

# Instituto de formación técnica superior N° 18

## Diagramación Lógica - Ejercicios de programación

Profesor Bonini Juan Ignacio

Escribir funciones que reciban uno o más parámetros y siempre devuelvan un resultado.

Las funciones NO deberán recibir datos por teclado.

Todas las funciones se deberán ejecutar para evaluar su correcto funcionamiento.

### Ejercicios de funciones en pseudocódigo

1. Realizar una función que reciba tres (3) notas y nos retorne como resultado el promedio.
2. Realizar una función que reciba 4 notas de exámenes finales y devuelva la cantidad de alumnos aprobaron la materia.
3. Realizar una función que reciba dos números y nos indique cual de los números es el mayor.
4. Realizar una función que reciba 4 números y nos indique cuantos números son mayores a 5.
5. Realizar una función que determine si un número es divisible por 3 o por 5.
6. Realizar una función que determine si un año es bisiesto.
7. Realizar una función que determine si un número es par o impar.
8. Realizar una función que concatene 2 cadenas de texto.

## Ejercicios de funciones en python3

Se deberá crear funciones en python3 y ejecutarlas debajo de la declaración.

Aún no está definida la cantidad de ejercicios pero se espera que sean más de 40.

### Consideraciones

1. Las funciones deben recibir parámetros y retornar resultados.
2. Todas las funciones deberán tener un “return” para devolver el valor.
3. No se permite escribir print dentro de las funciones.
4. Solo se permite el uso de print a la hora de ejecutar la función para ver su resultado.
5. Crear una carpeta llamada ejercicios y su número de documento separado por un guión bajo (\_) debería quedar algo así: ejercicios\_33456789
  - 5.1. Crear un archivo de texto dentro de la carpeta creada anteriormente llamado información. En dicho archivo deberán escribir nombre y apellido completo y su email de contacto y usuario en discord.
  - 5.2. Crear un archivo por ejercicio
  - 5.3. El archivo se deberá llamar ejercicio y el número de ejercicio, debería quedar algo así: ejercicio\_1.py, ejercicio\_1\_2.py, ejercicio\_4\_1.py

## Ejercicios a resolver

1. Traducir los ejercicios realizados anteriormente con pseint, y arreglarlos teniendo en cuenta la corrección realizada anteriormente. Quien no haya recibido corrección deberá mirar los ejercicios resueltos de un compañero/a y hacerlos correctamente.
2. El dueño de un reconocido marketplace desde ahora en más “El cliente” nos contrata para realizar cambios y/o mejoras en su sistema. Nos pedirá crear nuevas funciones de python que serán implementadas por sus ingenieros. El cliente nos pide crear una función que
  - 2.1. Dado un precio de un producto y un porcentaje variable, nos devuelva el precio aumentado en dicho porcentaje. Por ejemplo, si el valor es \$100 y aumenta en un 15% nos debería devolver \$115
  - 2.2. Dado un precio de un producto, la cantidad de productos a entregar y la cantidad de productos que deberá pagar nos calcule el precio unitario de cada producto. Ya que están implementando promociones del tipo 3x2, 4x2, 2x1, etc y quieren saber a cuanto queda el precio por unidad.
  - 2.3. Dado un precio y su respectivo IVA, deberá calcular el precio total más IVA del producto. Por ejemplo, si el producto vale \$100 y el IVA es del 10,5% nos deberá devolver \$110,50
  - 2.4. Dado un producto devolver el nombre del mismo, el color y su precio.  
producto={"nombre": "Mesa", "color": "negro", "precio": "30000"}  
Por ejemplo, deberá devolver: "Mesa negro \$30000"
  - 2.5. Dado el nombre de un producto y una lista de productos se deberá buscar el producto en dicha lista y en caso de encontrar una o más coincidencias se deberá devolver los resultados como una nueva lista.
  - 2.6. Dado el color de un producto, su tipo y una lista de productos, se deberá filtrar todos los resultados que coincidan. Se deberá devolver los resultados como una nueva lista.
  - 2.7. Dado el color de un producto, su tipo y una lista de productos, se deberá retornar si existe o no un producto con esas características en stock.

- 2.8. Dado el ID de un producto y una lista de productos se deberá retornar si existe el producto.
- 2.9. Dadas las características de un producto y una lista de productos, se deberá devolver si existe o no en la lista de productos.

### **Base de datos de nuestro cliente**

productos = [

{ "id": 1, "nombre": "Camiseta", "color": "rojo", "tipo": "ropa", "precio": 100},

{ "id": 2, "nombre": "Mesa", "color": "blanco", "tipo": "mueble", "precio": 300},

{ "id": 3, "nombre": "Silla", "color": "negro", "tipo": "mueble", "precio": 500},

{ "id": 4, "nombre": "Pantalón", "color": "azul", "tipo": "ropa", "precio": 200},

{ "id": 5, "nombre": "Cama", "color": "blanco", "tipo": "mueble", "precio": 10},

]