

# 3.2. Guessing Numbers: Using Loops & Conditionals

Note: The numbers in the Card match the Tasks numbers in the Jupyter Notebook.

1. To program the game, we need **conditional statements**. Computers understand conditions. For example, **if**  $x = 5$ , **then** stop counting. *How do we know it is a condition? Do your math teachers give you conditions? How are the conditions in CELLS similar or different from*

Run: CELLS A & B to explore conditions. Try out different numbers.

```
number = int(input(" Input your number "))
secret_number = 5
if (number == secret_number):
    # Which number(s) will execute the code below?
    # For number=???, the following code works:
    print("You guessed it!")
    print("You win.")
```

In the code, where's the 'condition'?

What do strings do here?

3. Run & Compare: What happens in CELLS 3.A and 3.B? Why?

3.A. `while(0):`  
`print(1)`

3.B. `while(1==2):`  
`print(1)`

If your code runs forever, press stop

Change 2 to 1. What happens? Why?

It's called an infinite loop. *What's the condition in the loop?*

Run and change values in CELLS C and D.

Which of these ideas can you use in your game?

`!=` means **not equal**  
Why is it useful here?

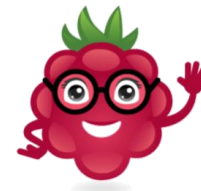
```
while (computer_guess_number != secret_number):
```

2. Clear **CELLS** to enter new values or numbers. Remember how to clear CELLS in jupyter?

Also, always save your work!!

Clear cell results by following these steps. Or click on run again:

4. Design: In **CELL E** create your game. Change the code as you need. Plan the steps to play the game.



Think: what code will help you create your game?

**Think & Compare:**  
*How are these statement different? What do they do?*  
`-for i in range(start_i, stop_i, step_i):`  
`-if (number == secret_number):`  
`-while (number != secret_number):`

# 3.2. Adivinando números: Usando Loops y Condicionales

**Nota:** Los números de la Tarjeta coinciden con los números de Tasks del Jupyter Notebook.

1. Para programar el juego se necesitan comandos condicionales. Las computadoras entienden condiciones. Por ejemplo, si  $x=5$ , entonces parar de contar. ¿Cómo saben que esta es una condición? ¿Les dan condiciones sus maestros de matemáticas? ¿Qué tanto se parecen las condiciones en las CELLS?

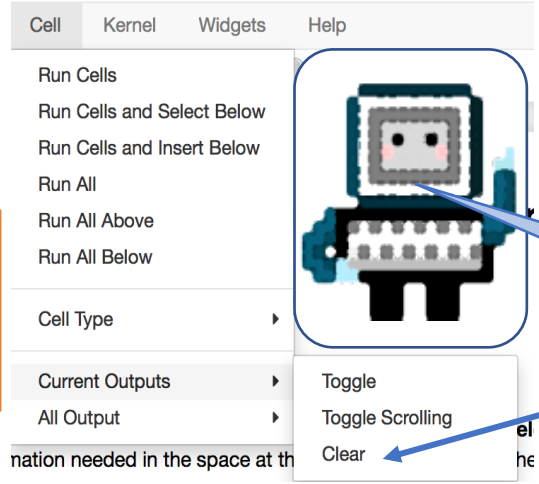
```
number = int(input(" Input your number "))
secret_number = 5
if (number == secret_number):
    # Which number(s) will execute the code
    # For number=???, the following code works
    print("You guessed it!")
    print("You win.")
```

¿Cuál es la "condición" en el código?

¿Para qué sirven las strings?

Corre y explora condiciones en CELLS A y B, y prueba números distintos

2. Limpia las **celdas** para introducir nuevos números o valores.



¿Recuerdan como limpiar las CELDAS en **jupyter**?

¡No olviden guardar su trabajo!



Sigan estos pasos para limpiar las celdas. O presiona run otra vez:



3. **Corran y Comparen** ¿Qué pasa **CELLs 3.A y 3.B**? ¿por qué?

3.A. **while(0):**

```
print(1)
```

3.B. **while(1==2):**

```
print(1)
```

Si tu código corre sin parar, presiona stop



Cambia el 2 por 1. ¿Qué pasa y por qué?

A esto se le llama loop infinito. ¿Cuál es la condición en el loop?

**Corran y** cambia valores en CELL C y D.

¿Cuál de estas ideas quieres usar en tu juego?

!= significa **no igual**  
¿Por qué se usa acá?

```
while (computer_guess_number != secret_number):
```



4. **Diseña:** En **CELL E** crea tu juego. Cambia el código como se necesite. Planeen los pasos de su juego.



**Piensa y Compara:**  
¿En qué se diferencian estos statements? ¿Qué hacen?  
-**for** i in range(start\_i, stop\_i, step\_i):  
-**if** (number == secret\_number):  
-**while** (number != secret\_number):

**Piensa:** qué código te ayudará a crear tu juego?