

Boletín 4: Funciones

En este boletín afrontamos un primer paso hacia la *modularización* de los programas. Se indique explícitamente o no, se debe diseñar en todos los casos una solución modular al problema planteado y añadir el código necesario para que el programa sea plenamente funcional.

1. Escribe un programa que haga uso de dos funciones para calcular, por una parte, el perímetro de una circunferencia de radio proporcionado por el usuario y, por la otra, la superficie del círculo correspondiente.
2. Crea un programa que defina una lista de números y determine la media, el máximo y el mínimo de dichos números.
3. Diseña una función que, dada una lista de cadenas, devuelva otra lista que sólo contenga las cadenas de mayor longitud.
4. Diseña una función que, dados dos números enteros a y b , calcule el *productorio* de todos los números comprendidos entre a y b . Debe tenerse en cuenta la posibilidad de que a sea mayor que b ; la función devolverá un 0 en ese caso. Por otro lado, si el 0 se encuentra entre a y b , la función determinará el resultado sin necesidad de iterar.
5. Escribe un programa que, a partir de un NIF proporcionado por el usuario, verifique que la letra es coherente con el número.
6. Diseña una función que reciba una pregunta, la muestre por pantalla y solicite una respuesta al usuario. Se aceptarán como únicas respuestas válidas 'sí', 's', 'Sí', 'SÍ', 'no', 'n', 'No' y 'NO', y habrá que informar adecuadamente al usuario en caso de que cometa un error. La función devolverá **True** en caso de que la respuesta sea afirmativa y **False** en caso contrario.
7. Implementa un procedimiento tal que, dado un número entero, muestre por pantalla sus cifras en orden inverso.
8. Se dispone de los tiempos de cada ciclista y cada etapa para los participantes en la última Volta da Ascensión. En este programa se manejarán los nombres de los ciclistas por medio de una lista y los tiempos invertidos (en segundos) en cada una de las cinco etapas por medio de una matriz, de la siguiente manera:

```
ciclistas = ['Álvaro Pino', 'Blanco Villar', 'Óscar Pereiro']
tiempo = [[10092.0, 12473.1, 13732.3, 10232.1, 10332.3],
          [11726.2, 11161.2, 12272.1, 11292.0, 12534.0],
          [10193.4, 10292.1, 11712.9, 10133.4, 11632.0]]
```

Se pide: (a) una función que reciba la lista y la matriz y devuelva el ganador de la Volta; (b) una función que reciba la lista, la matriz y un número de etapa y devuelva el nombre del ganador de la etapa; y (c) un procedimiento que reciba la lista y la matriz y que muestre el ganador de cada una de las etapas.

9. Diseña una función que reciba los tres coeficientes de una ecuación de segundo grado y devuelva una lista con sus soluciones reales. Si la ecuación sólo tiene una solución real, devolverá una lista con dos copias de la misma. Si no tiene solución real o si tiene infinitas soluciones, devolverá una lista con dos copias del valor **None**.
10. Diseña una función que reciba una lista de palabras y devuelva la primera y la última palabra según el orden alfabético.
11. Crea un programa que, partiendo de dos listas, sea capaz de obtener el conjunto de elementos comunes a ambas (intersección de conjuntos), el conjunto de elementos que aparecen en alguna de las dos (unión de conjuntos) y el conjunto de elementos que pertenecen a la primera pero no a la segunda (diferencia de conjuntos).

12. Escribe un programa que, dada una lista de números, obtenga otra lista que contenga sólo los números primos presentes en la primera. La determinación de si un número es o no primo debe encapsularse en una función aparte.
13. Diseña un procedimiento capaz de modificar una lista de números para que sólo sobrevivan los números que sean perfectos.
14. Diseña una función que reciba una lista y ordene sus elementos de menor a mayor haciendo uso del método de la burbuja. Escribe una segunda función que reciba una lista y devuelva otra que contenga los mismos elementos ordenados de forma creciente, sin que la lista original sufra ninguna alteración.
15. Diseña una función que reciba una matriz y, si es cuadrada, devuelva la suma de todos los elementos de la diagonal principal. Asegúrate de que la función devuelva un valor *apropiado* en caso de que la matriz no sea cuadrada.