

Exercises: Conditional Statements and loops

Exercise 1:

Write a Python program to construct the following pattern, using a nested for loop.
(red: output)

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Exercise 2:

Write a Python program that accepts a string and calculates the number of digits and letters. (green: input, red = output)

Sample Data (input) : Python 3.2

Expected Output :

Letters 6

Digits 2

Hints: isdigit(), isalpha() (Book page 188 ff)

'.' or ',' or ' ' (=Spaces) and so on are not letters!

Exercise 3:

Write a Python program to count the number of even and odd numbers from a series of numbers. (green: sample for 'input', red: output for this sample)

Sample numbers :

```
numbers = (11, 12, 13, 14, 15, 16, 17, 18, 19)
```

Expected Output:

Number of even numbers : 4

Number of odd numbers : 5

Exercise 4:

Write a Python program which takes two digits m (row) and n (column) as **input** and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be $i*j$. (**green**: input, **red**: output)

Note :

$i = 0, 1, \dots, m-1$

$j = 0, 1, \dots, n-1$

Input Test Data - Rows = **3**

Input Test Data - Columns = **4**

Expected Result: **[[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]**

Exercise 5:

Write a Python program to find numbers between 200 and 500 (both included) where each digit of a number is an even number. The numbers obtained should be printed in a comma-separated sequence.

Result (**red**: output):

200,202,204,206,208,220,222,224,226,228,240,242,244,246,248,260,262,264,266,268,280,282,284,286,288,400,402,404,406,408,420,422,424,426,428,440,442,444,446,448,460,462,464,466,468,480,482,484,486,488

Exercise 6:

Write a Python program to guess a number between 1 to 100.

First: The program "fetches" a random number from a **self-defined function** ...

then... User is prompted to enter a guess. If the user guesses wrong then the user gets a message "to big" or "to small" on the console prompt and the prompt (user input) appears again until the guess is correct. On a successful guess, user will get a **"Well guessed!"** message, and the program will exit.

Extension of Exercise 6:

The number of trials should be prompted: **"Well done - you have tried it 4 times!"**

Exercise 7:

Suppose we wish to draw a **Christmas tree**, and its height (without tree trunk¹) is provided by the user. (green: input)

Christmas tree - desired height: 7

A Christmas tree that is seven levels tall² would look like (red: output)

```
      x
     xxx
    xxxxx
   xxxxxxx
  xxxxxxxxx
 xxxxxxxxxxx
xxxxxxxxxxxxx
      xxx
      xxx
```

} 7
} 2

... whereas a four-level tree would look like

Christmas tree - desired height: 4

```
      x
     xxx
    xxxxx
 xxxxxxxx
      x
      x
```

} 4
} 2

Create a program that asks the user for the height of the Christmas tree (=input) and then draws the tree (output always with a trunk of 2 levels). If the height of the Christmas tree is <5, then the trunk should be 1 x wide, otherwise 3 x => xxx.

¹ trunk = Stamm

² tall = hoch