

## Exploration of While Loops

Download the script `guess.py` but don't look at the source code yet! We will begin by having you explore the software as a user.

### Part 1: Exploring the software as a user

1. Describe in English what this program does.
2. How does the program respond if you enter a string other than C, H, or L?
3. Give an example of a secret number that forces the program to make 7 guesses.
4. Let's say that your secret number is 55 but that you incorrectly answer the first question by suggesting that 50 is too **High** of a guess. After that, you answer all remaining questions accurately. What happens to the execution of the program?

## Part 2: Examining the source code

Okay. It's time to start examining the source code. Here is a copy, with line numbers added for reference in our discussion.

```
1 print("Think of a number from 1 to 100 but don't tell me!")
2 print("I'm going to guess it, but I need you to respond.")
3 print("to each question with character 'C', 'L', or 'H'.")
4 print( )
5
6 lowestPossible = 1
7 highestPossible = 100
8 found = False
9 guessCount = 0
10 while not found:
11     guess = (lowestPossible + highestPossible) // 2
12     prompt = 'Is ' + str(guess) + ' (C)orrect, too (L)ow, or too (H)igh? '
13     feedback = input(prompt).strip( )
14     if feedback in ('C','L','H'):
15         guessCount += 1
16         if feedback == 'C':
17             found = True
18         elif feedback == 'H':
19             highestPossible = guess - 1
20         else:
21             lowestPossible = guess + 1
22     else:
23         print('Invalid response')
24 print('I found it in', guessCount, 'guesses!' if guessCount > 1 else 'guess!')
```

5. What is the purpose of variable `guessCount`? Why is it initialized as such at line 9? How/when does it get updated?
6. What is the purpose of variable `lowestPossible`? Why is it initialized as such at line 6? How/when does it get updated?

7. What is the purpose of variable `highestPossible`? Why is it initialized as such at line 7? How/when does it get updated?
  
8. What is the purpose of variable `found`? Why is it initialized as such at line 8? How/when does it get updated?
  
9. Why is the `//` operator used on line 11? What would be the impact on the program at large if the `/` operator were used instead?
  
10. What is the impact of the `strip( )` method on line 13? Give a tangible example of a scenario in which the program behaves differently were this command `feedback = input(prompt)`.
  
11. Line 15 executes `guessCount += 1`. Why would the program be flawed if this command were instead done just after line 13 (with appropriate indentation). Given a tangible example of a scenario in which the program behaves differently with such a modification.
  
12. Line 20 uses a simple **else** statement. Given that the user could enter any string, why isn't it necessary for us to more explicitly check **elif feedback == 'L' ?**

### Part 3: Writing your own program

The original program is designed for the user to pick a secret and the software to guess it. We would like you to write a program now that has the computer picking the secret and the user trying to guess it. A sample run of the program might appear as follows:

```
I've picked a random number from 1 to 100, try to guess it.
Enter your guess: 50
Too high
Enter your guess: 25
Too low
Enter your guess: 37
Too low
Enter your guess: 43
Too high
Enter your guess: 40
You found it in 5 guesses!
```

We will guide you through the process by setting some intermediate goals. Write Python code for each step of the process.

14. First we need to have the computer pick a random number. This can be done using the `randint` function that is part of the `random` library:

```
import random
secret = randint(1,100)
```

Note that `randint` picks a random number anywhere from 1 to 100, including 1 and 100 as possibilities.

15. Initialize a variable `guess` that stores the current user's guess. You should determine what it should be initialized too.

16. Write the start of a **while** loop that will continue as long as the user has not correctly guessed the secret answer. (Hint: this should use the variables defined in the first two steps.)

