

Boletín 4

Ejercicio 01

Código

```
C: > DAVID > CICLO DAM > DI > DI- > Ejercicios04 > Ejercicio01.py > ...
1  ''' Realizar un programa que inicialice una lista con 10 valores aleatorios (del 1 al 10) y
2  posteriormente muestre en pantalla cada elemento de la lista junto con su cuadrado y su
3  cubo '''
4
5  import random
6
7  numeros = []
8
9  #Se inicializa la lista con valores aleatorios
10 for i in range(10):
11     numeros.append(random.randint(1, 10))
12
13 #Se muestra el cuadrado y el cubo de cada elemento de la lista
14 for i in range(10):
15     print("=====")
16     print("El cuadrado de", str(numeros[i]), "es", str(numeros[i]*2))
17     print("El cubo de", str(numeros[i]), "es", str(pow(numeros[i],3)))
```

Ejecución

```
=====
El cuadrado de 7 es 14
El cubo de 7 es 343
=====
El cuadrado de 3 es 6
El cubo de 3 es 27
=====
El cuadrado de 9 es 18
El cubo de 9 es 729
=====
El cuadrado de 3 es 6
El cubo de 3 es 27
=====
El cuadrado de 5 es 10
El cubo de 5 es 125
=====
El cuadrado de 5 es 10
El cubo de 5 es 125
=====
El cuadrado de 5 es 10
El cubo de 5 es 125
=====
El cuadrado de 6 es 12
El cubo de 6 es 216
=====
El cuadrado de 1 es 2
El cubo de 1 es 1
=====
El cuadrado de 10 es 20
El cubo de 10 es 1000
```

Ejercicio 02

Código

```
C: > DAVID > CICLO DAM > DI > DI- > Ejercicios04 > Ejercicio02.py > ...
1  '''Realizar un programa que lea por teclado las 5 notas obtenidas por un alumno (comprendidas
2  entre 0 y 10). A continuación, debe mostrar: todas las notas, la nota media, la nota más alta
3  que ha sacado y la menor.'''
4
5  numNotas = 5
6  notas = []
7  notaMedia = 0
8
9  #Metodo que pide la nota al usuario.
10 #En caso de no introducir un int, se lanza una excepcion
11 def introducirNota():
12     try:
13         nota = int(input("Introduce una nota: "))
14         while comprobarNota(nota) == False:
15             nota = int(input("Introduce una nota: "))
16         notas.append(nota)
17     except ValueError:
18         print("Solo se permiten datos numericos")
19         introducirNota()
20
21 #Metodo que comprueba que la nota introducida este entre el 0 y el 10
22 def comprobarNota(nota):
23     correcto = True
24     if nota < 0 or nota > 10:
25         correcto = False
26         print("La nota no puede ser mayor que 10 ni menor que 0")
27     return correcto
28
29 #Se usa un bucle for para introducir las 5 notas
30 for i in range(numNotas):
31     introducirNota()
32
33 #Se muestran las notas y se aprovecha el bucle for para saber la nota media
34 print("Las notas son:")
35 for i in range(5):
36     notaMedia += notas[i]
37     print(notas[i])
38 print("La nota media es:", notaMedia / 5)
39
40 #Se ordena la lista para saber la nota mas alta y la mas baja
41 notas.sort()
42
43 #Se muestran la nota mas alta y la nota mas baja
44 print("La nota mas alta es:", notas[4])
45 print("La nota mas baja es:", notas[0])
```

Ejecución

```
Introduce una nota: hola
Solo se permiten datos numericos
Introduce una nota: 23
La nota no puede ser mayor que 10 ni menor que 0
Introduce una nota: 10
Introduce una nota: 1
Introduce una nota: 5
Introduce una nota: 3
Introduce una nota: 8
Las notas son:
10
1
5
3
8
La nota media es: 5.4
La nota mas alta es: 10
La nota mas baja es: 1
```

Ejercicio 03

Código

```
C:\> DAVID > CICLO DAM > DI > DI- > Ejercicios04 > Ejercicio03

1  """Queremos guardar los nombres y las edades de los alumnos de un curso. Realiza un programa
2  que introduzca el nombre y la edad de cada alumno. El proceso de lectura de datos terminará
3  cuando se introduzca como nombre un asterisco (*) Al finalizar se mostrará los siguientes
4  datos:
5  - Todos los alumnos mayores de edad.
6  - Los alumnos mayores (los que tienen más edad)"""
7
8  #Se declaran dos listas, una con nombres y otra con edades de los alumnos
9  nombres = []
10  edades = []
11
12  #Metodo que comprueba que solo se introduzcan letras
13  #Si se introduce un * se finaliza el programa
14  def comprobarNombre(nombre):
15      correcto = False
16      if nombre == "*":
17          correcto = True
18      elif nombre.isalpha():
19          correcto = True
20      else:
21          print("Un nombre solo puede contener letras")
22      return correcto
23
24  #Metodo que comprueba que la edad sea de 1 a 120 años
25  def comprobarEdad(edad):
26      correcto = True
27      if edad < 1 or edad > 120:
28          correcto = False
29          print("La edad no puede ser mayor que 120 ni menor que 1")
30      return correcto
31
32  #Metodo que pide el nombre del alumno, si es correcto, lo añade a la lista de nombres
33  def introducirNombre():
34      nombre = input("Introduce un nombre: ")
35      while comprobarNombre(nombre) == False:
36          nombre = input("Introduce un nombre: ")
37      nombres.append(nombre)
38      return nombre
39
40  #Metodo que pida la edad del alumno, si es correcta, la añade a la lista de edades
41  #Se envuelve en un try/except por si el usuario no introduce un int
42  def introducirEdad():
43      try:
44          edad = int(input("Introduce una edad: "))
45          while comprobarEdad(edad) == False:
46              edad = int(input("Introduce una edad: "))
47          edades.append(edad)
48      except ValueError:
49          print("Solo se permiten datos numericos")
50          introducirEdad()
51
52  #Metodo que imprime los alumnos mayor de edad y los alumnos mas mayores
53  def imprimirDatos():
54      edadMasAlta = 0
55      print("Alumno/s mayor/es de edad:")
56      for i in range (len(nombres)):
57          if edadMasAlta < edades[i]:
58              edadMasAlta = edades[i]
59          if (edades[i] >= 18):
60              print(nombres[i])
61
62      print("Alumno/s mas mayor/es: ")
63      for i in range (len(nombres)):
64          if edadMasAlta == edades[i]:
65              print(nombres[i])
66
67  #Se pide el nombre y la edad de los alumnos. Si se introduce un * se finaliza el programa
68  while introducirNombre() != "*":
69      introducirEdad()
70  nombres.pop() #para eliminar el último elemento (*)
71  imprimirDatos()
```

Ejecución

```
Introduce un nombre: 23
Un nombre solo puede contener letras
Introduce un nombre: AlumnoUno
Introduce una edad: hola
Solo se permiten datos numericos
Introduce una edad: 23
Introduce un nombre: AlumnoDos
Introduce una edad: 12
Introduce un nombre: AlumnoTres
Introduce una edad: 0
La edad no puede ser mayor que 120 ni menor que 1
Introduce una edad: 48
Introduce un nombre: AlumnoCuatro
Introduce una edad: 18
Introduce un nombre: AlumnoCinco
Introduce una edad: 19
Introduce un nombre: *
Alumno/s mayor/es de edad:
AlumnoUno
AlumnoTres
AlumnoCuatro
AlumnoCinco
Alumno/s mas mayor/es:
AlumnoTres
```

Ejercicio 04

Código

```
1  '''Diseñar el algoritmo correspondiente a un programa, que:
2  - Crea una tabla (lista con dos dimensiones) de 5x5 enteros.
3  - Carga la tabla con valores numéricos enteros.
4  - Suma todos los elementos de cada fila y todos los elementos de cada columna
5  visualizando los resultados en pantalla'''
6
7  import random
8
9  filas = 5
10 contador = 0
11 tabla = []
12 #Se crea una tabla bidimensional de 5x5 con numeros aleatorios
13 for i in range(filas):
14     tabla.append([random.randint(1, 9), random.randint(1, 9), random.randint(1, 9),
15                  random.randint(1, 9), random.randint(1, 9)])
16
17 #Se imprime la tabla
18 for i in range(filas):
19     print(tabla[i])
20
21 #Se suman las filas
22 for i in range(filas):
23     for j in range(filas):
24         contador += tabla[i][j]
25     print("La suma de la fila", i + 1, "es", contador)
26     contador = 0
27
28 #Se suman las columnas
29 for i in range(filas):
30     for j in range(filas):
31         contador += tabla[j][i]
32     print("La suma de la columna", i + 1, "es", contador)
33     contador = 0
34
```

Ejecución

```
[9, 5, 1, 1, 8]
[4, 8, 1, 8, 2]
[1, 9, 8, 5, 4]
[5, 5, 4, 8, 5]
[6, 5, 4, 5, 7]
La suma de la fila 1 es 24
La suma de la fila 2 es 23
La suma de la fila 3 es 27
La suma de la fila 4 es 27
La suma de la fila 5 es 27
La suma de la columna 1 es 25
La suma de la columna 2 es 32
La suma de la columna 3 es 18
La suma de la columna 4 es 27
La suma de la columna 5 es 26
```