LDSL
Linguistic Data Science Lab

Topic: Comments and Control
Due date: Sunday, Oct 30 (6 pm CET)
Deliverable: One .py-file uploaded to Moodle

1. You should comment your code. See the slides for details. Uncommented code will be ignored.
2. Hand in a program that does not throw any error messages when we hit Run. If your code does not do everything you want it to do, please place `'''` before and after the parts that don't work and describe the problem within your file (this marks the content between as one large comment).
3. Do not leave output you added to check something in the final version. You can comment it out or delete it, but there should not be outputs not needed for the program.

Task 1 - Understanding Code (15)

Your first task is about understanding a program where all the rules we told you about have been ignored - naming conventions as well as guidelines on comments.

Read the code, try to understand it and then explain what is happening in about 3-10 sentences. Mention the input the program receives, the processing and the output. Describe how it works in general and also on a "step by step" level.

Put this explanation at the top of your program, with ''' before and after the text. This example shows your the format for a less complicated program:

```
'''
Solution Task 1:

The program works as a digital dice for up to four people to play with. It receives specific keyword-commands
and outputs steps in the play as text.

They can choose what dice(s) to play with and then start a random numbers generator by typing "throw" and
their player number. The program also adds up all results and responds to the prompt "results?" instead
of "throw" with the ranked output of the player's numbers and their sums from high to low.
'''
```

Tip: This task becomes much, much easier if you go through your code and comment each line. You can also rename the variables (use search and replace) once you found out what they are supposed to be.

Task 2 - Controlling the Triangle (20)

Solve the following task remembering to comment your code! Read all of the task before beginning to program

Ask the user for a height between 1 and 20 as input. If the input is a number between 1 and 20, proceed, if not, print an error message and ask again for a number - this should run until you receive a number as input. Save the result in a variable "height". Print out a symmetric triangle made out of *-symbols with that height. (for example 8 lines high for a height of 8). If the value is even, the base of the triangle should be at the bottom. If the value is odd, the tip should be at the bottom.

You need to check if the input is a number. Not knowing a quick way off the top of my head, I google something like "python check if input is int" - which is not even correct (the input is always string!) but other humans thankfully think similarly and Google translates between us. So I found this website: `https://bobbyhadz.com/blog/python-check-if-input-is-integer`.

To solve the task, please read through that website and pick a solution that you can use. (There is at least one that you can already understand!) Do not use anything else external, and comment the part of your code where you copy from that website or use its content with this as a source!

Task 3 Average numbers of letters per words - again! - (15)

Rewrite last week's program with improvements. This week, you can check each word (or even character?) to improve on the result from a linguistic point of view.

Add **three** improvements based on Linguistic criteria compared to simply counting words by spaces and characters by length of the list. For each, explain the improvement.