06 – FUNCIONES

Jorge Muñoz

IIC1103 – Introducción a la Programación

FUNCIONES

```
NUM 4 CIFRAS? 1110
Numero de 0: 1
Numero de 1: 3
Numero de 2: 0
...
```

```
n = int(input("NUM 4 CIFRAS? "))
d4 = (n // 1) % 10
d3 = (n // 10) % 10
d2 = (n // 100) % 10
d1 = (n // 1000) % 10

c0 = 0
if d4 == 0:
    c0 = c0 + 1
elif d3 == 0:
    c0 = c0 + 1
elif d2 == 0:
    c0 = c0 + 1
elif d1 == 0:
    c0 = c0 + 1
print("Numero de 0:", c0)

c1 = 0
if d4 == 1:
    c1 = c1 + 1
elif d3 == 1:
    c1 = c1 + 1
elif d2 == 1:
    c1 = c1 + 1
elif d1 == 1:
    c1 = c1
print("Numero de 1:", c1)

c2 = 0
if d4 == 2:
    c1 = c2 + 1
elif d3 == 2:
    c2 = c2 + 1
elif d2 == 2:
    c2 = c2 + 1
elif d1 == 1:
    c2 = c2 + 1
print("Numero de 2:", c2)
#Y seguir con 3,4,5,6,7,8,9
```

```
NUM 4 CIFRAS? 1110
Numero de 0: 1
Numero de 1: 3
Numero de 2: 0
...
```

```python
n = int(input("NUM 4 CIFRAS? "))
d4 = (n // 1) % 10
d3 = (n // 10) % 10
d2 = (n // 100) % 10
d1 = (n // 1000) % 10

c0 = 0
if d4 == 0:
    c0 = c0 + 1
elif d3 == 0:
    c0 = c0 + 1
elif d2 == 0:
    c0 = c0 + 1
elif d1 == 0:
    c0 = c0 + 1
print("Numero de 0:", c0)

c1 = 0
if d4 == 1:
    c1 = c1 + 1
elif d3 == 1:
    c1 = c1 + 1
elif d2 == 1:
    c1 = c1 + 1
elif d1 == 1:
    c1 = c1
print("Numero de 1:", c1)

c2 = 0
if d4 == 2:
    c1 = c2 + 1
elif d3 == 2:
    c2 = c2 + 1
elif d2 == 2:
    c2 = c2 + 1
elif d1 == 1:
    c2 = c2 + 1
print("Numero de 2:", c2)
#Y seguir con 3,4,5,6,7,8,9
```

```
NUM 4 CIFRAS? 1110
Numero de 0: 1
Numero de 1: 1
Numero de 2: 1
...
```

5 errores

¿Los puedes encontrar todos?

```
NUM 4 CIFRAS? 1110
Numero de 0: 1
Numero de 1: 3
Numero de 2: 0
...
```

```
n = int(input("NUM 4 CIFRAS? "))
d4 = (n // 1) % 10
d3 = (n // 10) % 10
d2 = (n // 100) % 10
d1 = (n // 1000) % 10

c0 = 0
if d4 == 0:
    c0 = c0 + 1
elif d3 == 0:
    c0 = c0 + 1
elif d2 == 0:
    c0 = c0 + 1
elif d1 == 0:
    c0 = c0 + 1
print("Numero de 0:", c0)

c1 = 0
if d4 == 1:
    c1 = c1 + 1
elif d3 == 1:
    c1 = c1 + 1
elif d2 == 1:
    c1 = c1 + 1
elif d1 == 1:
    c1 = c1
print("Numero de 1:", c1)

c2 = 0
if d4 == 2:
    c1 = c2 + 1
elif d3 == 2:
    c2 = c2 + 1
elif d2 == 2:
    c2 = c2 + 1
elif d1 == 1:
    c2 = c2 + 1
print("Numero de 2:", c2)
#Y seguir con 3,4,5,6,7,8,9
```

```
NUM 4 CIFRAS? 1110
Numero de 0: 1
Numero de 1: 1
Numero de 2: 1
...
```

elif => if

¡Que paja cambiar los todos 30!

falta +1

Al copiar y pegar se me cortó

es c2

Al copiar y pegar se paso cambiarlo

es == 2

Al copiar y pegar se paso cambiarlo

```
def contar(c1,c2,c3,c4,x):
    res = 0
    if c4 == x:
        res = res + 1
    if c3 == x:
        res = res + 1
    if c2 == x:
        res = res + 1
    if c1 == x:
        res = res + 1
    return res
```

parámetros

resultado

función es burbuja

Solo se pasa información con el exterior por parámetros y resultado

```
n = int(input("NUM 4 CIFRAS? "))
d4 = (n // 1) % 10
d3 = (n // 10) % 10
d2 = (n // 100) % 10
d1 = (n // 1000) % 10

c0 = contar(d1,d2,d3,d4,0)
print("Numero de 0:", c0)
c1 = contar(d1,d2,d3,d4,1)
print("Numero de 1:", c1)
c2 = contar(d1,d2,d3,d4,2)
print("Numero de 2:", c2)
c3 = contar(d1,d2,d3,d4,3)
print("Numero de 3:", c3)
c4 = contar(d1,d2,d3,d4,4)
print("Numero de 4:", c4)
c5 = contar(d1,d2,d3,d4,5)
print("Numero de 5:", c5)
c6 = contar(d1,d2,d3,d4,6)
print("Numero de 6:", c6)
c7 = contar(d1,d2,d3,d4,7)
print("Numero de 7:", c7)
c8 = contar(d1,d2,d3,d4,8)
print("Numero de 8:", c8)
c9 = contar(d1,d2,d3,d4,9)
print("Numero de 9:", c9)
```

llamar a la función

```
NUM 4 CIFRAS? 1110
Numero de 0: 1
Numero de 1: 3
Numero de 2: 0
...
```

ES PAR

```
1 es par: False
20 es par: True
38 es par: True
79 es par: False
```



```
#CODIGO PRINCIPAL
x1 = espar(1)
print("1 es par:", x1)

x20 = espar(20)
print("20 es par:", x20)

x38 = espar(38)
print("38 es par:", x38)

x79 = espar(79)
print("79 es par:", x79)
```

```
1 es par: False
20 es par: True
38 es par: True
79 es par: False
```

```python
def espar(num):

    #Calculo el resto (modulo) de dividir
    #por 2 con el numero que me den
    res = num % 2

    #Si el resto es 0 es par, si no es impar
    if res == 0:
        sol = True
    else:
        sol = False

    #Retorno (devuelvo) la solucion
    #a quien me llamo
    return sol


#CODIGO PRINCIPAL
x1 = espar(1)
print("1 es par:", x1)

x20 = espar(20)
print("20 es par:", x20)

x38 = espar(38)
print("38 es par:", x38)

x79 = espar(79)
print("79 es par:", x79)
```

GLOBOS

Chile			A	B	C	D	E	F	G	H	I	J	K	L	M
1	[PUC-Ing] Bella y Sensual	9 863	+ 4	X (5)	+1 152	X (2)	+ 36	+1 30	+1 98	+ 241	+ 53	-	-	+ 41	+1 128
2	[UChile] whatsapp 2 dev team	8 851	+ 5	-	+ 132	-	+ 215	+1 33	+ 193	-	+1 48	-	-	+ 37	+ 148
3	[PUC-Ing] McNanoL	7 727	+1 7	-	X (9)	-	+2 221	+ 21	+1 213	-	+ 45	-	-	+ 11	+ 129
4	[PUC-Mat] Club de Yate Pontificia Universidad Catolica del Perú	5 259	+ 3	-	-	-	X (2)	+ 23	X (2)	-	+ 52	-	-	+ 69	+ 112
5	[UTFSM] Churrasco a lo fuerza bruta	5 262	+ 3	-	+ 99	X (1)	-	+1 46	X (4)	-	+ 20	-	-	+1 54	X (5)
6	[UdeC] Flip Flop	5 534	+ 5	-	-	-	X (6)	+ 27	-	-	+1 136	-	-	+ 39	+1 287
7	[UTalca-Ing] Team Generico	4 653	+ 16	-	-	-	X (7)	+1 59	-	-	+2 265	-	-	+1 233	-
8	[PUC-Ing] Bubblegum	3 58	+ 2	-	-	-	-	+ 13	-	-	X (1)	-	-	+ 43	-
9	[UTFSM] duro duro y seco	3 223	+ 4	-	-	-	-	+1 41	-	-	-	-	-	+1 138	X (1)
10	[UCN] Empanada de queso	3 254	+ 11	-	-	-	-	+ 34	-	-	+1 189	-	-	X (7)	-
11	[UANDES] Los typos	3 262	+ 8	-	-	-	-	+ 97	-	-	-	-	-	+2 117	X (2)
12	[UDP] Busco pega	3 317	+ 5	-	-	-	X (1)	+4 99	-	-	X (7)	-	-	+2 93	-
13	[UdeC] Los Chakalitos	3 408	+ 21	-	-	X (2)	X (1)	+1 118	-	-	+ 249	-	-	X (1)	-
14	[UTFSM] WA in test 999	2 42	+ 19	-	-	-	-	+ 23	-	-	X (3)	-	-	X (2)	-
15	[UTFSM] son las 2 am, mi mente hizo .pop() a la materia gris	2 88	+ 8	-	-	-	X (5)	+ 80	-	-	-	-	-	X (7)	-
16	[UDP] 0111up	2 144	+1 41	-	-	-	X (1)	+ 83	X (1)	-	X (4)	-	-	-	-

socrative.com

Student Login

JORGEINTRO

JORGEINTRO

¡NOMBRE REAL!

1 IntroCoin a quien hacierte todas

'a','b', y 'r' se llaman igual dentro y fuera

Oh, que casualidad, … pero que más da!

```
#IN OUT IGUAL
def restar(a,b):
    r = a - b
    return r

a = 10
b = 2
r = restar(a,b)
print(r)
```

```
>>>
8
>>>
```

'r' se llaman igual dentro y fuera, pero 'a,b' y 'x,y' no.

Oh, que casualidad, ... pero que más da!

```
#IN DIF OUT IGUAL
def restar(a,b):
    r = a - b
    return r

x = 10
y = 2
r = restar(x,y)
print(r)
```

```
>>>
8
>>>
```

'a,b' se llaman igual dentro y fuera, pero 'r' y 'res' no.

Oh, que casualidad, … pero que más da!

```
#OUT DIF IN IGUAL
def restar(a,b):
    r = a - b
    return r

a = 10
b = 2
res = restar(a,b)
print(res)
```

```
>>>
8
>>>
```

Estas intentando usar 'a,b' en fuera de la función

¿!Pero no te he dicho, que lo que pasa en la función se queda en la función?!

```
#SCOPE IN EN PRINCIPAL
def restar(a,b):
    r = a - b
    return r

x = 10
y = 2
r = restar(x,y)
print(a,"menos",b,"es",r)
```

```
NameError: name 'a' is not defined
>>>
```

Estas intentando usar 'r' en fuera de la función

¿!Pero no te he dicho, que lo que pasa en la función se queda en la función?!

```
#SCOPE OUT
def restar(a,b):
    r = a - b
    return r

a = 10
b = 2
res = restar(a,b)
print(r)
```

```
NameError: name 'r' is not defined
>>>
```

Lo llamas con b y a, pero dentro se llaman a y b

No le importa los nombres, solo el ORDEN

```
#IN EN DESORDEN
def restar(a,b):
    r = a - b
    return r

a = 10
b = 2
r = restar(b,a)
print(r)
```

```
>>>
-8
>>>
```

Aunque parezca que la función esta pensada para sumar dos números, le llegan 2 str como parámetros

En Python nadie fuerza a que algo sea de un tipo concreto. Tu mismo has creado la función como has querido, simplemente llámala como pensaste

```
#SUMA STR
def sumar(num1,num2):
    r = num1 + num2
    return r

num1 = "6"
num2 = "7"
r = sumar(num1,num2)
print(r)
```

```
>>>
67
>>>
```

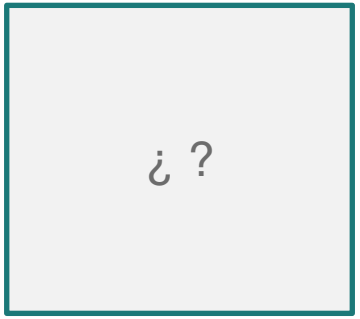
DIGICORRECT

```
el digito 0 de 32048 es 8? True
el digito 1 de 678 es 6? False
el digito 4 de 123456 es 3? False
```

```
def digixtract(num,p):
    a = 10 ** p
    b = 10 ** (p + 1)

    r1 = num % b
    res = r1 // a

    return res
```




```
#CODIGO PRINCIPAL
x = digicorrect(32048,0,8)
print("el digito 0 de 32048 es 8?", x)

y = digicorrect(678,1,6)
print("el digito 1 de 678 es 6?", y)

z = digicorrect(123456,4,3)
print("el digito 4 de 123456 es 3?", z)
```

```
el digito 0 de 32048 es 8? True
el digito 1 de 678 es 6? False
el digito 4 de 123456 es 3? False
```

```python
def digixtract(num,p):
    a = 10 ** p
    b = 10 ** (p + 1)

    r1 = num % b
    res = r1 // a

    return res

def digicorrect(n,p,x):
    d = digixtract(n,p)
    if d == x:
        sol = True
    else:
        sol = False
    return sol


#CODIGO PRINCIPAL
x = digicorrect(32048,0,8)
print("el digito 0 de 32048 es 8?", x)

y = digicorrect(678,1,6)
print("el digito 1 de 678 es 6?", y)

z = digicorrect(123456,4,3)
print("el digito 4 de 123456 es 3?", z)
```

FUNCIONES DADAS

```
x = int("5")
```

int(), str(), print(),
input(), …

BUILT IN
Se usan al tiro,
sin hacer nada
más antes

¡NUEVA!

```
x = round(3.398,1)
y = round(44.234,2)
```

```
3.4
44.23
```

```
# Los .5 => al PAR mas cercano
x = round(3.5)     4
y = round(4.5)     4
```

IMPORTABLES YA POR DEFECTO

Hay que importar el módulo donde están 1 sola vez al principio para indicar que las queremos usar

Para usarlas ponemos el módulo y . antes

No es necesario instalar nada (vienen por defecto en Python)

```
import math
x = math.sqrt(9)

import random
y = random.randint(1,10)
```

¡TEMARIO SÓLO RANDINT!

IMPORTABLES DESDE OTRO ARCHIVO

Funciones en un archivo de Python se pueden usar en otro, simplemente importándolas con el nombre del archivo

***Clearn* usa esto**
(aunque a veces te prohíbe ver lo que hay dentro del archivo)



areas.py



mitarea.py

```python
def calcular_rectangulo(base,altura):
    area = base * altura
    return area

def calcular_cuadrado(lado):
    area = lado ** 2
    return area
```

```python
import areas

#TAREA: CALCULAR BASE DE RECTANGULO
b = int(input("BASE? "))
a = int(input("ALTURA? "))
res = areas.calcular_rectangulo(b,a)
print("AREA:", res)
```


cuando queres un numero aleatorio
pero se te olvida importar el modulo
random.randint
import random
random.randint


mi código bien escrito
random.randint(0,100)
import random

IMPORTABLES INSTALABLES

Infinitos módulos en internet que se pueden instalar y usar

¡NO ENTRA EN EL TEMARIO!

Tweepy

An easy-to-use Python library for accessing the Twitter API.





Scrapy

An open source and collaborative framework for extracting the data you need from websites. In a fast, simple, yet extensible way.

Cinemagoer DOWNLOADS SUPPORT DEVELOPMENT ECOSYSTEM

Cinemagoer (previously known as IMDbPY) is a Python package for retrieving and managing the data of the IMDb movie database about movies and people.

spotipy-dev/
spotipy

A light weight Python library for the Spotify Web API



Create production-grade machine learning models with TensorFlow

CALCULADORA, QUINIELA, REACTION TIME


Every journey begins
with a choice.



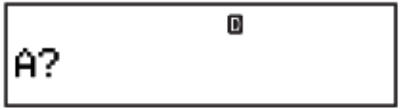
■ Creando un programa

◘ Creando un programa nuevo

Ejemplo: Crear un programa que determine el área de superficie de tres octaedros regulares, cada uno de los lados con un largo de 7 cm, 10 cm y 15 cm

EXE

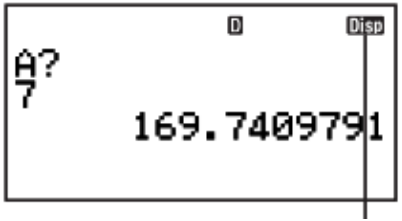
(Aparecerá un prompt para que se introduzca un valor para la variable A).



Introduzca 7 para A:

7 EXE

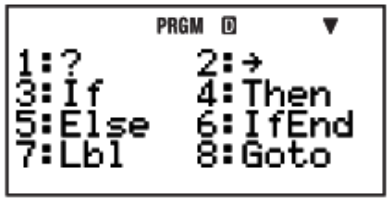
(Se visualizará el resultado calculado en el área de superficie S).



◘ Comandos de programa

Al seleccionar FUNCTION – {PROG} en la pantalla de edición de programas se visualizará el menú de los comandos de programa que se pueden usar para introducir el comando.

FUNCTION – {PROG}



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$x^2 - 5x + 6 = 0$

A? 1
B? -5
C? 6
SOLUCIONES: 3.0 y 2.0

$$x = \frac{5 \pm \sqrt{5^2 - 4 \cdot 6}}{2} = \frac{5 \pm \sqrt{25 - 24}}{2} = \frac{5 \pm 1}{2} = \begin{cases} \nearrow x_1 = \frac{6}{2} = 3 \\ \searrow x_2 = \frac{4}{2} = 2 \end{cases}$$

LA QUINIELA

1.	Granada	-	Espanyol	2-1	1
2.	Real Madrid	-	At. Madrid	1-0	1
3.	Mallorca	-	Valladolid	0-1	2
4.	Valencia	-	Celta	1-0	1
5.	Leganés	-	R. Sociedad	2-1	1
6.	Eibar	-	Betis	1-1	X
7.	Athletic Club	-	Getafe	0-2	2
8.	Sevilla	-	Alavés	1-1	X
9.	Ponferradina	-	Huesca	3-1	1
10.	Elche	-	Málaga	2-0	1
11.	Fuenlabrada	-	Girona	0-1	2
12.	Cádiz	-	R. Zaragoza	1-1	X
13.	R. Oviedo	-	Albacete	3-1	1
14.	Racing	-	Alcorcón	1-1	X

Estadísticamente

- *'1' es 50%*
- *'X' es 30%*
- *'2' es 20%*

Estadísticamente

- *'1' es 50%*
- *'X' es 30%*
- *'2' es 20%*

1	1	1
X	X	2
X	2	X
1	X	X
2	1	2
1	X	1
X	1	1
1	1	2
2	1	1
X	1	1
X	1	2
2	X	X
1	2	2
1	1	1



Reaction Time Test

When the red box turns green, click as quickly as you can.
Click anywhere to start.

ROJO
VERDE

TIEMPO (s) 0.25

ROJO
VERDE

TIEMPO (s) 0.338


Choose your starter

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$x^2 - 5x + 6 = 0$

```
A? 1
B? -5
C? 6
SOLUCIONES: 3.0 y 2.0
```

$$x = \frac{5 \pm \sqrt{5^2 - 4 \cdot 6}}{2} = \frac{5 \pm \sqrt{25 - 24}}{2} = \frac{5 \pm 1}{2} = \begin{cases} \nearrow x_1 = \frac{6}{2} = 3 \\ \searrow x_2 = \frac{4}{2} = 2 \end{cases}$$

```python
import math

a = int(input("A? "))
b = int(input("B? "))
c = int(input("C? "))

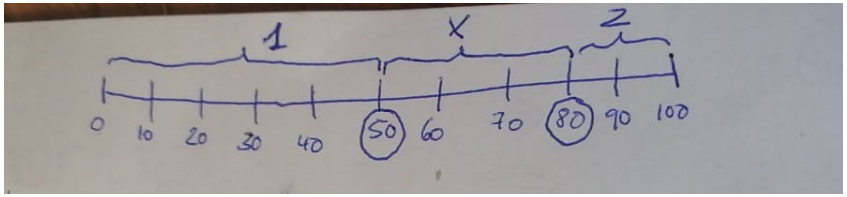
s1 = (b**2) - (4*a*c)
t1 = (-b) + math.sqrt(s1)
x1 = t1 / (2*a)

s2 = (b**2) - (4*a*c)
t2 = (-b) - math.sqrt(s2)
x2 = t2 / (2*a)

print("SOLUCIONES:", x1, "y", x2)
```

Estadísticamente

- *'1' es 50%*
- *'X' es 30%*
- *'2' es 20%*




```
------      ------      ------
1           1           1
X           X           2
X           2           X
1           X           X
2           1           2
1           X           1
X           1           1
1           1           2
2           1           1
X           1           1
X           1           2
2           X           X
1           2           2
1           1           1
------      ------      ------
```

```python
import random

print("------")

for i in range(1,15):
    a = random.randint(0,100)

    if a < 50:
        p = "1"
    elif a > 80:
        p = "2"
    else:
        p = "X"
    print(p)

print("------")
```

```
ROJO
VERDE

TIEMPO (s) 0.25
```

```
ROJO
VERDE

TIEMPO (s) 0.338
```

```python
import time
import random

print("ROJO")
a = random.randint(1,5)
time.sleep(a)

print("VERDE")
t1 = time.time()
x = input()
t2 = time.time()

t = t2 - t1
t = round(t,3)
print("TIEMPO (s)", t)
```

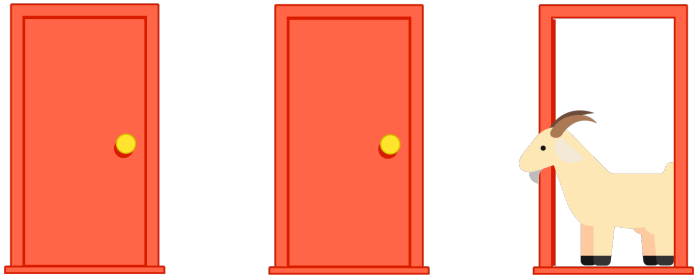
2 CABRAS Y UN FERRARI

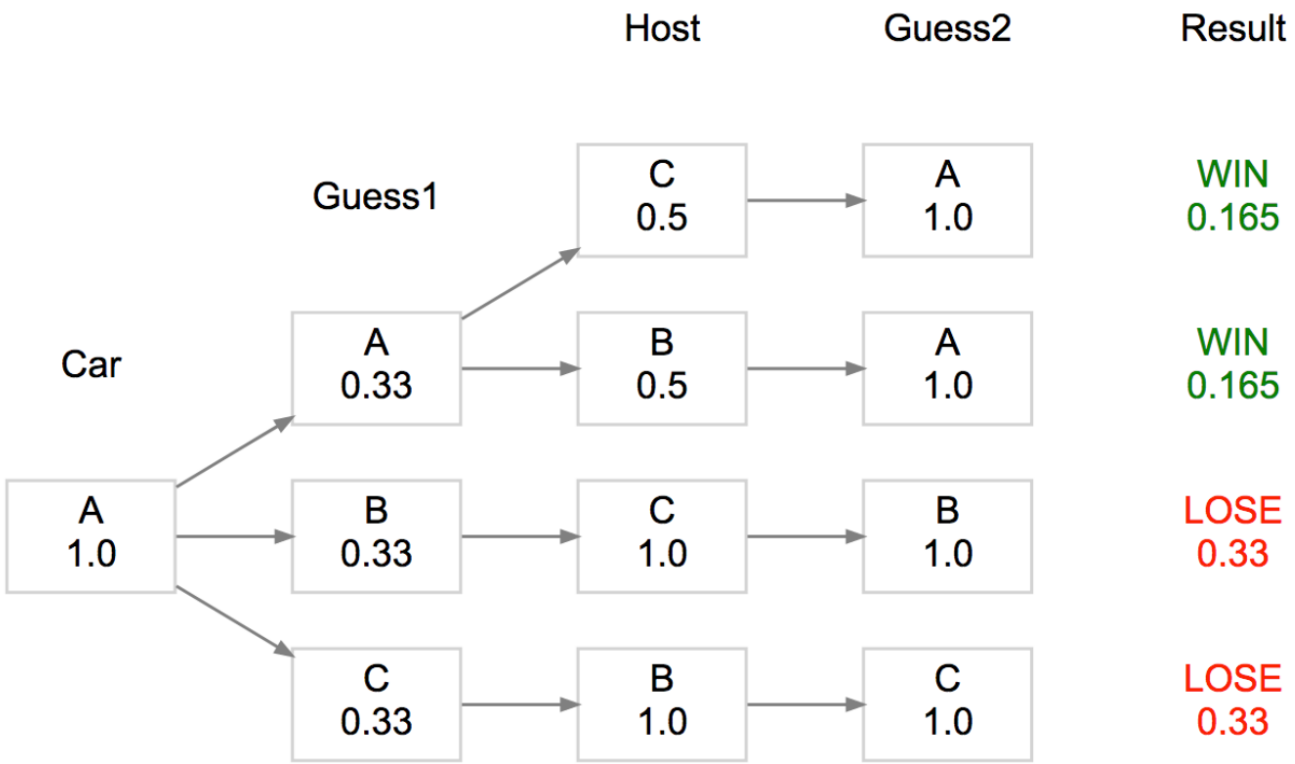
*Estás en un concurso de la tele. Hay 3 puertas **A**, **B**, y **C**. Detrás de una hay el **Ferrari** que siempre quisiste. En las otras dos hay una **cabra**. Odias las cabras.*

*Eliges la puerta **A**. El presentador abre la puerta **C** que descartaste y te muestra una cabra (**siempre abrirá una puerta con cabra** en el primer descarte).*

*Ahora te da a **elegir**:¿Quieres **seguir** con la puerta A o la **cambias** por la B?*

¿Qué harías?





Decision tree for the keeping strategy.

```
#PROGRAMA PRINCIPAL
num_partidas = 100000

#Jugar sin cambiar
vsin = 0
for i in range(0,num_partidas):
    cambiar = False
    ganado = jugar(cambiar)
    if ganado == True:
        vsin = vsin + 1

#Jugar cambiando
vcan = 0
for i in range(0, num_partidas):
    cambiar = True
    ganado = jugar(cambiar)
    if ganado == True:
        vcan =  vcan + 1

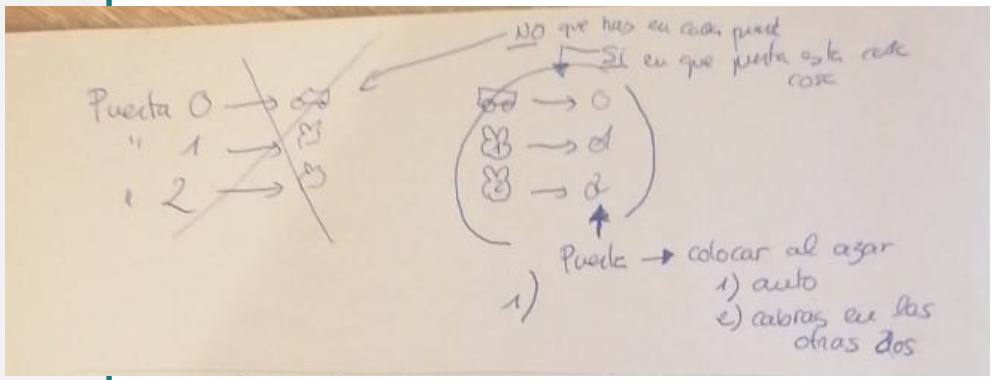
#Imprimir resultados
print("Victorias SIN CAMBIAR:",vsin,"/",num_partidas)
print("Victorias CAMBIANDO:",vcan,"/", num_partidas)
```

Victorias SIN CAMBIAR: 33452 / 100000
Victorias CAMBIANDO: 66440 / 100000

```
import random

'''Funcion que juega una partida.
Parametro: un bool que dice cambiara de puerta.
Retorno: un bool diciendo si gana.'''
```

¿ ?




```
#PROGRAMA PRINCIPAL
num_partidas = 100000

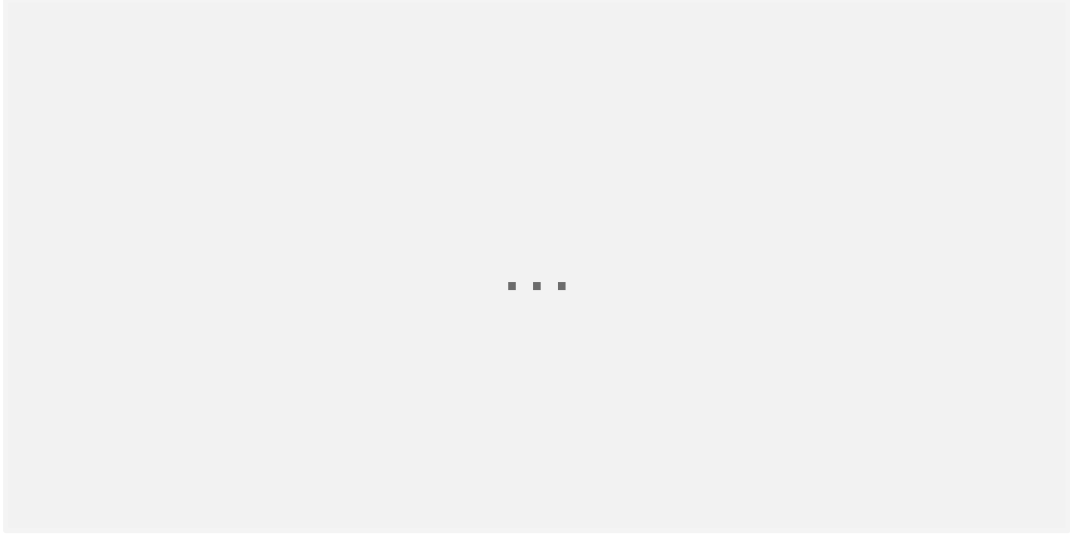
#Jugar sin cambiar
vsin = 0
```

```
def jugar(cambiar):
    ...
```

```python
def jugar(cambiar):
    #Colocar el auto
    auto = random.randint(0,2)

    #Colocar las cabras en las otras puertas
    if auto == 0:
        cabra1 = 1
        cabra2 = 2
    elif auto == 1:
        cabra1 = 0
        cabra2 = 2
    else:
        cabra1 = 0
        cabra2 = 1

    #El participante elige
    eleccion = random.randint(0,2)
```

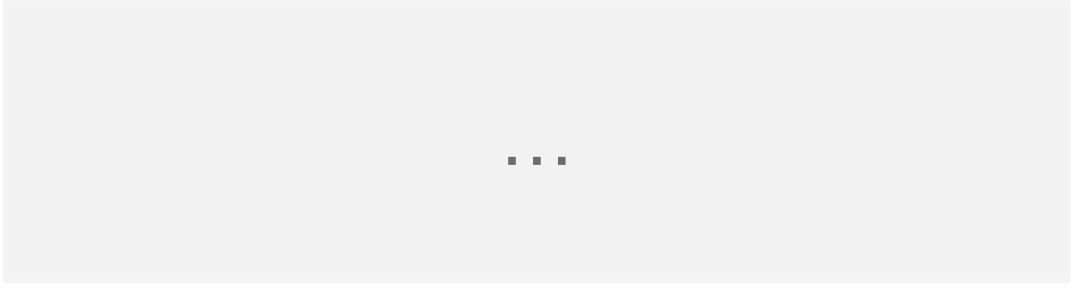


```
def jugar(cambiar):
    #Colocar el auto
    auto = random.randint(0,2)

    #Colocar las cabras en las otras puertas
    if auto == 0:
        cabra1 = 1
        cabra2 = 2
    elif auto == 1:
        cabra1 = 0
        cabra2 = 2
    else:
        cabra1 = 0
        cabra2 = 1

    #El participante elige
    eleccion = random.randint(0,2)

    #Calcular cual es la puerta alternativa que queda despues
    #de elegir una cabra...
    if (eleccion == cabra1) or (eleccion == cabra2):
        #...abres la otra cabra, y la alternativa es la puerta del auto
        alternativa = auto
    else:
        #Si elije el auto, abrimos la cabra1 por ejemplo, y solo queda cabra2
        alternativa = cabra2
```



```
def jugar(cambiar):
    #Colocar el auto
    auto = random.randint(0,2)

    #Colocar las cabras en las otras puertas
    if auto == 0:
        cabra1 = 1
        cabra2 = 2
    elif auto == 1:
        cabra1 = 0
        cabra2 = 2
    else:
        cabra1 = 0
        cabra2 = 1

    #El participante elige
    eleccion = random.randint(0,2)

    #Calcular cual es la puerta alternativa que queda despues
    #de elegir una cabra...
    if (eleccion == cabra1) or (eleccion == cabra2):
        #...abres la otra cabra, y la alternativa es la puerta del auto
        alternativa = auto
    else:
        #Si elije el auto, abrimos la cabra1 por ejemplo, y solo queda cabra2
        alternativa = cabra2

    #Cambia su eleccion o no dependiendo del parametro
    if cambiar == True:
        eleccion = alternativa

    #Miramos si ha ganado
    ganado = (eleccion == auto)

    #Devolvemos si ha ganado o no
    return ganado
```

EL GATO

- J2 pierde un turno
- J1 gana

```
EMPIEZA EL JUEGO

  |  |
-+-+-
  |  |
-+-+-
  |  |

Turno Jugador 1
Fila? 1
Columna? 1

  |  |
-+-+-
  |▫|
-+-+-
  |  |

Turno Jugador 2
Fila? 0
Columna? 2

  |  |⚑
-+-+-
  |▫|
-+-+-
  |  |
```

```
Turno Jugador 1
Fila? 0
Columna? 0

▫|  |⚑
-+-+-
  |▫|
-+-+-
  |  |

Turno Jugador 2
Fila? 1
Columna? 1
Casilla Ocupada! Persiste el turno

▫|  |⚑
-+-+-
  |▫|
-+-+-
  |  |

Turno Jugador 1
Fila? 2
Columna? 2
Ganaste Jugador 1

▫|  |⚑
-+-+-
  |▫|
-+-+-
  |  |▫

ACABA EL JUEGO
```

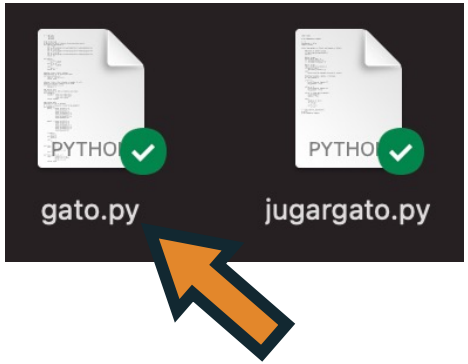
```python
#Recibe 2 ints: fila y columna
#Retorna bool si la celda esta libre
def esta_libre(f, c):

#Recibe 3 ints: fila, columna, y jugador (1 o 2)
#Pone la ficha del jugador en la celda
def poner_ficha(f, c, j):

#No Recibe nada
#Retorna bool si todo el tablero esta lleno
def ocupado():

#No recibe nada
#Retorna int con el ganador:
# 1 (Jug.1), 2(Jug.2), 0 (Aun no hay ganador)
def ganador():

# No recibe nada
# Retorna str con el tablero listo para hacer print
def tablero_imprimible():
```



```
import gato

print("EMPIEZA EL JUEGO")

j = 1
hay_ganador = False
empate = False

while (hay_ganador == False) and (empate == False):
```



```
t = gato.tablero_imprimible()
print(t)
print("ACABA EL JUEGO")
```

```
import gato

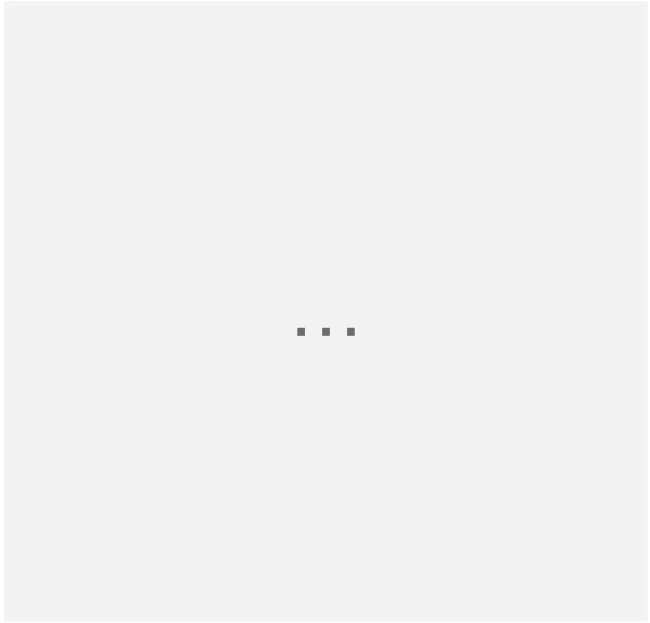
print("EMPIEZA EL JUEGO")

j = 1
hay_ganador = False
empate = False

while (hay_ganador == False) and (empate == False):

    #Imprimir el tablero actual
    t = gato.tablero_imprimible()
    print(t)

    #Pedir tirada
    print("Turno JUG", j)
    f = int(input("Fila? "))
    c = int(input("Columna? "))
```



```
t = gato.tablero_imprimible()
print(t)
print("ACABA EL JUEGO")
```

```python
import gato

print("EMPIEZA EL JUEGO")

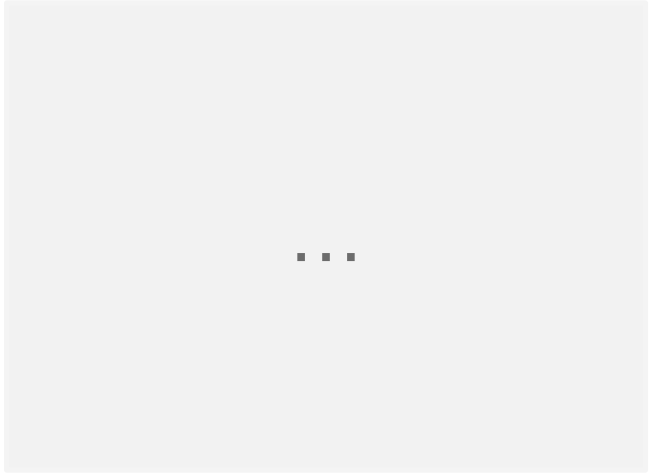
j = 1
hay_ganador = False
empate = False

while (hay_ganador == False) and (empate == False):

    #Imprimir el tablero actual
    t = gato.tablero_imprimible()
    print(t)

    #Pedir tirada
    print("Turno JUG", j)
    f = int(input("Fila? "))
    c = int(input("Columna? "))

    #Hacer tirada
    libre = gato.esta_libre(f,c)
    if libre == True:
        gato.poner_ficha(f,c,j)
    else:
        print("Casilla Ocupada! Persiste el turno")
```



```python
t = gato.tablero_imprimible()
print(t)
print("ACABA EL JUEGO")
```

```python
import gato

print("EMPIEZA EL JUEGO")

j = 1
hay_ganador = False
empate = False

while (hay_ganador == False) and (empate == False):

    #Imprimir el tablero actual
    t = gato.tablero_imprimible()
    print(t)

    #Pedir tirada
    print("Turno JUG", j)
    f = int(input("Fila? "))
    c = int(input("Columna? "))

    #Hacer tirada
    libre = gato.esta_libre(f,c)
    if libre == True:
        gato.poner_ficha(f,c,j)
    else:
        print("Casilla Ocupada! Persiste el turno")

    #Comprobar ganador, empate, o continuar
    g = gato.ganador()

    if g == 1:
        print("Ganaste Jugador 1")
        hay_ganador = True

    elif g == 2:
        print("Ganaste Jugador 2")
        hay_ganador = True

    elif g == 0 and gato.ocupado():
        print("Empate")
        empate = True

    else:
        #Cambiar el turno
        if j == 1:
            j = 2
        elif j == 2:
            j = 1

t = gato.tablero_imprimible()
print(t)
print("ACABA EL JUEGO")
```