

```

enemies = 1
def increase_enemies():
    enemies = 2
    print(f"enemies inside function: {enemies}")

increase_enemies()
print(f"enemies outside function: {enemies}")

    enemies inside function: 2
    enemies outside function: 1

```

```

game_level = 3

enemies=["Skeleton", "Zombie", "Creeper"]

if game_level < 5:
    new_enemy = enemies[0]

print(new_enemy)

```

📄 Skeleton

```

#Modifying the Global Scope

enemies.=.1.#Global.Scope
def.increase_enemies():
    ..enemies.+=.1#Local.Scope
    ..print(f"enemies.inside.function:.{enemies}")

increase_enemies()
print(f"enemies.outside.funciton:.{enemies}")

```

```
UnboundLocalError                                Traceback (most recent call last)
<ipython-input-9-3209029b04d4> in <module>
```

El error anterior se debe a que dentro de la función no se conocia a la variable global, hay que enunciarla explicatamente

```
print('elements outside function: ', elements)
```

```
enemies = 1 #Global Scope
def increase_enemies():
    global enemies    #Ahora si funciona, porque cambiamos la variable global dentro de un ent
    enemies += 1 #Local Scope
    print(f"enemies inside function: {enemies}")

increase_enemies()
print(f"enemies outside funciton: {enemies}")

enemies inside function: 2
enemies outside funciton: 2
```

```
enemies = 1 #Global Scope
def increase_enemies():
    print(f"enemies inside function: {enemies}")
    return enemies + 1    #This calls the GLOBAL variable and changes it

enemies = increase_enemies()
print(f"enemies outside funciton: {enemies}")

enemies inside function: 1
enemies outside funciton: 2
```

#Global constants

PI = 3.1415926

```
logo = ""
```

The figure shows three 4x4 grids. Each grid has a path highlighted by a solid line. The path starts at the bottom-left cell (row 4, column 1) and ends at the top-right cell (row 1, column 4). The grid is divided into three sections by vertical dashed lines. The first section is 2x2, the second is 2x2, and the third is 2x2. The path is highlighted by a solid line.

■ ■ ■

```

import random as rd
print("Welcome to guess the number, I'm thinking a number between 1 and 100")

number = rd.choice(range(1,100))

def guesser():
    guess=int(input('Make a guess: '))
    if guess == number:
        return 0
    elif guess < number:
        print("Too low")
        print("Guess again!")
    else:
        print("Too high")
        print("Guess again!")

def game():

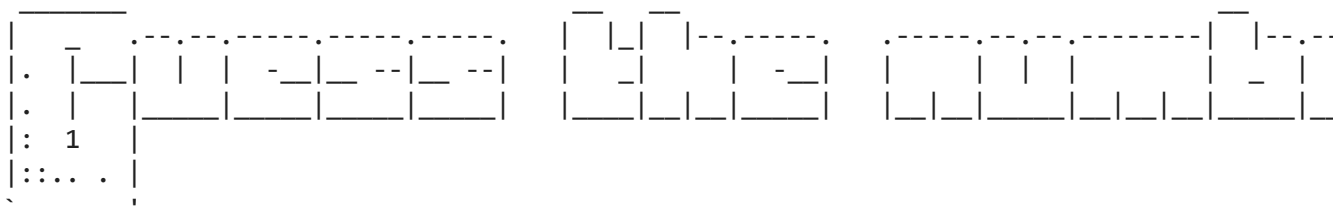
    Should_stop = False
    print(logo)
    dif = input("Choose your difficulty type 'easy' or 'hard': ")
    if dif == "easy":
        lives=10
    else:
        lives=5

    while not Should_stop:
        if lives !=0:
            print(f"You have {lives} attempts remaining to guess the number")
            a=guesser()
            lives -= 1
            if a==0:
                print(f"You win the number was {number}")
                return
            else:
                print(f"You've run out of guesses, you lose... The number was {number}")
                return

game()

```

Welcome to guess the number, I'm thinking a number between 1 and 100



```
Choose your difficulty type 'easy' or 'hard': y
You have 5 attempts remaining to guess the number
Make a guess: 3
Too low
Guess again!
You have 4 attempts remaining to guess the number
Make a guess: 3
Too low
Guess again!
You have 3 attempts remaining to guess the number
Make a guess: 3
Too low
Guess again!
You have 2 attempts remaining to guess the number
Make a guess: 3
Too low
Guess again!
You have 1 attempts remaining to guess the number
Make a guess: 3
Too low
Guess again!
You've run out of guesses, you lose... The number was 55
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 6s completed at 22:04

● ✕