Lab 2 - functions

Practice to write functions.

1. Basic functions

Practice on basic functions.

Exercise 1.1 (1 points)

Create a function `sumRangeNumbers()` that returns the sum of all numbers between two chosen numbers. The function should take two arguments, one representing the lowest boundary and one that represents the highest boundary. For example, the arguments 10 and 20 should return the sum of 10+11+12+13…+20.

Test it using the values `22 and 91`, answer with the result.

Exercise 1.2 (1 points)

Create a function called `fruitColor()` that takes one argument called `fruit` and returns the color of the fruit.

The function should be aware of the following fruits (`banana, apple, kiwi, plum`) with respective color (`yellow, green, green, red`).

Try it out using the fruit `plum` and answer with the result.

Exercise 1.3 (1 points)

Create a function `printRange()` that takes two arguments `rangeStart` and `rangeStop` and returns a string with all numbers comma-separated in the range.

Try it using the arguments `25 and 46`.

Answer with the result.

Exercise 1.4 (1 points)

Create a function `printRangeReversed()` that takes two arguments `rangeStart` and `rangeStop` and returns a string with all numbers comma-separated in the range.

Try it using the arguments `46 and 25`.

Answer with the result.

Exercise 1.5 (1 points)

Create a function `printAnyRange()` that takes two arguments `rangeStart` and `rangeStop` and returns a string with all numbers comma-separated in the range.

If `rangeStart` is smaller than `rangeStop` then call the function `printRange()`.

If `rangeStart` is greater than `rangeStop` the call the function `printRangeReversed()`.

Try it using the arguments `25 and 46` (both ways).

Answer with the result using arguments 25 and 46.

Exercise 1.6 (1 points)

Create a function called `stringRepeat()` that returns a string a specific number of times. The function should take two arguments, one string and one number: `grey` and `12`. The number represents the number of times the string should be added to a string.

Test the function and answer with the result.

Exercise 1.7 (1 points)

Create a function `inRange()` that takes three arguments, `rangeStart`, `rangeStop` and `value`.

The function should return boolean `true` if value is greater than rangeStart and less than rangeStop. Otherwise it should return boolean `false`.

Try it out using the range `131 - 547` and the value `434`.

Answer with the result.

Exercise 1.8 (1 points)

Try out the function `inRange()` using the range `131 - 547` and the value `636`.

Answer with the result.

Exercise 1.9 (1 points)

Create a function called `degreesToRadians()` that should take one argument, a degree value. The function should convert the value to radians and return the result with max 4 decimals.

Test your function with the value `32` and answer with the result.

Exercise 1.10 (1 points)

Create a function called `fizzBuzz()` that takes two arguments `start` and `stop` and returns a comma-separated string.

The arguments represents the starting point and stop point of the game `Fizz Buzz` (http://en.wikipedia.org/wiki/Fizz_buzz). The function should run from start to stop and add `Fizz`, `Buzz` or both to your result-variable at appropriate numbers.

Each entry to your result should be comma-separated. If `stop` is equal or lower than `start`, the function should return an appropriate error message.

Test the function using `start=1 and stop=30`.

2. Extra assignments

These questions are worth 3 points each. Solve them for extra points. In this section, you could re-use your code from lab 1 in exercise 2.1 and 2.2.

Exercise 2.1 (3 points)

Create a function called `printSum()` that should take two variables, the sum of the players hand and the sum of the dealers hand.

Your hand should be three cards with the values: `4, 10 and 3`.
The dealers hand should be three card with the values: `3, 6, 11`.
The function should return the sum and the player: `Player: 17, Dealer: 20`.

Test your function with the given values and answer with the result.

Exercise 2.2 (3 points)

Create a function called `printResult()` that should take two variables, the sum of the players hand and the sum of the dealers hand.

Players hand should be three cards with the values: `4, 10 and 3`. The dealers hand should be three card with the values: `3, 6, 11`.

This time you should include the check from lab 2 where you find out the boundaries of the player and the dealer.
Player hand = 21 (black jack).
Player hand less than 21 (safe).
Player hand larger than 21 (busted).
Dealer hand less than 17 (safe).
Dealer hand larger or equal to 17 and less than 21 (stop).
Dealer hand = 21 (black jack).
Delaer hand larger than 21 (busted).

Return a string in the format: `Player: safe, Dealer: busted`.

Test your function with the given values and answer with the result.

Exercise 2.3 (3 points)

Create a function called `calculateInterest()` that returns the money you have after x years of interest, given three arguments: `745, 31 and 5`. First argument represents the start money, the second argument represents the number of years your money produces interest. The third argument is the interest rate in percent (%).

Test your function and answer with the result with a maximum of 4 decimals.