

EJERCICIOS CON FUNCIONES

1. Solicitar al usuario que ingrese su dirección email. Imprimir un mensaje indicando si la dirección es válida o no, valiéndose de una función para decidirlo. Una dirección se considerará válida si contiene el símbolo "@".
2. Solicitar números al usuario hasta que ingrese el cero. Por cada uno, mostrar la suma de sus dígitos (utilizando una función que realice dicha suma)
3. Solicitar números al usuario hasta que ingrese el cero. Por cada uno, mostrar la suma de sus dígitos. Al finalizar, mostrar la sumatoria de todos los números ingresados y la suma de sus dígitos. Reutilizar la misma función realizada en el ejercicio 2aaa
4. Solicitar al usuario un número entero y luego un dígito. Informar la cantidad de ocurrencias del dígito en el número, utilizando para ello una función que calcule la frecuencia.
5. Escribir un programa que pida números al usuario, mostrar el factorial de cada uno y, al finalizar, la cantidad total de números leídos en total. Utilizar una o más funciones, según sea necesario.
6. Escribir un programa que pida números positivos al usuario. Mostrar el número cuya sumatoria de dígitos fue mayor y la cantidad de números cuya sumatoria de dígitos fue menor que 10. Utilizar una o más funciones, según sea necesario.
7. El siguiente programa debería imprimir el número 2 si se le ingresan como valores $x=5$, $y=1$ pero en su lugar imprime 5. ¿Qué hay que corregir?
8. Escribir una función que, dado un número de DNI, retorne True si el número es válido y False si no lo es. Para que un número de DNI sea válido debe tener entre 7 y 8 dígitos.
9. Escribir una función que, dado un string, retorne la longitud de la última palabra. Se considera que las palabras están separadas por uno o más espacios. También podría haber espacios al principio o al final del string pasado por parámetro.
10. Escribir la función `titulo()`, la cual recibe un string y lo retorna convirtiendo la primera letra de cada palabra a mayúscula y las demás letras a minúscula, dejando inalterados los demás caracteres. Precondición: el separador de palabras es el espacio: " ". Agregar doctests con suficientes casos de prueba para validar que la función retorna el valor esperado ante distintos argumentos.
11. Realizar una función llamada **`area_rectangulo(base, altura)`** que devuelva el área del rectángulo a partir de una base y una altura. Calcular el área de un rectángulo de

15 de base y 10 de altura.

12. Realizar una función llamada **intermedio(a, b)** que a partir de dos números, devuelva su punto intermedio.
13. Realizar una función **separar(lista)** que tome una lista de números enteros y devuelva dos listas ordenadas. La primera con los números pares y la segunda con los números impares.
14. Crear una función que, a partir de un dato de entrada que sea en horas, nos informe cuántos minutos y cuántos segundos serían esas horas. Imprimir por pantalla dichos valores.
15. Crear una función que devuelva una lista con todos los números pares del 0 al 100 inclusive. Imprimir esa lista por pantalla.
16. Crear una función que, a partir de un mensaje, nos devuelva una lista con todos los números, si los hay, que aparecen en dicho mensaje.
17. Crear una función que devuelva True si los parámetros ingresados son todos números, False si hay al menos uno de los parámetros ingresados que no es un número, y None si ninguno de los parámetros ingresados es un número. Imprimir resultado por pantalla.
18. Crear una función que verifique si una palabra es un palíndromo o no. En caso de que lo sea devolver por pantalla "La palabra es un palíndromo", en caso contrario, devolver "La palabra no es un palíndromo".
19. Crear una función que calcule cuántos litros de nafta gasta un auto que consume 2 litros x 100km, en un viaje ida y vuelta MdP-Bue si la distancia es de 400km. Luego crear una función que, a partir de esos datos, devuelva cuánto significa eso en pesos si el litro de nafta está 60\$.
20. Crear un diccionario con 10 estudiantes y sus respectivas notas. Luego crear una función que nos informe los estudiantes aprobados (nota ≥ 7), los estudiantes desaprobados ($4 \leq \text{nota} < 7$) y los estudiantes aplazados ($0 \leq \text{nota} < 4$).