguments
   D. Dividing of 2 arguments

II. Execute those function with some numbers, and print them on the screen
   A. You should not print it inside function, function need to return specific value and not use echo.

III. Print on the screen summary of execution all of those 4 functions with parameters 10 and 2

IV. Commit your changes and push them to github repository
   A. Your changes will be visible in history, so don’t skip that part!

# Exercise 7 - Create calculator.

I. Create function calculator, that will be responsible for calculate given arguments.
   A. num1 - first number
   B. operator - mathematical operator like + - / *
   C. num2 - second number

II. Check if given operator is valid - if not return 0

III. Print executed function with following arguments
   A. 1, "+", 2 -> should print 3
   B. 1, "-", 2 -> should print -1
   C. 1, "*", 2 -> should print 2
   D. 1, "/", 2 -> should print 0,5

IV. Commit your changes and push them to github repository
   A. Your changes will be visible in history, so don’t skip that part!

# Exercise 8 - Refactor your calculator, with multi params.

I. Now from your current branch, create new one called „super-calculator” and push it into remote repository like in Exercise 5

II. Change your functions responsible for calculations for a function that may have multiply parameters (spread operator …)
   A. You don’t have to refactor your calculator function, since it has just 2 parameters - that should still works

III. Commit your changes and push them to your new branch to remote repository
   A. Your remote repository will in that moment not contains that branch, so you need to publish it

# Exercise 9 - Create text input calculator.

I. Change your calculator function, that will be responsible now for executing user given value as a text.
   A. User can input string like „1+2*3” and it should execute your function from left to right (without math precedence).
   B. Using that input use your functions to execute input. For given example we should execute following functions:
      1. summary(1, 2) = 3
      2. multiply(3, 3) = summary(1, 2) * 3 = 9

C. TIP: to achieve that you should use function recurrence

II. Commit your changes and push them to github repository

A. Your changes will be visible in history, so don’t skip that part!

# Exercise 10 - Create pull request.

I. After finish you task create pull request from brach „super-calculator” to your main branch, and send me a link to that pull request as a answer