

# Cuaderno con ejercicios (con soluciones) del tema *Librerías*.

Este cuaderno forma parte del curso de [Iniciación a la programación con Python](#) del programa de Open Course Ware (OCW) de la Universidad de La Laguna.

En este cuaderno encontrarás una serie de ejercicios con su enunciado, seguido de celdas de código con las soluciones. Pueden existir múltiples soluciones válidas en programación, por lo que las soluciones proporcionadas pueden variar entre personas.

Puedes acceder al cuaderno sin las soluciones pichando en [este enlace](#).

## Ejercicio 1

Utilizando la librería `math`, calcular el área de un círculo dado su radio.

```
import math
radio = 6
print("El área del círculo es:",math.pi*radio**2)
```

## Ejercicio 2

Utilizando la librería `math`, definir una función que devuelva la raíz cuadrada de un número.

```
import math
def cuadrada(num):
    return math.sqrt(num)
x=34
print("La raíz cuadrada de",x,"es:",cuadrada(x))
```

## Ejercicio 3

Utilizando la función del ejercicio anterior, calcular las raíces cuadradas de 5 números enteros generados de manera aleatoria entre 12 y 120, utilizando la librería `random`.

```
import math, random
def cuadrada(num):
    return math.sqrt(num)
for i in range(0,5):
```

```
x=random.randint(12, 120)
print("La raíz cuadrada de",x,"es:",cuadrada(x))
```

## Ejercicio 4

Definir una función que reciba dos argumentos y permita calcular el seno, coseno o la tangente de un ángulo. El primero de los argumentos consiste en una cadena de texto que determina que operación realizar, y el segundo consiste en el ángulo en grados, que se debe convertir a radianes.

Utilizar la librería `math`.

```
import math

def funciones(operacion, angulo):
    resultado = math.radians(angulo)
    if operacion == "seno":
        return math.sin(resultado)
    elif operacion == "coseno":
        return math.cos(resultado)
    elif operacion == "tangente":
        return math.tan(resultado)
    else:
        return "La operacion introducida no es válida."

x = 45
print("El seno del ángulo",x,"es:",funciones("seno",x))
print("El coseno del ángulo",x,"es:",funciones("coseno",x))
print("La tangente del ángulo",x,"es:",funciones("tangente",x))
```

## Ejercicio 5

Utilizando la librería `datetime`, calcular la diferencia en años, meses y días entre dos fechas.

```
import datetime

fecha_1 = datetime.datetime(2020,12,20)
fecha_2 = datetime.datetime(2023,1,23)

df = fecha_2 - fecha_1
años = df.days // 365
días = df.days % 365
meses = días // 30 # Se tiene en cuenta que todos los meses son de 30 días
días = días % 30

print("Han pasado",años,"años,",meses,"meses y",días,"días")
```

## Ejercicio 6

Utilizando la librería `numpy`, realizar el producto escalar de dos vectores.

```
import numpy

vector_1 = numpy.array([2,4,6])
vector_2 = numpy.array([3,6,9])

print("El producto escalar de los dos vectores es",
numpy.dot(vector_1,vector_2))
```

## Ejercicio 7

Utilizando la librería `numpy`, resolver un sistema de ecuaciones lineal. Por ejemplo:

$$7x + 8y = 23.5$$

$$2.4x - 2y = 17$$

```
import numpy

ec_1 = numpy.array([[7, 8], [2.4, -2]])
ec_2 = numpy.array([23.5, 17])

solucion = numpy.linalg.solve(ec_1, ec_2)

print("La x vale:", solucion[0])
print("La y vale:", solucion[1])
```

## Ejercicio 8

Utilizando la librería `pandas`, calcular la calificación media de unas estudiantes.

Los datos serían:

```
nombres = ["María", "Mercedes", "Miriam", "Mónica"]
calificaciones = [[2,4.5,9],[5,3,10],[7.5,8.3,9],[4,6.5,7]]

import pandas

nombres = ["María", "Mercedes", "Miriam", "Mónica"]
calificaciones = [[2,4.5,9],[5,3,10],[7.5,8.3,9],[4,6.5,7]]

df = pandas.DataFrame({'Nombre': nombres, 'Calificaciones':
calificaciones})

medias = []
for x in df['Calificaciones']:
    medias.append(sum(x)/len(x))
```

```
df['Media'] = medias

print("Resultado:")
print(df[['Nombre', 'Media']])
```

## Ejercicio 9

Teniendo en cuenta el ejercicio anterior, mostrar en un gráfico las calificaciones medias usando `matplotlib`, en un gráfico de barras con etiquetas de datos en cada una de las barras.

```
import pandas, matplotlib.pyplot as plt

nombres = ["María", "Mercedes", "Miriam", "Mónica"]
calificaciones = [[2,4.5,9],[5,3,10],[7.5,8.3,9],[4,6.5,7]]

df = pandas.DataFrame({'Nombre': nombres, 'Calificaciones':
calificaciones})

medias = []
for x in df['Calificaciones']:
    medias.append(sum(x)/len(x))

df['Media'] = medias
cnt=0

plt.bar(df['Nombre'], df['Media'], color='darkorchid')
plt.xlabel('Estudiantes')
plt.ylabel('Medias')
plt.title('Medias de calificaciones por estudiante')
for i in df['Media']:
    plt.text(cnt, i, f'{i:.4f}', ha='center', va='bottom')
    cnt+=1
plt.show()
```

## Ejercicio 10

Utilizando la función de la raíz cuadrada del ejercicio 2, y las librerías `math`, `random` y `matplotlib`, generar un gráfico de dispersión en el que se muestre 20 números enteros aleatorios entre el 0 y 100 en el eje x, y su raíz cuadrada en el eje y.

```
import random, math, matplotlib.pyplot as plt

def cuadrada(num):
    return math.sqrt(num)

numeros = []
raices = []

for i in range(0,20):
```

```
x=random.randint(0, 100)
numeros.append(x)
raices.append(cuadrada(x))

plt.scatter(numeros, raices, color='red')
plt.xlabel('Número')
plt.ylabel('Raíz cuadrada')
plt.title('Gráfico de dispersión de números y sus raíces')

plt.show()
```