

LO IDEAL PARA LA RESOLUCION DE ESOTS EJERCICIOS ES QUE NO SE USEN FUNCIONES PROPIAS DE PYTHON COMO, EL APPEND, POP, INSERT, MIN, MAX, ETC. ESTOS EJERCICIOS SON PARA APRENDER Y ESAS FUNCIONES TE FACILITAN EL TRABAJO, NO TE ENSEÑAN MUCHOS CONCEPTOS UTILES. LOS EJERCICIOS SON PERFECTAMENTE POSIBLES SIN UTILIZAR ESTAS FUNCIONES.

- 1) Diseña una función que tome como parámetro 2 números, y que devuelva una lista que contenga todos los números enteros entre estos 2 incluyendo ambos parámetros.

Ejemplo: los parámetros para mi función son 1 y 9, por lo tanto, mi función retornara: [1,2,3,4,5,6,7,8,9]

- 2) Escribir una función que tome como parámetro 2 números, y retorne una lista con todos los números pares entre estos, excluyendo a los parámetros.

Ejemplo: los parámetros son 4 y 9, por lo tanto, la función retornara: [6,8]

- 3) Escribir una función que tome 2 parámetros, el primero que reciba una cadena, y el segundo que reciba un carácter. La función tendrá que retornar la cantidad de veces que aparece ese carácter en esa cadena.

Ejemplo: si los parámetros son "Hola mi nombre es Sebastián" y "s", la función tendrá que retornar 3 ya que la s se encuentra 3 veces repetidas en mi cadena.

- 4) Elaborar una función que tome como parámetro 2 números, y retorne una lista con todos los números primos entre ese rango de números.

Ejemplo: mis parámetros son 4 y 45, la función tendrá que retornar: [5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43].

- 5) Elaborar una función que tome como parámetro una lista, y devuelva un bool que diga si en esa lista todos sus números son pares.
- 6) Elaborar una función que tome como parámetro una lista y devuelva un bool que diga si en esa lista todos sus números son primos.

A PARTIR DE AQUÍ, COMO LOS EJERCICIOS SON MAS COMPLEJOS LAS FUNCIONES ESTAN PERMITIDAS. SE PUEDE UTILIZAR FUNCIONES PROPIAS DE PYTHON ASI COMO FUNCIONES CREADAS POR USTEDES

- 1) Elaborar un programa que con un menú el usuario tenga la opción de ingresar una entre 6 opciones.

Un menú puede ser creado de esta manera:

```
while True:
    opcion = int(input("Ingrese una opcion"))
    if opcion == 1:
        pass
        #opcion 1
    elif opcion == 2:
        pass
        #opcion 2
    elif opcion == 3:
        pass
        #opcion 3
```

- a. Para cada opción se tendrá que resolver un punto de los realizados anteriormente.
Por ejemplo, si el usuario selecciona la opción 1 y programa pedirá 2 números, y en base a esos 2 números resolverá el ejercicio 1 de esta práctica (devolver y mostrar la lista de números entre ese rango).
O si por ejemplo el usuario selecciona la opción 3 el usuario tendrá que ingresar una cadena y un carácter y el programa tendrá que mostrar la cantidad de veces que aparece ese carácter en esa cadena.
Así para cada uno de los puntos.

En los ejercicios 5 y 6 el usuario tendrá que ingresar una lista, por lo que puede ser útil crear una función que cree y retorne una lista según la cantidad de caracteres y los elementos que el usuario quiera.

- 2) Hacer una calculadora, que tenga las siguientes operaciones.
 - a. Suma
 - b. Resta
 - c. Multiplicación
 - d. División
 - e. Potencia

El usuario podrá elegir qué operación hacer y con qué números.

En caso de que el usuario ingrese como opción "SALIR" la calculadora se cerrará.

A la calculadora se le puede agregar cualquier otra operación que quieran. Raíz, Modulo, etc.

3) Hacer un programa que gestione datos para una escuela.

El programa tiene que ser capaz de:

- a) Llevar un registro de todos los datos de alumnos de la escuela (Nombre, Apellido, fecha de nacimiento, DNI, Nombre de Tutor, registro de todas las notas, cantidad de faltas, cantidad de amonestaciones recibidas.

Recomendación: Para llevar un registro de estos datos se puede utilizar un diccionario estructurado de la siguiente manera:

```
{  
  "Alumnos" : [alumno1,alumno2,alumno3 ]  
}
```

Donde cada alumno es otro diccionario estructurado de la

siguiente forma:

```
{  
  "Nombre": nombre de alumno,  
  "Apellido" : apellido de alumno,  
  "DNI" : DNI de alumno  
  "Fecha de nacimiento", fecha de nacimiento de alumno,  
  "Tutor" : nombre y apellido de tutor,  
  "Notas" : todas las notas del alumno,  
  "Faltas" : cantidad de faltas,  
  "amonestaciones" : cantidad de amonestaciones  
  
}
```

En esta estructura:

```
Datos = {  
  "Alumnos" : [alumno1,alumno2,alumno3 ]  
}
```

Para acceder por ejemplo al número de DNI del tercer alumno podríamos hacer algo así:

```
Datos["Alumnos"][2]["DNI"]
```

Este es un ejemplo de estructura, se puede cambiar completamente o hacer algunos cambios sobre el para mejorar el orden (si lo consideran necesario)

- b) Mostrar los datos de cada alumno
- c) Modificar los datos de los alumnos
- d) Agregar alumnos
- e) Expulsar alumnos

RECOMEDACIONN GENERAL:

El programa es extenso, hacer por partes.

Llevará mucho tiempo, la paciencia es importante.

Internet es una gran ayuda.

La prolijidad es fundamental

Las funciones tendrán que recibir como parámetros los diccionarios que representan a los alumnos.