

# Table of Contents

- [Aim](#)
- [Implementation](#)
- [For versus While](#)
- [Sample Output](#)

## Aim

Create a number guessing game divided into three levels of difficulty. I divide this project into small functions and place them in separate files to be used as modules.

## Implementation

I begin by greeting the user and prompt them to choose which level (easy, medium, or hard) they want to play | welcome.py. With that option, I then start the corresponding function which is stored in options.py

Taking easy level as an example, when the game starts (which now go to the module operations.py, I generate random numbers by importing the random module and referencing the randint() function defined in that module (ranging from 1 to 10). I then assign the random number generated to a variable called random. Since the user fails after six tries, I assign 6 to the guess\_limit variable and they will lose one every time a guess is made. The guess is assigned to the guess variable. A False Boolean is assigned to the done variable to keep the game running until it becomes True, which means that the game is done

Whenever a guess is made, it is evaluated by the function validate\_input(), using a while loop to check whether it is valid, which means whether it is in the range between 1 and 6. If it is an invalid guess (the function returns False), the user has to make a new one until it is valid (the function returns True). Provided that the guess is invalid, the user still maintains their number of guesses.

Now that the guess is valid, one chance will be taken away. Afterwards, a while loop is used to evaluate that guess so long as the user still has at least one chance, and the game is not done.

If the user has more than zero chance left, their guess will be compared to the random number generated to check whether it is too low or too high, or equal:

- If they are equal, a congratulatory message will be printed, which also shows how many times it took the user to guess correctly. In this case, the done variable becomes True and ends the game.
- If they are not equal, the user then makes another guess on the condition that they have not lost all chances and the game is still running. As always, the new guess is then evaluated again by the function validate\_input() to check for its validity.

When the user reaches their last chance and the game still runs, meaning that the user has used up all of the six attempts, their last guessed number will be evaluated with the number generated, and only two options are presented:

- If they are equal, a congratulatory message will be printed.
- If they are still not equal, a 'Game Over' message will be printed.

In both cases, the game ends and the while loop is displayed.

## For versus While

<b><i>While loops</i></b>	<b><i>For loops</i></b>
Condition-controlled loop	Count-controlled loop
They are better to use whenever the programmer needs to keep on repeating some code while a certain condition is met	They are not necessary but simplify the code when the number of repetitions is known

Regarding the while loop evaluating whether the guessed number is valid, it helps rule out any invalid guess. In this case, the number of times this loop would repeat is unpredictable, given that it depends upon the input from the user. Furthermore, this loop only gets finished provided that a certain condition – the user input guess is between 1 and 5 – is met. This problem, therefore, should be addressed with a while loop instead of a for loop.

## Sample Output

```
Hello, Welcome to the Number Guessing Game!
I'm Henry! What's Your Name? Jack
Okay, Jack. Let's Begin The Guessing Game!
Choose a level:
1. Easy
2. Medium
3. Hard
Pick a number: 3

You are to guess a number between 1 and 50 in no more than 3 attempts.
What is your guess? 3
It should be higher than 3.
You still have 2 chances left.

Try a new guess: 15
It should be higher than 15.
You still have 1 chances left.

Try a new guess: 20
GAME OVER! It took you more than 3 attempts. The correct number is 41.

Do you want to play again? [Y/n] n

Thanks for playing !
```