



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Lenguaje de Programación Python

Actividad 7:

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 4 de Octubre del 2023

CO

RMBA_PY_ACT7.ipynb

☆

Comentario

Compartir

⚙

B

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaÚltima modificación: 2 de octubre

+ Código+ Texto

Conectar

ACTIVIDAD 7

1.-Función que utilice una lista con los nombres de tus mascotas, o artistas favoritos, (minimo 5, maximo 10) imprimir las cadenas y la cantidad de caracteres de cada cadena.

EJEMPLO SALIDA:

[0]SCARLETT -> 8 CARACTERES

[1]GATINA -> 6 CARACTERES

[2]PELUSA -> 6 CARACTERES

[3]PELUSINA -> 8 CARACTERES

[4]MILO -> 4 CARACTERES

[5]KHALIFA -> 7 CARACTERES

[] def lista():
 print("Lista de Artistas Favoritos: ")
 lista = ["DEPECHE MODE", "21 SAVEGE", "GHOST MANE", "50 CENT", "EMINEM", "POST MALONE","FUTURE"]
 for i,cadena in enumerate(lista):
 print(f"({i+1}): {cadena} -- TIENE {len(cadena)} CARACTERES")

 print(len(lista))

CO

RMBA_PY_ACT7.ipynb

☆

Comentario

Compartir

⚙

B

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaÚltima modificación: 2 de octubre

+ Código+ Texto

Conectar

[] def lista():
 print("Lista de Artistas Favoritos: ")
 lista = ["DEPECHE MODE", "21 SAVEGE", "GHOST MANE", "50 CENT", "EMINEM", "POST MALONE","FUTURE"]
 for i,cadena in enumerate(lista):
 print(f"({i+1}): {cadena} -- TIENE {len(cadena)} CARACTERES")

 print(len(lista))

lista()

Lista de Artistas Favoritos:
1: DEPECHE MODE -- TIENE 12 CARACTERES
2: 21 SAVEGE -- TIENE 9 CARACTERES
3: GHOST MANE -- TIENE 10 CARACTERES
4: 50 CENT -- TIENE 7 CARACTERES
5: EMINEM -- TIENE 6 CARACTERES
6: POST MALONE -- TIENE 11 CARACTERES
7: FUTURE -- TIENE 6 CARACTERES
7

2.-Programa que utilice 2 funciones,

Función que genere y regrese una lista con 10 números aleatorios entre el 30 y 50 (no repetidos). Función que reciba una lista , Imprimir la lista (Índice y Valor)

Mandar llamar la función e imprimir toda la lista.

[] from IPython.display import clear_output

CO

RMBA_PY_ACT7.ipynb

☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Última modificación: 2 de octubre

ComentarioCompartirB

+ Código

+ Texto

```
i=0
[ ] print("Esta es tu lista: ")
    for cadena in lista:
        print(f"{{i+1}}.- {{cadena}}")
        i=i+1
    input("Presiona enter para continuar")
    global op
    op=0

while True:
    clear_output()
    print(" M E N U")
    print("1. Aleatorios")
    print("2. Lista")
    print("3. Salir")
    op=validar_num("Que opcion eliges? ")
    if op==1:
        clear_output()
        aleatorio()
    elif op==2:
        clear_output()
        lista()
    elif op==3:
        break
    else:
        print("Opcion invalida, intente de nuevo")
        continue
```

M E N U

1. Aleatorios

2. Lista

CO

RMBA_PY_ACT7.ipynb ☆

Comentario

Compartir

⚙

B

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaÚltima modificación: 2 de octubre

+ Código+ Texto

Conectar

3.-Escribe una función que reciba dos listas de números del mismo tamaño y calcule la suma de los elementos correspondientes de cada lista. (regresar una nueva lista) . Luego, muestra una lista con los resultados de cada suma. Nota: si las listas no son del mismo tamaño mandar msge y utilizar el tamaño de la lista mas pequeña)

```
[ ] from IPython.display import clear_output
def sumlist(list1,list2):
    list1=[]
    list2=[]
    tam1=validar_num("Ingresa el tamaño de tu primer lista:")
    for i in range(tam1):
        num=validar_num(f"Ingresa el elemento {i+1} a tu lista: ")
        list1.append(num)
        clear_output()
    tam2=validar_num("Ingresa el tamaño de tu segunda lista: ")
    for i in range(tam2):
        num=validar_num(f"Ingresa el elemento {i+1} a tu lista: ")
        list2.append(num)
    if (tam1!=tam2):
        print("Las listas son de diferente tamaño, se utilizará la lista mas pequeña")
        tam=min(len(list1),len(list2))
    else:
        tam=len(list1)
    sumalistas=[]
    for i in range(tam):
        suma=list1[i]+list2[i]
        sumalistas.append(suma)
    clear_output()
    print(f"LISTA 1: {list1}")
```

CO

RMBA_PY_ACT7.ipynb ☆

Comentario

Compartir

⚙

B

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaÚltima modificación: 2 de octubre

+ Código+ Texto

Conectar

```
[ ] def sumlist(list1,list2):
    list1=[]
    list2=[]
    tam1=validar_num("Ingresa el tamaño de tu primer lista:")
    for i in range(tam1):
        num=validar_num(f"Ingresa el elemento {i+1} a tu lista: ")
        list1.append(num)
        clear_output()
    tam2=validar_num("Ingresa el tamaño de tu segunda lista: ")
    for i in range(tam2):
        num=validar_num(f"Ingresa el elemento {i+1} a tu lista: ")
        list2.append(num)
    if (tam1!=tam2):
        print("Las listas son de diferente tamaño, se utilizará la lista mas pequeña")
        tam=min(len(list1),len(list2))
    else:
        tam=len(list1)
    sumalistas=[]
    for i in range(tam):
        suma=list1[i]+list2[i]
        sumalistas.append(suma)
    clear_output()
    print(f"LISTA 1: {list1}")
    print(f"LISTA 2: {list2}")
    print(f"La suma de las listas es igual a: {sumalistas}")

list1=[]
list2=[]

sumlist(list1,list2)
```

CO

RMBA_PY_ACT7.ipynb

☆

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaÚltima modificación: 2 de octubre

ComentarioCompartirB

+ Código+ TextoConectar

4.-Escribe una función llamada eliminar_duplicados que reciba una lista como parámetro y elimine los elementos duplicados. El resultado debe ser una nueva lista sin duplicados. Pide al usuario que ingrese una lista y luego llama a la función eliminar_duplicados para mostrar la lista sin duplicados. (Solo Numeros enteros, validar)

```
def elimdups(lista):
    lista=[]
    tam=validar_num("Ingresa el tamaño de tu lista")
    for i in range(tam):
        num=validar_num(f"Ingresa el elemento {i+1} a tu lista: ")
        lista.append(num)
    nuevalista=[]
    for elemento in lista:
        if elemento not in nuevalista:
            nuevalista.append(elemento)
    print(f"Su lista es: {lista}")
    print(f"La nueva lista (sin duplicados) es: {nuevalista}")

elimdups(lista)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-10-d512d920b16c> in <cell line: 14>()
     12     print(f"La nueva lista (sin duplicados) es: {nuevalista}")
     13
--> 14 elimdups(lista)

<ipython-input-10-d512d920b16c> in elimdups(lista)
      1 def elimdups(lista):
```

CO

RMBA_PY_ACT7.ipynb

☆

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaÚltima modificación: 2 de octubre

ComentarioCompartirB

+ Código+ TextoConectar

5.-Escribe una función que calcule la media y la mediana de una lista de números enteros.

La media es la suma de todos los elementos y dividido entre la cantidad de elementos

La mediana es el valor que queda en la mitad de la lista cuando se ordena de forma ascendente. Si la lista tiene un número par de elementos, la mediana se calcula como el promedio de los dos valores centrales. Puedes utilizar el método sort() para ordenar la lista y luego calcular la mediana según el tamaño de la lista.

```
[ ] def medmed(lista):
    lista=[]
    tam=validar_num("Ingresa el tamaño de tu lista: ")
    for i in range(tam):
        num=validar_num(f"Ingresa el elemento {i+1} a tu lista: ")
        lista.append(num)
    tam=len(lista)
    listord=sorted(lista)
    media=sum(lista)/tam

    if tam%2==0:
        mediana=(listord[tam//2-1]+listord[tam//2])/2
    else:
        mediana=listord[tam//2]
    clear_output()
    print(f"La media de la lista es: {media}")
    print(f"La mediana de la lista es: {mediana}")

medmed(lista)
```



+ Código + Texto

Conectar ▾



6.-Escribe una función que calcule la media móvil de una lista de números enteros. La media móvil es el promedio de un número determinado de elementos consecutivos de la lista. Puedes utilizar unciclo y una variable de suma para calcular la media móvil de cada sublista y guardar los resultados en otra lista.

```
[ ] def mediamovil(lista,n):
    lista=[]
    tam=validar_num("Ingresa el tamaño de tu lista:")
    for i in range(tam):
        num=validar_num(f"Ingresa el elemento {i+1} de tu lista (entero): ")
        lista.append(num)
    n=validar_num("Ingresa el numero de elementos de los que quieres la media movil: ")
    listameds=[]
    for i in range(len(lista)-n+1):
        sum=0
        print(f"Sublista {i+1}: {lista[i:i+n]}")
        suma=sum(lista[i:i+n: ])
        print(f"La media movil de la sublista {i+1} es: {suma/n}")
        listameds.append(suma/n)
    return listameds
    print(f"Lista de media movil: {listameds}")
```

```
mediamovil(lista,n)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-12-2bfa0219dd54> in <cell line: 18>()
     16     print(f"Lista de media movil: {listameds}")
```

