

2019

Proyecto Python



Fernando García Ballesteros

Saúl Lozano Selva

31/01/2019

ÍNDICE

No se encontraron elementos de tabla de contenido.

INTRODUCCIÓN

Hemos realizado un proyecto en Python llamado búsqueda binaria, que consiste en tratar de adivinar una posición (en este caso un número) elegido al azar por el programa y dando pistas que indican si el numero elegido por ti se halla por encima o por debajo del correcto.

CÓDIGO

```
import random
guessesTaken = 0
number = random.randint(1,100)
print("Hola, estoy pensando un número del 1 al 100. Tienes 5 oportunidades para adivinarlo")
while guessesTaken < 6:
    print("¿Cual es tu número?") # 4 espacios indican un bloque
    guess = input()
    guess = int(guess)
    if guess >= 1 and guess <= 100:
        guessesTaken = guessesTaken + 1
        if guess < number:
            print(str(guessesTaken) + ". Muy bajo") # 8 espacios indican un segundo nivel de un bloque.
        if guess > number:
            print(str(guessesTaken) + ". Muy alto")
        if guess == number:
            break
    else:
        print("Tu número esta fuera de rango. Intenta con otro número.")

if guess == number:
    guessesTaken = str(guessesTaken)
    print("Excelente Adivinaste mi número en  + guessesTaken +  oportunidades!")

if guess != number:
    number = str(number)
    print("No, El número que estaba pensando es el  + number")
```

EXPLICACIÓN DEL CÓDIGO

```
import random
guessesTaken = 0
number = random.randint(1,100)
print("Hola, estoy pensando un número del 1 al 100. Tienes 5 oportunidades para adivinarlo")
```

En esta primera parte del código importamos la librería random, en la segunda línea definimos una variable con el nombre guessesTaken que corresponde con los intentos para adivinar el número. Y definimos la variable number e indicamos que le dé un valor de forma aleatoria cada vez que ejecutemos el programa y que oscile entre los valores 1 y 100. Y en la última línea con el comando print escribimos al inicio del juego la frase que se encuentra entre paréntesis.

```

while guessesTaken < 5:
    print("¿Cual es tu número?")
    guess = input()
    guess = int(guess)
    if guess >= 1 and guess <= 100:
        guessesTaken = guessesTaken + 1
        if guess < number:
            print(str(guessesTaken) + ". Muy bajo")
        if guess > number:
            print(str(guessesTaken) + ". Muy alto")
        if guess == number:
            break
    else:
        print("Tu número esta fuera de rango. Intenta con otro número.")

```

En esta parte del código, primero indicamos que el `guessesTaken` es menor que 5 mediante la instrucción `while`. Posteriormente, mediante la instrucción `print`, hacemos que aparezca en el juego la frase entre paréntesis. Las dos siguientes líneas son las que permiten la introducción de tu número pensado, de forma que si dicho número está entre 1 y 100, aumentamos en una unidad el número de intentos y determinamos que si es mayor que el numero correcto se escriba “Muy alto”, si es menor se escriba “Muy bajo” y si es igual se acaba la función. Finalmente se escribe una línea para que si el número que piensas es mayor de 100 o menor de 1 te indique que esta fuera de rango.

```

if guess == number:
    guessesTaken = str(guessesTaken)
    print("Excelente Adivinaste mi número en" , guessesTaken , "oportunidades")

if guess != number:
    number = str(number)
    print("No, El número que estaba pensando es el", number)

```

Si adivinas el número elegido por el programa este indica que has acertado y en el número de intentos que lo has conseguido, si no aciertas el numero en los intentos definidos el programa escribe el número en el que estaba pensando.

APLICACIONES

La posibles aplicaciones de este programa, son bastante amplias, de esta forma un ejemplo podría ser, en una fuga de agua, para detectarla rápidamente sin tener que excavar todo lo largo de lo que abarca esta...

O por ejemplo en un circuito eléctrico, detectar una posible fuga de electricidad en mucho menos tiempo.

CONCLUSIÓN

Este programa puede servir como un juego como en el caso que nosotros lo hemos utilizado o de muchas maneras prácticas para facilitar tareas de reparación o comprobación de diferentes tipos de circuitos por ejemplo. Este programa no tiene mucha complejidad pero incluye cosas que antes no habíamos utilizado y hemos tenido que buscar en internet.