

Programming Winter 2023

Exercises

Number 03, Submission Deadline: November 8, 2pm, 2023

1 Conditions and comparisons

Being able to correctly manipulate condition and comparison operators is very important for executing useful and correct loops. Let's have a short recap:

- The `if` statement may be combined with operators such as equality (`==`), greater than (`>=`), smaller than (`<=`) and not equal (`!=`) to obtain comparison statements that act as conditions and are evaluated as Boolean values;
- In turn, we can combine conditions using the keywords `and` and `or`;
- Conditions are evaluated as Boolean values, thus can either be `True` or `False`;
- The syntax of the `if` statement is (note the **colon**): `if condition : expression`;
- When we make use of the `else` statement, we are referring to all the cases in which the previous `if` statement evaluates to `False`, for instance: if we want to compare someone's age (`age < 5`), the `else` statement will evaluate when `age >= 5`;
- If we want to evaluate multiple cases, we can make use of the `elif` clause (where `elif` is short for "else if").

1. Use an `if-{elif}*-else` clause to evaluate if a variable is `<0`, `>25` or if the length to its type conversion to string is `==2`. Provide an output for each case using string formatting to insert the value of your variable. (2 P)

2 For loops

1. Create a `for` loop that sums numbers from 17 to 113 (**included**) using the `range()` function; (1 P)
2. Create a `for` loop that outputs the numbers from 1 to 10 backwards; (2 P)
3. Create a `for` loop that counts how many odd numbers are up to 15 (**included**); (2 P)
4. Create a `for` loop that counts how many vowels are in the word "programming" using `continue`; (2 P)
5. Use a `for` loop to compute the *arithmetic mean* of the following list of numbers: (2 P)
[87, 98, 95, 9, 80, 70, 1, 43, 92, 23]

3 While loops

1. Create a `while` loop that sums numbers up to 113 (**included**); (1 P)
2. Create a `while` loop that outputs the number of times that a number can be divided by 2 before it becomes less or equal than 5. Use the quotient operator first and then do the same but with the floored quotient operator; (2 P)
3. Python provides a module called `random` for generating pseudo-randomized numbers. (3 P)
Use the `random()` function of this module to sample pseudo-random floating point numbers from the interval $[0, 1)$. Use a `while`-loop to count the number of samples needed to receive a pseudo-random number that is smaller than a given threshold value a , e.g., say $a = 0.1$.

Important:

Please submit your solution as (adequately commented) Python file or Jupyter Notebook.