

## Practica Python 5. Funciones

### Nota

- ✓ Muchos de los ejercicios llevan un ejemplo de interacción con el programa, una ejecución en la que se introduce una entrada y el programa proporciona una salida. La entrada que introduce el usuario aparece subrayada.

### Ejercicios

**1.-** Escribir un programa que pida un número entre 0 y 400, a continuación debe calcular todos los factoriales entre el cero y el número solicitado. El programa debe tener una función factorial que realice el cálculo. Ejemplo:

```
Introduce un número: 5
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
```

**2.-** Escribir un programa que solicite un número n y a continuación imprima todos los números primos comprendidos en el intervalo [2-n]. El programa debe tener una función `es_primo` que dado un número devuelva si el número es o no primo. Ejemplo:

```
Introduzca un número: 100
Números primos [2-100]: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 y 97
```

2.b) Haciendo uso de la función `es_primo`. Diseña a continuación un procedimiento `muestra_primos` que reciba un número y muestre por pantalla todos los números primos entre 1 y dicho número.

**3.-** Escribe un programa que lea una cadena de texto, acto seguido debe comprobar que se trata de un número de NIF correcto, 8 dígitos y una letra. El programa debe contar con la función `es_nif` que devuelva si el NIF es o no correcto

La función debe calcular la letra según el parámetro pasado, calcular la letra correspondiente a un NIF es bastante sencillo, hay que realizar la división entera del número del NIF entre 23, se toma el resto y se asigna una letra según la siguiente tabla:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

**4.-** Escribe un programa que lea un NIF sin letra por teclado, a continuación debe mostrar el NIF con la letra asociada. El programa debe contar con la función `letra_nif` que devuelve la letra correspondiente.

**5 .-** Escribe una función que dado el día, el mes y el año, devuelva un entero comprendido entre 1 y 366 indicando el número de día del año. La función tendrá esta declaración:

```
dia_agno(dia, mes, agno)
```

Ten en cuenta que un año es bisiesto si es divisible por 4 y no divisible por 100, excepto si es también divisible por 400, en cuyo caso es bisiesto.

(Ejemplos: El número de días de 2002 es 365: el número 2002 no es divisible por 4, así que no es bisiesto. El año 2004 es bisiesto y tiene 366 días: el número 2004 es divisible por 4, pero no por 100, así que es bisiesto. El año 1900 es divisible por 4, pero no es bisiesto porque es divisible por 100 y no por 400. El año 2000 sí es bisiesto: el número 2000 es divisible por 4 y, aunque es divisible por 100, también lo es por 400).

6.- Diseña una función llamada *es\_repetición* que reciba una cadena y nos diga si la cadena está formada mediante la concatenación de una cadena consigo misma. Por ejemplo, *es\_repetición('abab')* devolverá **True**, pues la cadena está formada con la cadena 'ab' repetida; por contra *es\_repetición('ababab')* devolverá **False**.

7.- Diseña una función que, dada una lista de cadenas, devuelva una lista con todas las cadenas más largas, es decir, si dos o más cadenas miden lo mismo y son las más largas, la lista las contendrá a todas. (Ejemplo: dada la lista ["Pepe", "Ana", "Juan", "Paz"], la función devolverá la lista de dos elementos ["Pepe", "Juan"]).

8.-

Supón que mantenemos dos listas con igual número de elementos. Una de ellas, llamada *alumnos*, contiene una serie de nombres y la otra, llamada *notas*, una serie de números flotantes entre 0.0 y 10.0. En *notas* guardamos la calificación obtenida por los alumnos cuyos nombres están en *alumnos*: la nota *notas[i]* corresponde al estudiante *alumnos[i]*. Una posible configuración de las listas sería esta:

```
1 alumnos = ['Ana_Pi', 'Pau_López', 'Luis_Sol', 'Mar_Vega', 'Paz_Mir']
2 notas    = [10, 5.5, 2.0, 8.5, 7.0]
```

De acuerdo con ella, el alumno Pau López, por ejemplo, fue calificado con un 5.5.

Nos piden diseñar un procedimiento que recibe como datos las dos listas y una cadena con el nombre de un estudiante. Si el estudiante pertenece a la clase, el procedimiento imprimirá su nombre y nota en pantalla. Si no es un alumno incluido en la lista, se imprimirá un mensaje que lo advierta.

```
clase.py
1 def muestra_nota_de_alumno(alumnos, notas, alumno_buscado):
2     for i in range(len(alumnos)):
3         if alumnos[i] == alumno_buscado:
4             print(alumno_buscado, notas[i])
5             return
6     print('El alumno {0} no pertenece al grupo'.format(alumno_buscado))
```

Realizar las siguientes funciones:

- 1) Diseñar una función que reciba las dos listas y que devuelva el nombre de todos los estudiantes que aprobaron el examen
- 2) Diseñar una función que reciba la lista de notas y devuelva el número de aprobados
- 3) Diseñar una función que reciba las dos listas y devuelva el nombre de todos los estudiantes que obtuvieron la máxima nota.
- 4) Diseñar una función que reciba las dos listas y devuelva el nombre de todos los estudiantes cuya calificación es igual o superior a la calificación media.
- 5) Diseñar una función que reciba las dos listas y un nombre (una cadena); si el nombre está en la lista de estudiantes, devolverá su nota, si no, devolverá None.

Haciendo uso de las funciones anteriores y diseñando nuevas funciones si es necesario. Construir un programa que presente el siguiente menú y permita ejecutar las acciones correspondientes a cada opción:

- 1) Añadir estudiante y calificación
- 2) Mostrar lista de estudiantes con sus calificaciones
- 3) Mostrar estudiantes aprobados
- 4) Número de aprobados
- 5) Estudiantes con máxima nota
- 6) Estudiantes con nota mayor o igual a la media
- 7) Nota estudiante
- 8) Finalizar ejecución del programa

9.- Modificar la estructura utilizada en la resolución del ejercicio 8, para que sea más eficiente.

10.- Diseña una función que reciba una lista de números y la modifique duplicando el valor de cada uno de sus elementos. (Ejemplo: la lista [1, 2, 3] se convertirá en la lista [2, 4, 6]).

11.- Diseña una función que reciba una lista y devuelva otra lista cuyo contenido sea el resultado de concatenar la lista original consigo misma. La lista original no debe modificarse.

12.- Realizar una función que genere matrices. Esta función no tendrá parámetros. Haciendo uso del módulo random generará matrices de tamaño entre 2 y 6 tanto en filas como en columnas (pudiendo ser diferente). Luego el contenido de la matriz también se generará aleatoriamente con números entre 0 y 100.

13.- Diseña una función que reciba una matriz (generada a partir de la función del ejercicio anterior) y, si es cuadrada (es decir, tiene igual número de filas que de columnas), devuelva la suma de todos los componentes dispuestos en la diagonal principal (es decir, todos los elementos de la forma  $A_{i,i}$ ). Si la matriz no es cuadrada, la función devolverá **None**.

Existirá un bucle que irá mostrando las matrices generadas y el resultado, hasta que encuentre una que sea cuadrada y cuyo resultado no sea None.

14.- Realizar una función que realice las operaciones de suma, resta, multiplicación, división, valor absoluto y ecuaciones de segundo grado.

Tendrá un parámetro obligatorio que será el operador, un parámetro con valor por defecto con valor igual a 0, se le podrán mandar más operandos (para que haga las operaciones sobre más de un valor, puede ser una cantidad variable de parámetros) y para las operaciones de segundo grado se le pasarán mediante paso de parámetros con nombre (a, b, c – en el modo con diccionario).

La función devolverá el resultado.

15.- Modificar el ejercicio 8 (original) de tal forma que la opción 2 y 7 sean la misma función. Esta tendrá como parámetros obligatorios las listas de estudiantes y de notas.

Si no se le pasa ningún nombre de estudiante mostrará la lista de todos los estudiantes con sus calificaciones.

Si se le pasa el nombre de un estudiante la de ese estudiante, indicando que no existe (este puede hacer uso de la ya creada en el ejercicio).

16.- Modificar el ejercicio 8 (original) de tal forma que para las opciones 3, 4, 5, 6, 7 siempre se llame a la misma función. Donde se le pasarán los parámetros necesarios (pudiendo ser alguno o todos opcionales, según se estime) según la opción que se le ha pasado.