Python project: Guess the number

* **Challenge type: Learning & Consolidation**
* **Duration: 1 day**
* **Deadline**: **Wednesday 4.30PM**
* **Team challenge** : solo

# Goal

In this game, the computer will generate a random number between 1 and 100, and the player will have to guess what the number is. The player will have a maximum of 10 guesses to get the number right. After each guess, the computer will tell the player whether the number they guessed is higher or lower than the actual number. If the player guesses the number correctly within the maximum number of guesses, they win the game!

# Steps to complete

* Create a function that generates a random number between 1 and 100
* Create a function that manages one round of play:
  + Ask the player to enter a number
  + Ensure that the number is valid and included in the interval 1-100
  + Check if the player’s guess is correct, too high or too low and inform him/her
  + The function should return True if the guess is correct, else return False
* Create a loop for managing the game
  + Use the functions created above to generate a number and check if the player guessed correctly
  + If the player guesses correctly the loop should stop and a congratulations message should appear: “You guessed the correct number in XX rounds”
  + If the number is not guessed after 10 rounds, a message should appear: “You failed, try again!”
* [Optional]
  + Prompt the player to enter their name at the beginning of the game and include the name in the congratulations or failure message

# Deliverables

* Create a folder in your learner space that contains:
* Your Google Colab notebook (use markdown for structuring your file)
* A report in the MD format that contains:
  + A description of the project
  + The structure of your code
  + Instructions: how to play the game?
  + (Visuals)
  + (Contributors)
  + (Timeline)
  + (Personal situation)

# Pedagogical objectives

Be able to implement basic Python programming concepts in a full project:

* Variables and data types
* Conditional statements
* Loops and iterations
* Functions and modular programming
* User input and output
* Basic error handling