

WALTER GIOVANNY CUADROS RINCÓN

## EJERCICIO DE CONDICIONALES

Para condicionales hice un pequeño juego

```
from random import randint, uniform, random
numero = randint(0,100)
print('BIENVENIDO TRATA DE ADIVINAR EL NÚMERO')
isWin = False
count = 1
while isWin == False:
    print('Por favor digita un número')
    a = int(input())
    if a > numero:
        print('Digita un número más pequeño')
    elif a < numero:
        print('Digita un número más grande')
    elif a == numero:
        isWin = True
        print('GANASTE')

    count = count+1

print('TU VISTE: '+str(count)+' intentos')
```

## EJERCICIO PAGOS MENSUALES CON DICCIONARIO

Para este ejercicio en lugar de pagos mensuales de recibos, lo hice con un listado de repuestos, donde el usuario ingresa todo el listado de repuestos y después puede escoger que repuestos sumar

```

print('VAMOS A CREAR NUESTRO LISTADO DE REPUESTOS')
repuestos = {}
isOk = True
while isOk:
    print('Ingresa el nombre del repuesto')
    nombre = input()
    print('Ingresa el precio del repuesto')
    precio = int(input())
    repuestos[nombre]=precio
    print('Oprima 0 si desea agregar otro repuesto, o cualquier número si desea salir')
    a = int(input())
    isOk = isContinue(a)

isOk = True
my_list = []
while isOk:
    print('Ingresa el nombre que deseas escoger')
    nombre = input()
    my_list.append(nombre)
    print('Oprima 0 si desea escoger más repuestos de la lista, o cualquier número si desea salir')
    a = int(input())
    isOk = isContinue(a)

pago_total = 0
print(my_list)
for key,value in repuestos.items():
    print(key)
    if key in my_list:
        pago_total=pago_total+value

print('El valor total a pagar es:'+str(pago_total))

def isContinue(a):
    if a>0:
        return False
    else:
        return True

```

## EJEMPLOS DE FOR

```

my_list= my_lista=['WALTER','YORLEY','LUCAS','PEDRO','GIOVANNY']
print('Resultado iterar lista')
for i in my_list:
    print(i)
print('Resultado iterar lista con range')
for i in range(len(my_list)):
    print(i)

the_dictionary ={
    'matematicas':100,
    'español':94,
    'ciencias':50,
    'informatioca':100,
    'fisica':95
}

print('Resultado iterar diccionario')
for key in the_dictionary.keys():
    print(key)

print('Resultado iterar lista con range')
for value in the_dictionary.values():
    print(value)

print('Resultado iterar lista con range')
for key in the_dictionary.keys():
    print(key,',',the_dictionary.get(key))

```

## CLASES, FUNCIONES Y DECORADORES

### Ejemplos de clases

```
class Address():
    def __init__(self, street=None, town=None, city=None):
        self.street = street
        self.town = town
        self.city = city

    def get_full_address(self):
        return '{} {} {}'.format(self.street, self.town, self.city)

class Persona():
    def __init__(self, first_name=None, last_name=None, age=None, id=None, address=None):
        self.first_name = first_name
        self.last_name = last_name
        self.age = age
        self.id = id
        self.address = address

    def set_last_name(self, last_name):
        self.last_name = last_name

    def saludar(cls):
        return 'Hola mi nombre es: {full_name}, tengo {age} años, y mi numero id es: {id}'.format(
            full_name='{} {}'.format(cls.first_name, cls.last_name),
            id=cls.id,
            age=cls.age,
        )

    def saludar_con_direccion(cls):
        inicio_saludo = cls.saludar()
        return inicio_saludo + ' mi direccion es: {}'.format(cls.address.get_full_address())

address = Address('calle 27', 'La Cumbre', 'Floridablanca')
walter = Persona(
    first_name='Walter',
    age='27',
    last_name='Cuadros',
    id=1090090,
    address=address
)
```

## EJEMPLOS DE FUNCIONES

```
def saluda(nombre_persona):
    print('HOLA {nombre_persona}'.format(nombre_persona=nombre_persona))

def promedio(data):
    mean = sum(data)/len(data)
    print(mean)

lista=[1,2,3,4,5,6,7,8,9]
nombre_persona = 'Walter Cuadros'
print('FUNCION SALUDA')
saluda(nombre_persona)
print('Funcion promedio')
promedio(lista)

personas = [ {
    'nombre':'Walter',
    'edad':'27',
    'profesion':'Ingeniero Electrónico'
},
{
    'nombre':'Yorley',
    'edad':'25',
    'profesion':'docente'
}
]
def saludar_personas(personas):
    nombre = personas.get('nombre')
    edad = personas.get('edad')
    profesion = personas.get('profesion')
    return 'Hola mi nombre es: {nombre}, tengo {edad}, profesion {profesion}'.format(nombre=nombre,edad=edad,profesion=profesion)

for persona in personas:
    print(saludar_personas(persona))
```

## EJERCICIO CON DECORADORES

```
C:\Users\diplomado_django\clase27_03>python decoradores.py
DIVISION EN CERO
ocurrio un error
PARAMETRO EN NULL
ocurrio un error
None
ocurrio un error
None
OMITIENDO PARAMETRO EN DICCIONARIO
ocurrio un error
None
ocurrio un error
None
PARAMETRO INDEX
ocurrio un error
```

```

1 def prevet_error(f):
2     def wrapper(*args, **kwargs):
3         try:
4             f(*args, **kwargs)
5         except:
6             print('ocurrio un error')
7     return wrapper
8
9
10 @prevet_error
11 def dummy_code(nota_1, nota_2):
12     prom = nota_1 + nota_2 / nota_2
13     print(prom)
14     print('DIVISION EN CERO')
15     dummy_code(5, 0)
16
17 @prevet_error
18 def saludar_personas(personas):
19     nombre = personas.get('nombre')
20     edad = personas['edad']
21     profesion = personas.get('profesion')
22     return 'Hola mi nombre es: {nombre}, tengo {edad}, profesion {profesion}'.format(nombre=nombre, edad=edad, profesion=profesion)
23
24 personas = [
25     {
26         'nombre': 'Walter',
27         'edad': '27'
28     },
29     {
30         'nombre': 'Vorley',
31         'edad': '25'
32     }
33 ]
34
35 print('PARAMETRO EN NULL')
36 for persona in personas:
37     print(saludar_personas(None))
38
39 print('OMITIENDO PARAMETRO EN DICCIONARIO')
40 for persona in personas:
41     print(saludar_personas(personas))
42
43 @prevet_error
44 def imprimir_lista(lista):
45     print(lista[5])
46
47 print('PARAMETRO INDEX')
48 lista= ['Walter', 'Giovanny']
49 imprimir_lista(lista)

```

REPOSITORIO ACTUALIZADO

http: [https://github.com/wagiodev/diplomado\\_python\\_django.git](https://github.com/wagiodev/diplomado_python_django.git)

SSH: git@github.com:wagiodev/diplomado\_python\_django.git