

## Solución a los problemas del día 5:

### Soluciones a las Prácticas del Día 5

#### Soluciones a las Prácticas del Día 5

Antes de dar por finalizadas las prácticas del día de hoy y pasar al proyecto, te comparto las posibles soluciones para cada una de las prácticas que fueron propuestas el día de hoy, en caso de que hayas tenido alguna dificultad para terminarlas.

**¡Ten en cuenta que estas soluciones no son únicas!** Si has arribado por otro camino a un código que arroja un resultado correcto en PyCharm o tu editor de código de preferencia, quiero que sepas que has hecho un buen trabajo (no siempre pueden contemplarse todos los casos a la hora de evaluar una tarea).

#### Práctica Métodos y Ayuda 1

```
1. print(",:_#,,,,,:::____##Pyt%on_ _Total,,,,,::#".lstrip(",:%_#"))
```

#### Práctica Métodos y Ayuda 2

```
1. frutas = ["mango", "banana", "cereza", "ciruela", "pomelo"]
2. frutas.insert(3,"naranja")
```

#### Práctica Métodos y Ayuda 3

```
1. marcas_smartphones = {"Samsung", "Xiaomi", "Apple", "Huawei", "LG"}
2.
3. marcas_tv = {"Sony", "Philips", "Samsung", "LG"}
4.
5. conjuntos_aislados = marcas_tv.isdisjoint(marcas_smartphones)
```

#### Práctica Crear Funciones 1

```
1. def saludar():
2.     print("¡Hola mundo!")
```

#### Práctica Crear Funciones 2

```
1. nombre_persona = "Luis"
2.
3. def bienvenida(nombre_persona):
```

```
4.     print(f'¡Bienvenido {nombre_persona}!')
```

### Práctica Crear Funciones 3

```
1. un_numero = 5
2.
3. def cuadrado(un_numero):
4.     print(un_numero**2)
```

### Práctica Return 1

```
1. def potencia(num1, num2):
2.     return num1**num2
```

### Práctica Return 2

```
1. dolares = 1200
2.
3. def usd_a_eur(dolares):
4.     return dolares*0.90
```

### Práctica Return 3

```
1. palabra = "Curso de Python"
2.
3. def invertir_palabra(palabra):
4.     palabra = palabra[::-1]
5.     palabra = palabra.upper()
6.     return palabra
```

### Práctica Funciones Dinámicas 1

```
1. lista_numeros = [1,-50,502,-5000,755,600,33,61]
2.
3. def todos_positivos(lista_numeros):
4.     for numero in lista_numeros:
5.         if numero < 0:
6.             return False
7.         else:
8.             pass
9.     return True
```

### Práctica Funciones Dinámicas 2

```
1. lista_numeros = [1,50,500,5000,750,600]
2.
3. def suma_menores(lista_numeros):
4.     suma=0
5.     for numero in lista_numeros:
6.         if numero in range(1,1000):
7.             suma += numero
```

```
8.         else:
9.             pass
10.    return suma
```

### Práctica Funciones Dinámicas 3

```
1. lista_numeros = [1,50,502,5000,755,600,33,61]
2.
3. def cantidad_pares(lista_numeros):
4.     cantidad=0
5.     for numero in lista_numeros:
6.         if numero % 2 == 0:
7.             cantidad += 1
8.         else:
9.             pass
10.    return cantidad
```

### Práctica sobre Interacción entre Funciones 1

```
1. import random
2.
3. def lanzar_dados():
4.     return random.randint(1,6), random.randint(1,6)
5.
6. def evaluar_jugada(dado1, dado2):
7.     suma_dados = dado1 + dado2
8.     if suma_dados <= 6:
9.         return f"La suma de tus dados es {suma_dados}. Lamentable"
10.    elif suma_dados > 6 and suma_dados < 10:
11.        return f"La suma de tus dados es {suma_dados}. Tienes buenas
chances"
12.    else:
13.        return f"La suma de tus dados es {suma_dados}. Parece una jugada
ganadora"
```

### Práctica sobre Interacción entre Funciones 2

```
1. lista_numeros = [1,2,15,7,2,8]
2.
3. def reducir_lista(lista):
4.     lista = list(set(lista))
5.     lista.sort()
6.     lista.pop(-1)
7.     return lista
8.
9. def promedio(lista):
10.    valor_medio = sum(lista)/len(lista)
11.    return valor_medio
```

### Práctica sobre Interacción entre Funciones 3

```
1. lista_numeros = [1,2,15,7,2,8]
2.
```

```

3. import random
4.
5. def lanzar_moneda():
6.     resultado = random.choice(["Cara", "Cruz"])
7.     return resultado
8.
9. def probar_suerte(moneda, lista):
10.    if moneda == "Cara":
11.        print("La lista se autodestruirá")
12.        return []
13.    elif moneda == "Cruz":
14.        print("La lista fue salvada")
15.        return lista

```

### Práctica sobre Argumentos Indefinidos (\*args) 1

```

1. def suma_cuadrados(*args):
2.     suma = 0
3.     for arg in args:
4.         suma += arg**2
5.
6.     return suma

```

### Práctica sobre Argumentos Indefinidos (\*args) 2

```

1. def suma_absolutos(*args):
2.     suma = 0
3.     for arg in args:
4.         suma += abs(arg)
5.
6.     return suma

```

### Práctica sobre Argumentos Indefinidos (\*args) 3

```

1. def numeros_persona(nombre, *args):
2.     suma_numeros = sum(args)
3.     return f'{nombre}, la suma de tus números es {suma_numeros}'

```

### Práctica sobre Argumentos Indefinidos (\*kwargs) 1

```

1. def cantidad_atributos(**kwargs):
2.     cantidad = 0
3.     for clave in kwargs.items():
4.         cantidad += 1
5.     return cantidad

```

### Práctica sobre Argumentos Indefinidos (\*kwargs) 2

```

1. def lista_atributos(**kwargs):
2.     lista = []
3.     for valor in kwargs.values():
4.         lista.append(valor)

```

```
5.     return lista
```

### Práctica sobre Argumentos Indefinidos (\*kwargs) 3

```
1. def describir_persona(nombre, **kwargs):  
2.     print(f"Características de {nombre}:")  
3.     for clave, valor in kwargs.items():  
4.         print(f'{clave}: {valor}')
```

Recuerda que si tienes alguna duda que aún no quede resuelta, tienes a tu disposición el espacio de Preguntas y Respuestas.

Te veo en el proyecto!