

## Desafíos - Funciones

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Desarrollo desafío:
  - El desafío se debe desarrollar de manera Individual.

### Instrucciones

A continuación se detallan variados problemas a desarrollar. Se le pide que implemente las funciones pedidas y al momento de subir su solución genere un comprimido `.zip` de la siguiente manera:

#### Desafio-funciones.zip

```
|—— letras.py  
|—— menu_banco.py
```

#### NOTA:

- Los ejercicios que piden funciones se evalúan llamando a la función directamente y comparando el resultado.
- Para tener la evaluación correcta del ejercicio, se considera el nombre de la función y el resultado.
- El Desafío - Menú de banco es Opcional.
- El símbolo `>>>` representa la línea de entrada de comandos de la terminal del intérprete de python, no de la terminal de su computador, para acceder a la terminal del intérprete (si python está instalado correctamente en su sistema), basta con que llame a python desde su terminal escribiendo `python` y apretando enter.

## 1. Letras

Genere un script llamado `letras.py` que contenga la definición de las funciones pedidas a continuación.

### 1.1 Concatenando letras

Crear una función llamada `gen` que reciba el número de letras a generar del abecedario y devuelva un `string` con todas las letras generadas concatenadas.

Ejemplo:

```
>>> gen(4)

"abcd"

>>> gen(10)

"abcdefghij"
```

## 1.2 Dibujando letras

**a)** Crear la función `letra_o(n)`, la cual recibe un número entero positivo que determina el tamaño de una letra "o" a ser dibujada en la terminal según el ejemplo:

```
>>> letra_o(5)
```

```
*****  
*      *  
*      *  
*      *  
*      *  
*****
```

**b).** Crear la función `letra_i(n)`, la cual recibe un número entero positivo que determina el tamaño de una letra "i" a ser dibujada en la terminal según el ejemplo:

```
>>> letra_i(5)
```

```
*****  
      *  
      *  
      *  
*****
```

**c)** Crear la función `letra_x(n)`, la cual recibe un número entero positivo que determina el tamaño de una letra "x" a ser dibujada en la terminal según el ejemplo:

```
>>> letra_x(5)
```

```
*      *  
*  *  *  
      *  
*  *  *  
*      *
```

**Nota:** El valor de *n* indica la cantidad de filas (líneas en la terminal) y columnas (espacios horizontales en la terminal) que debe tener la letra, además, en el caso que el usuario ingrese un número par como argumento a sus funciones, sume 1 a dicho valor de forma de siempre obtenga un número impar y así tener un dibujo simétrico.

## 2. Menú de banco (Opcional)

Crear el programa `menu_banco.py` que, opcionalmente, pueda recibir un argumento numérico correspondiente al saldo inicial de la cuenta. Este programa, al ser ejecutado, debe llamar a la función `mostrar_menu(saldo = x)` la cual debe mostrar un menú con las siguientes opciones:

```
Bienvenido al portal del Banco Amigo. Escoja una opción:  
  
1. Consultar saldo  
2. Hacer depósito  
3. Realizar giro  
4. Salir
```

El parámetro `saldo` es *opcional*, y corresponde al saldo inicial con el que inicia el programa al ejecutarse, por defecto debe ser `0`.

Dos ejemplos de la invocación al programa desde la terminal serían:

```
$ python menu_banco.py
```

y

```
$ python menu_banco.py 5000
```

El programa debe contar además con las siguientes funciones:

Estas funciones **no deben ser llamadas al ejecutar el programa**. El llamado hacia cada una se explica después en la sección de requerimientos.

```
depositar(saldo, cantidad)
```

Función que recibe los parámetros `saldo` (int) y `cantidad` (int). Debe retornar el nuevo `saldo`, correspondiente al `saldo` ingresado más la `cantidad`.

```
girar(saldo, cantidad)
```

Función que recibe los parámetros `saldo` (int) y `cantidad` (int). Debe validar que `cantidad` no exceda a `saldo`. Si es así, debe retornar `False`. En caso contrario, debe restar esta `cantidad` a `saldo`, y retornar el resultado.

## Requerimientos

1. Una vez que el usuario escoge una opción, se debe validar que las únicas opciones que se pueda ingresar son 1, 2, 3 o 4. Si se introduce otra opción, se debe mostrar el mensaje "Opción inválida. Por favor ingrese 1, 2, 3 ó 4.", y volver a solicitar una opción.
2. Las funciones `depositar()` y `girar()` no deben contener `print()`. Las salidas deben manejarse desde `mostrar_menu()`.
3. Al escoger la opción 1, se debe mostrar el saldo actual en pantalla. Luego se vuelve a mostrar el menú.
4. Al escoger la opción 2, se debe solicitar la cantidad a depositar, y con ella llamar a la función `depositar(saldo, cantidad)`. Luego se debe mostrar el nuevo saldo en pantalla, y volver a mostrar el menú.
5. Al escoger la opción 3, primero se debe validar que exista saldo (debe ser mayor a 0). Si no existe saldo, debe mostrar mensaje "No puede realizar giros. Su saldo es 0". Si existe saldo, se debe solicitar la cantidad a girar, y con ella llamar a la función `girar(saldo, cantidad)`.
  - Si la función retorna `False`, se debe mostrar el mensaje "No se puede girar esta cantidad. Su saldo es de " (concatenar el valor de `saldo`). Se debe solicitar nuevamente la cantidad a girar y volver a llamar a la función `girar(cantidad)`.

**Nota:** El operador de comparación `is` es más estricto que `==`. En Python, de manera general se considera el valor `0` como `False`. Si se desea comprobar que un objeto tiene **exactamente** el valor `False` (y no `0`), se debe usar `is`.

  - Si la función retorna un nuevo saldo (o sea, "no retorna `False`"), se debe mostrar el nuevo saldo en pantalla, y volver a mostrar el menú.
5. Solo se detiene la ejecución del programa al escoger la opción 4.
6. **El saldo debe conservarse**, de acuerdo a las operaciones que se realicen, durante una misma ejecución del programa.

## Ejemplo de flujo esperado

¡Bienvenido al Banco Amigo!. Escoja una opción:

1. Consultar saldo
2. Hacer depósito
3. Realizar giro
4. Salir

1

Su saldo es de 0

¡Bienvenido al Banco Amigo!. Escoja una opción:

1. Consultar saldo
2. Hacer depósito
3. Realizar giro
4. Salir

2

1000

Su nuevo saldo es de 1000

¡Bienvenido al Banco Amigo!. Escoja una opción:

1. Consultar saldo
2. Hacer depósito
3. Realizar giro
4. Salir

1

Su saldo es de 1000