Homework 1:
Python Basics, DocTests

Benjamin Roth, Marina Sedinkina
Symbolische Programmiersprache

Due: Thursday October 26, 2017, 16:00

In this exercise you will:

- Review some basic Python functionality
- Get some hands-on experience using the python `doctest` framework

Exercise 1: Setting up the Git project

In order to have access to the Git project with the exercise code, and to be able to submit your solution, you need to do the following steps (ask the tutors if any of the steps is unclear to you):

1. Make sure you have a Gitlab account for `gitlab.cip.ifi.lmu.de`
2. Form teams of 2 or 3 students (4 students are not allowed).
3. Use the web form (which you can find on `sp1718.github.io`) to provide the following information for your group:
   - Gitlab id, name, and matriculation number of each team member
   - Your team name will be the concatenation of your Gitlab ids.
4. We will then create a project in Gitlab for you, that will contain the code to the exercises. Submit your solution by pushing to this project.
5. **Please do not create separate files or folders to submit your solution. Instead, change the files we provided.**
6. **Important: do NOT change the tests themselves, implement the missing functionality instead. Changing the tests will result in your exercise sheet scored with 0 points.**

Exercise 2: Python Basics

Exercise 2.1: Doctests

Use the doctest module to test your implementation of the functions in the module `hw01_basics.basics`.

Run your tests with (this assumes that you are in the `src/` directory of your repository):
`python3 -m doctest -v hw01_basics/basics.py`

Exercise 2.2: Adding Functionality [16 points]

For each function, replace the `pass` statement, so that the function works properly as described below and indicated by the doctests.

- `hello_world()`: Print the string 'Hello, world!'. Use the Python3 Syntax for `print`.
- `expon(x, y)`: Return the value of x to the power of y.
- `dividable(x,y)`: return True or False wether x is dividable by y. **Hint:** You can check, whether the remainder of the division of x by y is 0. You can get the remainder by the so-called **modulo operator**.
- `hello(name)`: Print *"Hello, "* followed by the content of `name`.
- `wordlength(w)`: Return the length of the string w. You can use built-in functions.
- `caps(w)`: Return w, but IN ALL CAPS.
- `substring(v, w)`: Return True or False whether v is a substring of w.
- `thirdElem(liste)`: Return the third element of a list.
- `lastElem(liste)`: Return the last element of a list.
- `firstHalf(liste)`: Return the first Half of liste. For uneven-lengthed lists, return the smaller part. **Hint:** Use Python3 integer division.
- `concatenate(liste1, liste2)`: Return a list that contains all elements of liste1 followed by all elements of liste2.
- `isNegative(k)`: Return True if k is smaller than 0 and False otherwise.
- `printElements(liste)`: Print each element in liste. You can use a for-loop for this problem.
- `countToZero(k)`: Print out the numbers counting from k to 0, excluding 0. If k is negative, count 'up' to 0, excluding 0. You can use a while-loop for this problem.

- `noVowels(w)`: Return True or False whether w contains no vowels. **Hint:** If you know regular expressions, use them. Otherwise, you can use a loop and the membership operator (`in`) for his problem.
- `umlautsAndPunct(w)`: Return True or False whether w contains umlauts (äöü) and also a punctuation mark (.!?) at the end.