

▼ Ejercicios de Funciones

▼ Ejercicio 1

Escribir una función que muestre por pantalla el saludo ¡Hola amiga! cada vez que se la invoque.

```
def saludo():  
    print("¡Hola amiga!")  
  
saludo()  
  
    ¡Hola amiga!
```

▼ Ejercicio 2

Escribir una función a la que se le pase una cadena <nombre> y muestre por pantalla el saludo ¡hola <nombre>!.

```
def saludo(n):  
    print(f'¡Hola {n}!')  
  
nombre = input("Ingrese un nombre para el saludo : ")  
saludo(nombre)  
  
    Ingrese un nombre para el saludo : Felipe  
    ¡Hola Felipe!
```

▼ Ejercicio 3

Escribir una función que reciba un número entero positivo y devuelva su factorial.

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)
```

```

n = False
while not n:
    numero = int(input("Ingrese un numero entero positivo: "))
    n = numero > 0
    if not n:
        print("Error, solo puede ingresar numeros positivos")

factorial(numero)

```

```

    Ingrese un numero entero positivo: 4
    24

```

▼ Ejercicio 4

Escribir una función que calcule el total de una factura tras aplicarle el IVA. La función debe recibir la cantidad sin IVA y el porcentaje de IVA a aplicar, y devolver el total de la factura. Si se invoca la función sin pasarle el porcentaje de IVA, deberá aplicar un 21%.

```

def calcularIva(total, iva_percent=21):
    return total*(1+iva_percent/100)

calcularIva(14300)

```

```

    17303.0

```

▼ Ejercicio 5

Escribir una función que calcule el área de un círculo y otra que calcule el volumen de un cilindro usando la primera función.

```

import numpy as np

def areaCirculo(r):
    return 2*np.pi*r**2

radio=2
altura=6
print(f"El volumen de un cilindro de radio 5 y altura 10: {areaCirculo(radio)*altura}")

```

```

    El volumen de un cilindro de radio 5 y altura 10: 150.79644737231007

```

▼ Ejercicio 6

Escribir una función que reciba una muestra de números en una lista y devuelva su media.

```
import numpy as np

def calcularMean(data):
    return np.mean(data)

misdatos=[1,5222,2,2,67,2,4,34,12,3,5,2,5,23,24,34,2]

muestra = np.array(misdatos)
print(f"La media de la muestra es: {calcularMean(muestra)}")

La media de la muestra es: 320.2352941176471
```

▼ Ejercicio 7

Escribir una función que reciba una muestra de números en una lista y devuelva otra lista con sus cuadrados.

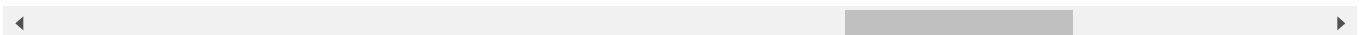
```
from multiprocessing import Pool
import numpy as np

def squared(x):
    return x**5

pool = Pool(4)
datos = np.arange(2**4,2**6)

result_1 = list(pool.map(squared,datos))
print(result_1)
```

1345007, 254803968, 282475249, 312500000, 345025251, 380204032, 418195493, 459165024, 503



▼ Ejercicio 8

Escribir una función que reciba una muestra de números en una lista y devuelva un diccionario con su media, varianza y desviación típica.

```
def dictMedia(datos):
    return dict({'media':np.mean(datos),
                'std':np.std(datos)})

display(dictMedia(datos))
type(dictMedia(datos))
```

▼ Ejercicio 9

Escribir una función que calcule el máximo común divisor de dos números y otra que calcule el mínimo común múltiplo.

```
def MCD(x,y):
    return np.gcd(x,y)

def MCM(x,y):
    return x*y/MCD(x,y)

display(MCD(13,22))
display(MCM(5,22))
```

▼ Ejercicio 10

Escribir una función que convierta un número decimal en binario y otra que convierta un número binario en decimal.

```
def decimal_a_binario(decimal):
    return print(int(bin(decimal)[2:]))

def binario_a_decimal(binario):
    return print(int(str(binario),2))

resp = str(input("Que tipo de numero decaea ingresar [Binario - Decimal]: "))

if resp.lower() == "binario":
    binario = int(input("Ingresa el numero: "))
```

```
    binario_a_decimal(binario)
elif resp.lower() == "decimal":
    decimal = int(input("Ingrese el numero: "))
    decimal_a_binario(decimal)
```

```
Que tipo de numero decea ingresar [Binario - Decimal]: binario
Ingrese el numero: 10101011
171
```

▼ Ejercicio 11

Escribir un programa que reciba una cadena de caracteres y devuelva un diccionario con cada palabra que contiene y su frecuencia. Escribir otra función que reciba el diccionario generado con la función anterior y devuelva una tupla con la palabra más repetida y su frecuencia.

```
def get_tupla(palabra):
    palabras = palabra.split()
    diccionario= {}

    for p in palabras:
        diccionario[p]=palabras.count(p)

    maximo = max(diccionario, key = lambda key: diccionario[key])

    for k, v in diccionario.items():
        if k == maximo:
            tupla = (k, v)

    return print(tupla)

resp = str(input("Ingrese una cadena de caracteres: "))
get_tupla(resp)

Ingrese una cadena de caracteres: Felipeeee
('Felipeeee', 1)
```

✓ 4 s completado a las 18:22

● ×