20.09.2025

Fundamentos de Programación
Actividad 4
Diego Couto

Funcionamiento:

1. Tupla de Números

```
[null@archlinux actividad_4]$ python act_4.py

1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?: 1

numeros: (1, 2, 3, 4, 5)
tercer elemento: 3
Ingrese un numero para agregar: 2
Ingrese otro numero: 3

numeros: (1, 2, 3, 4, 5, 2, 3)
numeros ordenados: (1, 2, 2, 3, 3, 4, 5)
numeros sumados:  20


1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?:
```

2. Diccionario de contactos

```
1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?: 2

Contactos:
David : 442-432-1432
José : 442-923-7698
Sandra : 442-987-1004

Ingrese el nombre del nuevo contacto: Pedro
Ingrese el número telefónico de Pedro: 1234-1234-1234
David
José
Sandra
Pedro

Ingrese el nombre del contacto que busca: Sandra
Sandra:442-987-1004


1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?:
```

3. Excepciones

```
1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?: 3


Ingrese el dividendo (cantidad a dividir): 60
Ingrese el divisor (cantidad por la que se divide el dividendo): 0

El valor ingresado no puede ser igual a cero y debe de ser un número entero.
Intentelo de nuevo...

Ingrese el divisor (cantidad por la que se divide el dividendo): 7

Cociente: 8.571428571428571

1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?:
```

4. Strings

```
1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?: 4

Este es un mensaje :)
longitud del mensaje: 21 caracteres.

Mensaje en mayúsculas:
ESTE ES UN MENSAJE :)

Reemplazo de "mensaje" por "texto":
Este es un texto :)

string: Este es un mensaje diferente.
número de palabras: 5

1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?:
```

5. Salir

```
1. Tupla de Números
2. Diccionario de contactos
3. Excepciones
4. Strings
5. Salir

¿Qué opcion elije?: 5
[null@archlinux actividad_4]$
```

Código fuente:

```
#Uso de tuplas
def cap_tupla():
    #Crea una tupla llamada "numeros" que contenga, al menos, cinco números.
    numeros = (1,2,3,4,5)
```

```
    print(f"\nnumeros: {numeros}")
    #Accede al tercer elemento de la tupla e imprímelo en pantalla.
    print(f"tercer elemento: {numeros[2]}")
    #Por medio de una captura, agrega dos números adicionales a la tupla (emplea input).
    arreglo_numeros = list(numeros)
    arreglo_numeros.append(int(input('Ingrese un numero para agregar:' )))
    arreglo_numeros.append(int(input('Ingrese otro numero: ')))
    numeros = tuple(arreglo_numeros)
    print(f"\nnumeros: {numeros}")
    #Convierte la tupla en una lista y, después, ordena los elementos.
    arreglo_numeros = list(numeros)
    arreglo_numeros.sort()
    numeros = tuple(arreglo_numeros)
    print(f"numeros ordenados: {numeros}")

    #Crea una función que reciba la tupla de números como parámetro y retorne la suma de
todos los elementos.
    def sum_tupla(numeros):
        arreglo_numeros = list(numeros)
        sum = 0
        insisos = len(arreglo_numeros)
        for x in range(0, insisos):
            sum = sum + arreglo_numeros[x]
        return sum
    #Aplica la función creada en el paso anterior a la tupla y, luego, muestra el resultado.
    print('numeros sumados: ',sum_tupla(numeros),'\n')

#Uso de diccionarios
def cap_dic():
    #Crea un diccionario llamado "contactos" con, al menos, tres entradas, donde las claves
sean los nombres de las personas y los valores correspondan a sus números de teléfono.
    contactos = {
            "David": "442-432-1432",
            "José": "442-923-7698",
            "Sandra": "442-987-1004"
            }

    print("\nContactos: ")
    for x in contactos:
        print(f"{x} : {contactos[x]}")
    #Por medio de una captura, agrega un nuevo contacto al diccionario.
    nombre = input('\nIngrese el nombre del nuevo contacto: ')
    numero = input(f"Ingrese el número telefónico de {nombre}: ")
    contactos[nombre] = numero
```

```
    #Itera sobre las claves del diccionario e imprime en pantalla los nombres de los contactos.
    for contacto in contactos:
        print(contacto)
    #Crea una función que reciba el diccionario de contactos y un nombre como parámetro para que, posteriormente, retorne el número de teléfono correspondiente al contacto ingresado.
    def buscar_nombre(contactos):
        nombre_bus = input('\nIngrese el nombre del contacto que busca: ')
        return nombre_bus + ':' + contactos.get(nombre_bus)
    #Aplica la función creada en el paso anterior para buscar el número de teléfono de uno de los contactos y muestra el resultado.
    print(f"{buscar_nombre(contactos)}\n")

#Uso de excepciones
#Crea un programa que solicite al usuario que ingrese dos números enteros.
def excepciones():
#Utiliza un bloque try-except para manejar la excepción que ocurrirá si el usuario ingresa un valor no numérico.
    #bucle de captura del primer numero
    b = 1
    while b == 1:
        try:
            print('\n')
            n1 = int(input('Ingrese el dividendo (cantidad a dividir): '))
        except:
            print("\nEl valor ingresado debe de ser un número entero. \nIntentelo de nuevo...\n")
        else:
            b = 0
    #bucle de captura del segundo numero
    c = 1
    while c == 1:
    #Si el usuario ingresa valores no numéricos, muestra un mensaje de error; en cambio, si los valores son numéricos, muestra la suma de los dos números ingresados.
        try:
            n2 = int(input('Ingrese el divisor (cantidad por la que se divide el dividendo): '))
            #Agrega una excepción personalizada para manejar la división entre cero; por tanto, en caso de que el segundo número ingresado sea cero, muestra un mensaje de error adecuado.
            if n2 == 0:
                raise Exception("El divisor no puede ser 0")
                print("El divisor no puede ser 0")
        except:
            print("\nEl valor ingresado no puede ser igual a cero y debe de ser un número entero. \nIntentelo de nuevo...\n")

        else:
```

```
            c = 0

    cociente = n1 / n2
    print(f"\nCociente: {cociente}")

#Uso de strings
def strings():
    #Crea una variable llamada "mensaje" que contenga un texto de tu elección.
    mensaje = "Este es un mensaje :)"
    print(f"\n{mensaje}")
    #Imprime en pantalla la longitud del mensaje.
    print(f"longitud del mensaje: {len(mensaje)} caracteres.")
    #Utiliza un método de strings para convertir todo el texto del mensaje a mayúsculas.
    print(f"\nMensaje en mayúsculas: \n{mensaje.upper()}\n")
    #Utiliza otro método de strings para reemplazar una palabra específica en el mensaje por
otra de tu elección.
    print(f"Reemplazo de \"mensaje\" por \"texto\":\n{mensaje.replace("mensaje", "texto")}\n")
    #Crea una función que reciba un string como parámetro y retorne la cantidad de palabras
que contiene.
    def cantidad_palabras(texto):
        print(f"string: {texto}")
        palabras = texto.split()
        print(f"número de palabras: {len(palabras)}")
    #Aplica la función creada en el paso anterior al mensaje y muestra el resultado.
    cantidad_palabras('Este es un mensaje diferente.')

#Menú principal
a = 1
while a == 1:
    #Crea un menú principal que permita elegir qué tipo de dato se desea capturar.
    print('\n1. Tupla de Números')
    print('2. Diccionario de contactos')
    print('3. Excepciones')
    print('4. Strings')
    print('5. Salir\n')
    #De acuerdo con la elección hecha por el usuario, llama a la función indicada para proceder
a su ejecución.
    try:
        eleccion = int(input('¿Qué opcion elije?: '))
        if eleccion == 1:
            cap_tupla()
        if eleccion == 2:
            cap_dic()
        if eleccion == 3:
```

```
        excepciones()
    if eleccion == 4:
        strings()
    #Agrega una opción para finalizar la ejecución.
    if eleccion == 5:
        a = 0
except:
    print('Intentelo de nuevo')
```