Curso: Tópicos de Ciencia de Datos
Tarea: Investigación Unidad 1
Fecha de vencimiento: Saturday, 24 de February de 2024, 16:00

FRANCISCO JAVIER ATAXCA GOXCON
201U0223@alumno.itssat.edu.mx

Valor: 20

Cambiar usuario

1 de 6

Página 1 de 10

Entrega

Enviado para calificar
Calificado
La tarea fue enviada 1 día 19 horas antes
El estudiante puede editar esta entrega
InvestigacioinUnidadI-FranciscoAtaxca.pdf
Comentarios (0)

Calificación

Calificación:

Hoja de presentación	No contien todos los datos 0 puntos	Datos incompletos 0.5 puntos	Completo 1 puntos	
Introducción	No contiene 0 puntos	Muy pequeña 1.5 puntos	Completa 3 puntos	
Contenido	No cubre los temas 0 puntos	La mitad de los temas 6 puntos	Completo 11 puntos	
Referencias IEEE	No contiene 0 puntos	Una o no tiene el formato 1 puntos	Más de una y formato correcto 2 puntos	
Conclusión	No contien 0 puntos	Muy pequeña 1 puntos	Completa 2 puntos	
Archivo PDF	Sin formato 0 puntos	Correcto 1 puntos		

Calificación actual en el libro de calificaciones

20,00

Comentarios de retroalimentación

Notificar a los estudiantes Guardar cambios Reiniciar

Instituto Tecnológico Superior de San Andrés Tuxtla

Ingeniería Informática

Tópicos de ciencia de datos

Octavo Semestre

Alumno: Francisco Javier Ataxca Goxcon

Docente: M.T.I. Rogelio Enrique Telona Torres.

Introducción

En la siguiente información vamos a ver diferentes operaciones que podemos realizar mediante el lenguaje Python para análisis de datos, veremos parte teórica y algunos ejemplos de cómo se ejecuta cada operación que vamos a presentar, como; Operaciones que no modifican una lista, Operaciones que modifican una lista, Copia de listas, Operaciones que no modifican un diccionario entre otras.

Listas

Una lista es unas secuencias ordenadas de objetos de distintos tipos.

Se construyen poniendo los elementos entre corchetes [ ] separados por comas.

Se caracterizan por:

Tienen orden.

Pueden contener elementos de distintos tipos.

Son mutables, es decir, pueden alterarse durante la ejecución de un programa.

```
# Lista vacía
>>> type([])
<class 'list'>
# Lista con elementos de distintos tipos
>>> [1, "dos", True]
# Listas anidadas
>>> [1, [2, 3], 4]
```

Operaciones que no modifican una lista

len(l) : Devuelve el número de elementos de la lista l.

min(l) : Devuelve el mínimo elemento de la lista l siempre que los datos sean comparables.

max(l) : Devuelve el máximo elemento de la lista l siempre que los datos sean comparables.

sum(l) : Devuelve la suma de los elementos de la lista l, siempre que los datos se puedan sumar.

dato in l : Devuelve True si el dato dato pertenece a la lista l y False en caso contrario.

l.index(dato) : Devuelve la posición que ocupa en la lista l el primer elemento con valor dato.

l.count(dato) : Devuelve el número de veces que el valor dato está contenido en la lista l.

all(l) : Devuelve True si todos los elementos de la lista l son True y False en caso contrario.

any(l) : Devuelve True si algún elemento de la lista l es True y False en caso contrario.

```
>>> a = [1, 2, 2, 3]
>>> len(a)
4
>>> min(a)
1
>>> max(a)
3
>>> sum(a)
8
>>> 3 in a
True
>>> a.index(2)
1
>>> a.count(2)
2
>>> all(a)
True
>>> any([0, False, 3<2])
False
```

Operaciones que modifican una lista

l1 + l2 : Crea una nueva lista concatenan los elementos de la listas l1 y l2.

l.append(dato) : Añade dato al final de la lista l.

l.extend(sequencia) : Añade los datos de sequencia al final de la lista l.

l.insert(índice, dato) : Inserta dato en la posición índice de la lista l y desplaza los elementos una posición a partir de la posición índice.

l.remove(dato) : Elimina el primer elemento con valor dato en la lista l y desplaza los que están por detrás de él una posición hacia delante.

l.pop([índice]) : Devuelve el dato en la posición índice y lo elimina de la lista l, desplazando los elementos por detrás de él una posición hacia delante.

l.sort() : Ordena los elementos de la lista l de acuerdo al orden predefinido, siempre que los elementos sean comparables.

l.reverse() : invierte el orden de los elementos de la lista l.

```
>>> a = [1, 3]
>>> b = [2 , 4, 6]
>>> a.append(5)
>>> a
[1, 3, 5]
>>> a.remove(3)
>>> a
[1, 5]
>>> a.insert(1, 3)
>>> a
[1, 3, 5]
>>> b.pop()
6
>>> c = a + b
>>> c
[1, 3, 5, 2, 4]
>>> c.sort()
>>> c
[1, 2, 3, 4, 5]
>>> c.reverse()
>>> c
[5, 4, 3, 2, 1]
```

Copia de listas

Existen dos formas de copiar listas:

Copia por referencia l1 = l2: Asocia la la variable l1 la misma lista que tiene asociada la variable l2, es decir, ambas variables apuntan a la misma dirección de memoria. Cualquier cambio que hagamos a través de l1 o l2 afectará a la misma lista.

Copia por valor l1 = list(l2): Crea una copia de la lista asociada a l2 en una dirección de memoria diferente y se la asocia a l1. Las variables apuntan a direcciones de memoria diferentes que contienen los mismos datos. Cualquier cambio que hagamos a través de l1 no afectará a la lista de l2 y viceversa.

```
>>> a = [1, 2, 3]
>>> # copia por referencia
>>> b = a
>>> b
[1, 2, 3]
>>> b.remove(2)
>>> b
[1, 3]
>>> a
[1, 3]
```

```
>>> a = [1, 2, 3]
>>> # copia por referencia
>>> b = list(a)
>>> b
[1, 2, 3]
>>> b.remove(2)
>>> b
[1, 3]
>>> a
[1, 2, 3]
```

Diccionarios

Un diccionario es una colección de pares formados por una clave y un valor asociado a la clave.

Se construyen poniendo los pares entre llaves { } separados por comas, y separando la clave del valor con dos puntos :.

Se caracterizan por:

- No tienen orden.
- Pueden contener elementos de distintos tipos.
- Son mutables, es decir, pueden alterarse durante la ejecución de un programa.
- Las claves son únicas, es decir, no pueden repetirse en un mismo diccionario, y pueden ser de cualquier tipo de datos inmutable.

```
# Diccionario vacío
type({})
<class 'dict'>
# Diccionario con elementos de distintos tipos
{'nombre':'Alfredo', 'despacho': 218, 'email':'asalber@ceu.es'}
# Diccionarios anidados
{'nombre_completo':{'nombre': 'Alfredo', 'Apellidos': 'Sánchez Alberca'}}
```

Operaciones que no modifican un diccionario

- len(d) : Devuelve el número de elementos del diccionario d.
- min(d) : Devuelve la mínima clave del diccionario d siempre que las claves sean comparables.
- max(d) : Devuelve la máxima clave del diccionario d siempre que las claves sean comparables.
- sum(d) : Devuelve la suma de las claves del diccionario d, siempre que las claves se puedan sumar.
- clave in d : Devuelve True si la clave clave pertenece al diccionario d y False en caso contrario.
- d.keys() : Devuelve un iterador sobre las claves de un diccionario.
- d.values() : Devuelve un iterador sobre los valores de un diccionario.
- d.items() : Devuelve un iterador sobre los pares clave-valor de un diccionario.

```
>>> a = {'nombre':'Alfredo', 'despacho': 218, 'email':'asalber@ceu.es'}
>>> len(a)
3
>>> min(a)
'despacho'
>>> 'email' in a
True
>>> a.keys()
dict_keys(['nombre', 'despacho', 'email'])
>>> a.values()
dict_values(['Alfredo', 218, 'asalber@ceu.es'])
>>> a.items()
dict_items([('nombre', 'Alfredo'), ('despacho', 218), ('email', 'asalber@ceu.es')])
```

Operaciones que modifican un diccionario

- d[clave] = valor : Añade al diccionario d el par formado por la clave clave y el valor valor.
- d.update(d2). Añade los pares del diccionario d2 al diccionario d.
- d.pop(clave, alternativo) : Devuelve del valor asociado a la clave clave del diccionario d y lo elimina del diccionario. Si la clave no está devuelve el valor alternativo.
- d.popitem() : Devuelve la tupla formada por la clave y el valor del último par añadido al diccionario d y lo elimina del diccionario.
- del d[clave] : Elimina del diccionario d el par con la clave clave.
- d.clear() : Elimina todos los pares del diccionario d de manera que se queda vacío.

```
>>> a = {'nombre':'Alfredo', 'despacho': 218, 'email':'asalber@ceu.es'}
>>> a['universidad'] = 'CEU'
>>> a
{'nombre': 'Alfredo', 'despacho': 218, 'email': 'asalber@ceu.es', 'universidad': 'CEU
>>> a.pop('despacho')
218
>>> a
{'nombre': 'Alfredo', 'email': 'asalber@ceu.es', 'universidad': 'CEU'}
>>> a.popitem()
('universidad', 'CEU')
>>> a
{'nombre': 'Alfredo', 'email': 'asalber@ceu.es'}
>>> del a['email']
>>> a
{'nombre': 'Alfredo'}
>>> a.clear()
>>> a
{}
```

Copia de diccionarios

Existen dos formas de copiar diccionarios:

- **Copia por referencia** d1 = d2: Asocia la la variable d1 el mismo diccionario que tiene asociado la variable d2, es decir, ambas variables apuntan a la misma dirección de memoria. Cualquier cambio que hagamos a través de l1 o l2 afectará al mismo diccionario.
- **Copia por valor** d1 = list(d2): Crea una copia del diccionario asociado a d2 en una dirección de memoria diferente y se la asocia a d1. Las variables apuntan a direcciones de memoria diferentes que contienen los mismos datos. Cualquier cambio que hagamos a través de l1 no afectará al diccionario de l2 y viceversa.

```
>>> a = {1:'A', 2:'B', 3:'C'}
>>> # copia por referencia
>>> b = a
>>> b
{1:'A', 2:'B', 3:'C'}
>>> b.pop(2)
>>> b
{1:'A', 3:'C'}
>>> a
{1:'A', 3:'C'}
```

```
>>> a = {1:'A', 2:'B', 3:'C'}
>>> # copia por referencia
>>> b = dict(a)
>>> b
{1:'A', 2:'B', 3:'C'}
>>> b.pop(2)
>>> b
{1:'A', 3:'C'}
>>> a
{1:'A', 2:'B', 3:'C'}
```

Conclusión

Para terminar, considero que es importante conocer estos puntos ya que nos permite saber manejar datos de una forma adecuada. Y como bien sabemos el manejar los datos de una manera adecuada favorece y da resultados positivos en donde se aplique tal estudio. Los elementos anteriores son herramientas poderosas de Python.

Fuentes:

[1] Corporativo. Sep 21 2020. Manual de Python. Aprendeconalf. Disponible: https://aprendeconalf.es/docencia/python/manual/

[2] Rolando Cobon. Platzi. Listas en Python. Disponible: https://platzi.com/tutoriales/4227-python/24835-listas-en-python-y-sus-operaciones-basicas/

Curso: Tópicos de Ciencia de Datos
Tarea: Practicas Unidad 1
Fecha de vencimiento: Saturday, 24 de February de 2024, 21:00

FRANCISCO JAVIER ATAXCA GOXCON
201U0223@alumno.itssat.edu.mx

Valor: 40

Cambiar usuario

1 de 6

Página 1 de 13

Entrega

Enviado para calificar
Calificado
La tarea fue enviada 1 día 11 horas después
El estudiante puede editar esta entrega
PracticaUnidadI-FJAG.pdf
Comentarios (0)

Calificación

Calificación:

Practicas

Hoja de presentación	No contien todos los datos 0 puntos	Datos incompletos 1 puntos	Completo 2 puntos	
Indice	No contiene 0 puntos		Contiene 2 puntos	
Practicas	No contiene 0 puntos	Parcialmente 15 puntos	Todas 25 puntos	Extra clases 30 puntos
Conclusión	No contiene 0 puntos	Pequeña 3 puntos	Completa 6 puntos	

Calificación actual en el libro de calificaciones

40,00

Comentarios de retroalimentación

Notificar a los estudiantes

Guardar cambios

Reiniciar

Unidad1

Introduccion al lenguaje python

07/02/24

Frncisco Javier Ataxca Goxcon

funcion print()

funcion print sirve para mostrar una variable o texto en la pantalla

ejemplo1

In [ ]:
```
print ("HOLA MUNDO")
```

```
HOLA MUNDO
```

FUNCION input()

-inputraw es equivalente pero es version 2.

permite la lectura de datos. siempre es cadena por defecto

In [ ]:
```
# suma de 2 numeros, pedido por teclado
# la FEA
print("script que suma de 2 numeros dados por el teclado")
print("Escribe el valor de primer numero")
num1 = input()
# la forma CUQUIS
num2 = input ("Escribe el valor del segundo numero")
suma = num1+num2
print(num1, " + ",num2, " = ",suma)
```

```
script que suma de 2 numeros dados por el teclado
Escribe el valor de primer numero
  +    =
```

El codigo anterior funciona, pero en lugar de realizar la suma concatena. Para esto utlizamos la funcion

int()

La funcion int() que convierte lo que pase como parametro a numero entero

In [ ]:
```
# suma de 2 numeros, pedido por teclado
# la FEA
print("script que suma de 2 numeros dados por el teclado")
print("Escribe el valor de primer numero")
num1 = input()
# la forma CUQUIS
num2 = input ("Escribe el valor del segundo numero")
num1 = int(num1)
num2= int(num2)
suma = num1+num2
print(num1, " + ",num2, " = ",suma)
```

```
script que suma de 2 numeros dados por el teclado
Escribe el valor de primer numero
2  +  2  =  4
```

Mejorar el codigo anterior

In [ ]:
```
# suma de 2 numeros, pedido por teclado
num1 = int(input("Escribe el valor del primer numero"))
num2 = int(input("Escribe el valor del segundo numero"))
suma = num1+num2
print(num1, " + ",num2, " = ",suma)
```

```
5  +  5  =  10
```

Ejercicio1

Escribir un programa en el cual se ingresan cuatro números, calcular e informar la suma de los primeros y el producto del tercero y cuarto.

In [ ]:
```
# progrma que suma los dos primeros numeros y multiplica los ultimos
print ("PROGRAMA QUE SUMA LOS DOS NUMEROS Y MULTIPLICA LOS ULTIMOS")
numero1 = float(input("Ecribe el valor del primer numero"))
numero2 = float(input("Escriba el valor del segundo numero"))
numero3 = float(input("Ecribe el valor del tercer numero"))
numero4 = float(input("Escriba el valor del cuarto numero"))
print(f"{numero1} + {numero2} = {numero1+numero2}")
print(f"{numero3} + {numero4} = {numero4*numero4}")
```

```
PROGRAMA QUE SUMA LOS DOS NUMEROS Y MULTIPLICA LOS ULTIMOS
5.0 + 6.0 = 11.0
8.0 + 8.0 = 64.0
```

Python conditions and if statements

In [ ]:
```
numero1 = int(input("Escribe el valor del primer numero"))
numero2 = int(input("Escribe el valor del segundo numero"))
```

```
if numero1 == numero2:
    print(f"{numero1} y {numero2} son iguales")
elif numero2>numero2:
    print(f"{numero2} es mayor que {numero1}")
else:
     print(f"{numero1}es mayor que {numero2}")
```

```
33 y 33 son iguales
```

Ejercicio Dia de semana con if anidados

In [ ]:
```
numero = int(input("Escribe un Numero"))
if numero == 1:
    print("Es Lunes")
elif numero == 2:
    print("Es Martes")
elif numero == 3:
    print("Es Miercoles")
elif numero == 4:
    print("Es Juevez")
elif numero == 5:
    print("Es Viernes")
elif numero == 6:
    print("Es Sabado")
elif numero == 7:
    print("Es Domingo")
else:
    print("Día no valido")
```

```
Día no valido
```

Ejercicio con Match

In [ ]:
```
dia = int(input("Escribe un Numero"))
match dia:
    case 1:
        print("Lunes")
    case 2:
        print("Martes")
    case 3:
        print("Miercoles")
    case 4:
        print("Juevez")
    case 5:
        print("Viernez")
    case 6:
        print("Sabado")
    case 1:
        print("Domingo")
    case _:
         print("Error")
```

```
Martes
```

Ejercicio de menu operaciones

In [ ]:
```
print ("Menu de Operaciones")
Operaciones= int(input("Indica la operacion a desplegar"))
match Operaciones:
    case 1:
        print("Suma")
        numero1= int (input("Ingrese el primer numero: "))
        numero2= int (input("Ingrese el segundo numero: "))
        print(f"{numero1} + {numero2} = {numero1+numero2}")
    case 2:
        print("Resta")
        numero1= int (input("Ingrese el primer numero: "))
        numero2= int (input("Ingrese el segundo numero: "))
        print(f"{numero1} - {numero2} = {numero1-numero2}")
    case 3:
        print("Multiplicacion")
        numero1= int (input("Ingrese el primer numero: "))
        numero2= int (input("Ingrese el segundo numero: "))
        print(f"{numero1} * {numero2} = {numero1*numero2}")
    case 4:
        print("Division")
        numero1= int (input("Ingrese el primer numero: "))
        numero2= int (input("Ingrese el segundo numero: "))
        print(f"{numero1} / {numero2} = {numero1/numero2}")
    case 5:
        print("Porcentaje")
        numero1= int (input("Ingrese el primer numero: "))
        numero2= int (input("Ingrese el segundo numero: "))
        print(f"{numero1} % {numero2} = {numero1*numero2/100}")
    case 6:
        print("Exponente")
        numero1= int (input("Ingrese el primer numero: "))
        numero2= int (input("Ingrese el segundo numero: "))
        print(f"{numero1} ** {numero2} = {numero1**numero2}")
    case 7:
        print("Division Compleja")
        numero1= int (input("Ingrese el primer numero: "))
        numero2= int (input("Ingrese el segundo numero: "))
        print(f"{numero1} // {numero2} = {numero1//numero2}")
    case _:
        print("Eror")
```

```
Menu de Operaciones Aritmeticas
Suma
2 + 2 = 4
```

Ciclos

For (Para)

El ciclo for es usado para interar sonbre una secuencia

Ejenplo

In [ ]:
```
# Ciclo for que imprime del  1 al 10
for i in range(10):
    print(i)
```

La funcion range() retorna una secuencia de numero iniciado en cero incrementado en uno y finalizando en n-1

Coreecion del codigo anterior(Indicamos el incio y alteramos el final.)

In [ ]:
```
# Ciclo for que imprime del  1 al 10
for i in range(1,11):
    print(i)
```

```
1
2
3
4
5
6
7
8
9
10
```

Ejercicio 3

Imprimir del 1 al 30 imcrementando de 5

In [ ]:
```
for i in range(5,31,5):
    print(i)
```

```
5
10
15
20
25
30
```

In [ ]:
```
#### Solictar inicio, fin del rango e incremento
inicio=int(input("Indica el valor inicial"))
fin=int(input("Indica el valor final"))
incremento=int(input("Indica inremto a aplicar"))
for rango in range(inicio,fin+1,incremento):
    print(rango)
```

```
5
10
15
20
25
30
```

Ciclos while()

Permite repetir la ejecucion de una serie de instrucciones siempre que la instruccion sea verdadera(True)

Se inicializa la variable y se debe de incrementar manuelmente

In [ ]:
```
# Mostrar por pantalla los numeros del 1 al 10
ind =1
while ind<=10:
    print(ind)
    ind+=1
```

```
1
2
3
4
5
6
7
8
9
10
```

Break

Sentencias brake, permite detener el ciclo si la condicion es True(Veradadera)

In [ ]:
```
# Leer un numero entero y determinar si es primo o no.
### Un numero es primo cuando solo es divisible entre el mismo y la unidad
### ejemplo: 5 es un numero primo, ya que solo sus divisores son 1 y 5
div = True
num= int(input("Indique un numero entero para analizar si es primo"))
for i in range(2,num):
    if num%i == 0:
        div=False
    print(i)
    break
if div:
    print(f"{num} es primo")
else:
    print(f"{num} no es primo")
```

```
2
5 es primo
```

Listas en python

La listas son usadas para almacenar multiples elementos en una sola variable

Se crean usando corchetes

In [ ]:
```
# Creamos la lista
frutas= ["manzana","bananda","fresa"]
# Imprimimos la lista
print(frutas)
```

```
['manzana', 'bananda', 'fresa']
```

-Las listas permiten elementos repetidos.

-Pueden ordenarse

-podemos saber la cantidad de elementos que contine

-se pueden accesar individualmente

In [ ]:
```
frutas= ["manzana","bananda","fresa","manzana","bananda","fresa"]
print(frutas)
```

```
['manzana', 'bananda', 'fresa', 'manzana', 'bananda', 'fresa']
```

In [ ]:
```
print(len(frutas))
```

```
6
```

In [ ]:
```
print(frutas[3])
```

```
manzana
```

In [ ]:
```
print(frutas.sort()) # Ordenamos y observamos que la listas son mutables
print(frutas)
```

```
None
['bananda', 'bananda', 'fresa', 'fresa', 'manzana', 'manzana']
```

In [ ]:
```
print(frutas[3])
```

```
fresa
```

In [ ]:
```
fruit =["pedro","pablo","paco"]
number =[1,2,3,4]
boleean =[True,False,True]
mix = ["a",34,True,40,"femenino"]
```

In [ ]:
```
print(type(fruit))
print(type(number))
print(type(boleean))
print(type(mix))
```

```
<class 'list'>
<class 'list'>
<class 'list'>
<class 'list'>
```

Acceso a item(elemenos)

In [ ]:
```
print(mix)
```

```
print(mix[3])
```

```
['a', 34, True, 40, 'femenino']
40
```

los indices negativos inician apuntando al elemento final de la lista

In [ ]:
```
print(mix[-1])
```

```
femenino
```

In [ ]:
```
#imprimir rango de elementos, considerar que llega hasta n-1
print(mix[1:4])
```

```
[34, True, 40]
```

In [ ]:
```
# muestra los primeros 4 elementos
print(mix[:4])
```

```
['a', 34, True, 40]
```

In [ ]:
```
# muestra los elementos a partir de la segunda posicion
print(mix[2:])
```

```
[True, 40, 'femenino']
```

In [ ]:
```
print(mix[-4:-1])
```

```
[34, True, 40]
```

Verficando si un elemento existe en la lista

In [ ]:
```
lista= ["apple","banana","cherry"]
if "apple" in lista:
    print("Si 'apple' esta en la lista de frutas")
```

```
Si 'apple' esta en la lista de frutas
```

agregando elementos a la lista(existente)

append()

los nuevos elementos se anexan al final

In [ ]:
```
print(lista)
lista.append("fresa")
print(lista)
```

```
['manzana', 'platano', 'kiwi', 'uva', 'mango', 'naranja', 'fresa', 'fresa']
['manzana', 'platano', 'kiwi', 'uva', 'mango', 'naranja', 'fresa', 'fresa', 'fresa']
```

podemos insertar un elemento en un a posicion dada por el indice

insert()

In [ ]:
```
print(lista)
lista.insert(2,"cereza")
print(lista)
```

```
['apple', 'banana', 'cherry', 'naranja', 'fresa', 'fresa']
['apple', 'banana', 'cereza', 'cherry', 'naranja', 'fresa', 'fresa']
```

Cambiar valores de los elementos

In [ ]:
```
print(lista)
lista[3]="cereza"
print(lista)
```

```
['manzana', 'platano', 'cereza', 'cherry', 'naranja', 'fresa', 'fresa']
['manzana', 'platano', 'cereza', 'cereza', 'naranja', 'fresa', 'fresa']
```

In [ ]:
```
# cambio de valores mediante rango
print(lista)
lista[2:5]=["kiwi","uva","mango"]
print(lista)
```

```
['manzana', 'platano', 'kiwi', 'uva', 'mango', 'naranja', 'fresa', 'fresa', 'fresa']
['manzana', 'platano', 'kiwi', 'uva', 'mango', 'naranja', 'fresa', 'fresa', 'fresa']
```

Remover elementos de la lista

remove()

Remueve la primer coicidencia que encuentre

In [ ]:
```
print(lista)
lista.remove("platano")
print(lista)
```

```
['manzana', 'platano', 'kiwi', 'uva', 'mango', 'naranja', 'fresa', 'fresa']
['manzana', 'platano', 'kiwi', 'uva', 'mango', 'naranja', 'fresa']
```

pop(), remuve el elemento indicado

In [ ]:
```
print(lista)
lista.pop(2)
print(lista)
```

```
['manzana', 'platano', 'kiwi', 'uva', 'mango', 'naranja', 'fresa']
['manzana', 'platano', 'uva', 'mango', 'naranja', 'fresa']
```

In [ ]:
```
print(lista)
lista.pop()
print(lista)
```

```
['manzana', 'platano', 'uva', 'mango', 'naranja', 'fresa']
['manzana', 'platano', 'uva', 'mango', 'naranja']
```

clear()

vacia la lista completa

In [ ]:
```
print(lista)
lista.clear()
print(lista)
```

```
['manzana', 'platano', 'uva', 'mango', 'naranja']
[]
```

Recorrer una lista

In [ ]:
```
lista = ["apple","banana","cherry"]
for fruta in lista:
    print(fruta)
```

```
apple
banana
cherry
```

In [ ]:
```
# Recorrido por indice
for i in range(len(lista)):
    print(lista[i])
```

```
apple
banana
cherry
```

Ordenamiento de lista

sort()

In [ ]:
```
lista.sort()
print(lista)
```

```
['apple', 'banana', 'cherry']
```

In [ ]:
```
numbers = [100,50,65,82,23]
numbers.sort()
print(numbers)
```

```
[23, 50, 65, 82, 100]
```

Ordenamiento descendente

In [ ]:
```
numbers.sort(reverse=True)
print(numbers)
```

```
[100, 82, 65, 50, 23]
```

Copia de lista

copy()

In [ ]:
```
personas =["pedro","Paco","Luis"]
personcopy=personas.copy()
print(personcopy)
```

```
['pedro', 'Paco', 'Luis']
```

In [ ]:
```
# metodo list()
copiapersona=list(personas)
print(copiapersona)
```

Union o concatenacion de listas

In [ ]:
```
lista1 = ["a","b","c"]
lista2 =[1,2,3]
lista3 = lista1 + lista2
print(lista3)
```

```
['a', 'b', 'c', 1, 2, 3]
```

In [ ]:
```
# Recorrido por elementos
for x in lista2:
    lista1.append(x)

print(lista1)
```

```
['a', 'b', 'c', 1, 2, 3]
```

In [ ]:
```
# extend()
lista1.extend(lista2)
print(lista1)
```

```
['a', 'b', 'c', 1, 2, 3, 1, 2, 3]
```

Tuplas

Estan ordenadas, son inmutables (no pueden cambiar) y permite elementos duplicadas

In [ ]:
```
frutas = ("manzana","platano","cereza","manzana","cereza")
print(frutas)
```

```
('manzana', 'platano', 'cereza', 'manzana', 'cereza')
```

In [ ]:
```
# Imprime numero de elementos de la lista
print(len(frutas))
```

```
5
```

In [ ]:
```
tupla = ("Paco",)
notupla = ("paco")
print(type(tupla))
```

```
print(type(notupla))
```

```
<class 'tuple'>
<class 'str'>
```

In [ ]:
```
frutasss = tuple(("manzana","platano","Cereza","Manzana","cereza",))
print(frutasss)
```

```
('manzana', 'platano', 'Cereza', 'Manzana', 'cereza')
```

Acceso a elementos de la tupla

In [ ]:
```
# Imprime el segundoi elemento de la tupla frutas
print(frutas[1])
print(frutas[-1])
print(frutas[2:5])
```

```
platano
cereza
('cereza', 'manzana', 'cereza')
```

In [ ]:
```
# verificacando si un elemeto existen
if "platano" in frutas:
    print("Si, 'platano' esta en al tupla frutas")
```

```
Si, 'platano' esta en al tupla frutas
```

Cambio de valores en la tupla

In [ ]:
```
# modificacion el segundo elemento
x = (1,2,3)
y = list(x)
y[1]=9
x=tuple(y)
print(x)
```

```
(1, 9, 3)
```

In [ ]:
```
z = list(x)
z.append(6)
x=tuple(z)
print(x)
```

```
(1, 9, 3, 6)
```

Acceder a elementos de la tupla mediante ciclos

In [ ]:
```
for f in frutas:
    print(f)
```

```
manzana
platano
cereza
manzana
cereza
```

In [ ]:
```
i = 0
while i < len(frutas):
    print(frutas[i])
    i +=1
```

```
manzana
platano
cereza
manzana
cereza
```

Diccionarios

In [ ]:
```
auto = {
    "marca" : "ford",
    "modelo": "Mustang",
    "año":1964
}
print(auto)
```

```
{'marca': 'ford', 'modelo': 'Mustang', 'año': 1964}
```

In [ ]:
```
print(auto["modelo"])
```

```
Mustang
```

Curso: Tópicos de Ciencia de Datos
Tarea: Examen Unidad 1 (Python)
Fecha de vencimiento: Saturday, 24 de February de 2024, 20:00

FRANCISCO JAVIER ATAXCA GOXCON
201U0223@alumno.itssat.edu.mx

Valor: 40

Cambiar usuario

1 de 6

Página 1 de 5

Entrega

Enviado para calificar
Calificado
La tarea fue enviada 1 día 9 horas antes
El estudiante puede editar esta entrega
ExamenU1-FranciscoAtaxca.pdf
Comentarios (0)

Calificación

Calificación:

Practicas

PDF	Otro formato 0 puntos			Con formato 1 puntos			
Reporte con capturas	No envió 0 puntos			Si envió 4 puntos			
Funcionamiento de código	No funciona 0 puntos	Soluciona 1 prueba 7 puntos	Soluciona 2 pruebas 14 puntos	Soluciona 3 pruebas 21 puntos	Soluciona 4 pruebas 28 puntos	Soluciona 5 pruebas 35 puntos	

Calificación actual en el libro de calificaciones

40,00

Comentarios de retroalimentación

Notificar a los estudiantes

Guardar cambios

Reiniciar

Instituto Tecnológico Superior de San Andrés Tuxtla

Ingeniería Informática

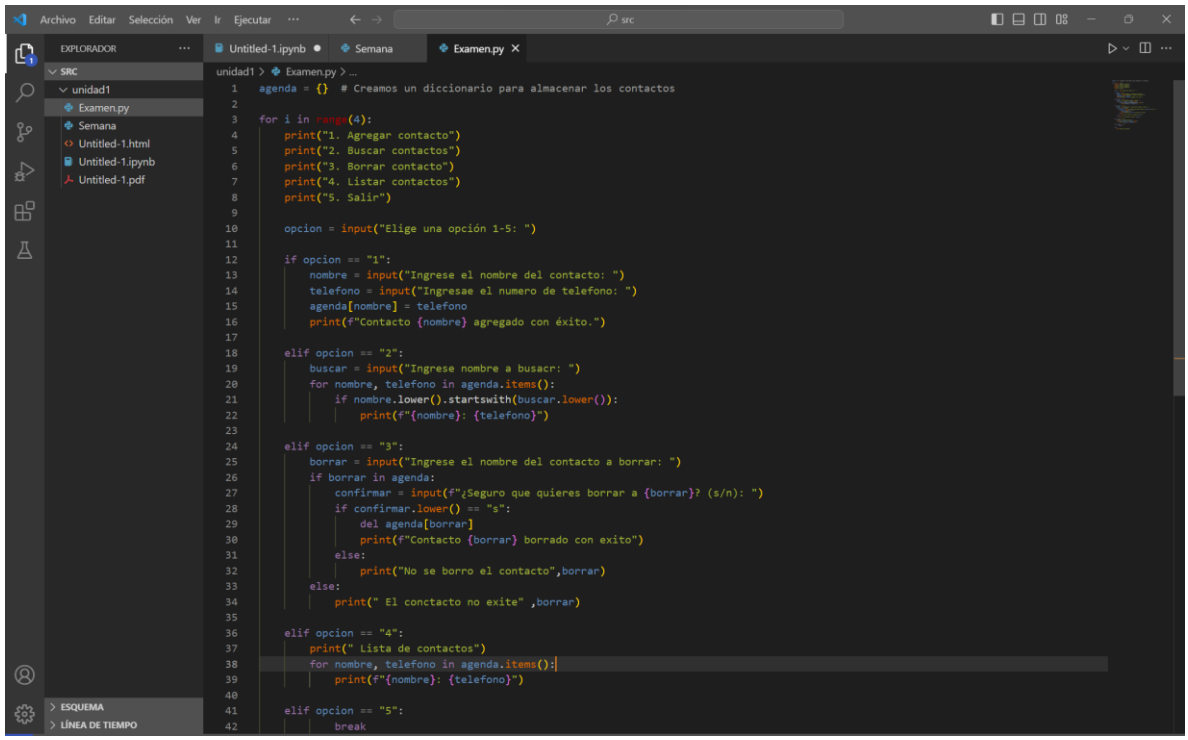
Tópicos de ciencia de datos

Octavo Semestre

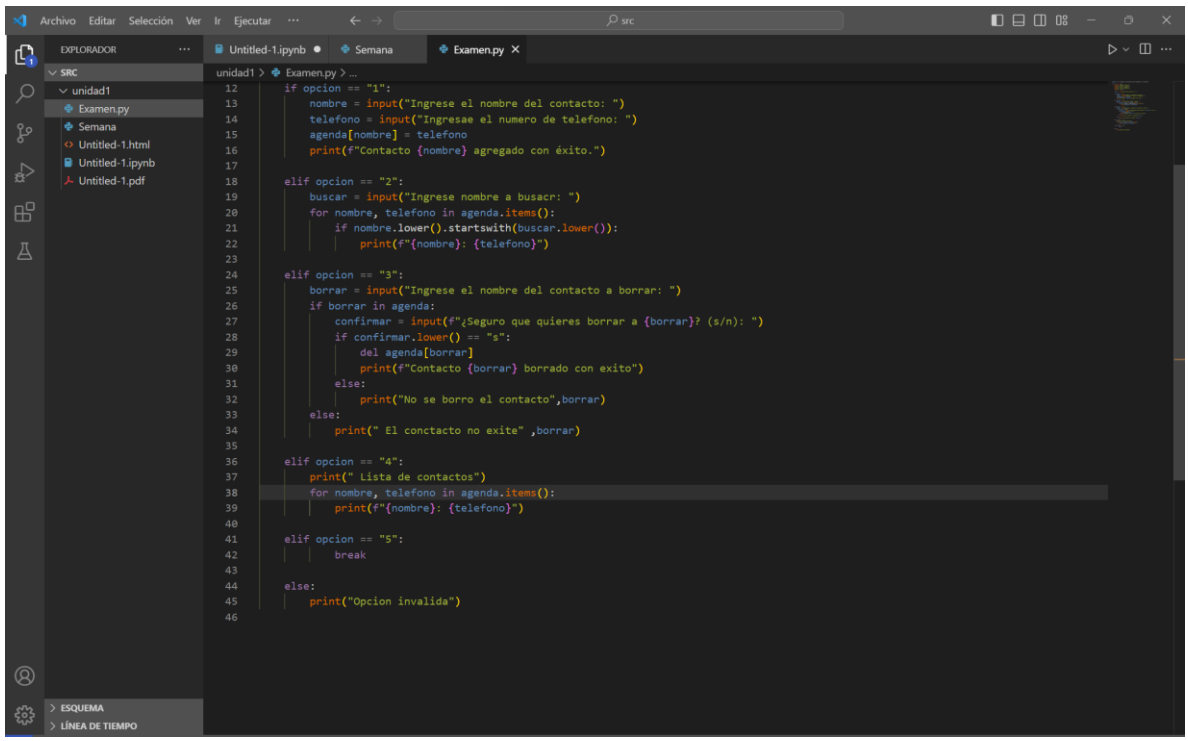
Alumno: Francisco Javier Ataxca Goxcon

Docente: M.T.I. Rogelio Enrique Telona Torres.

Código

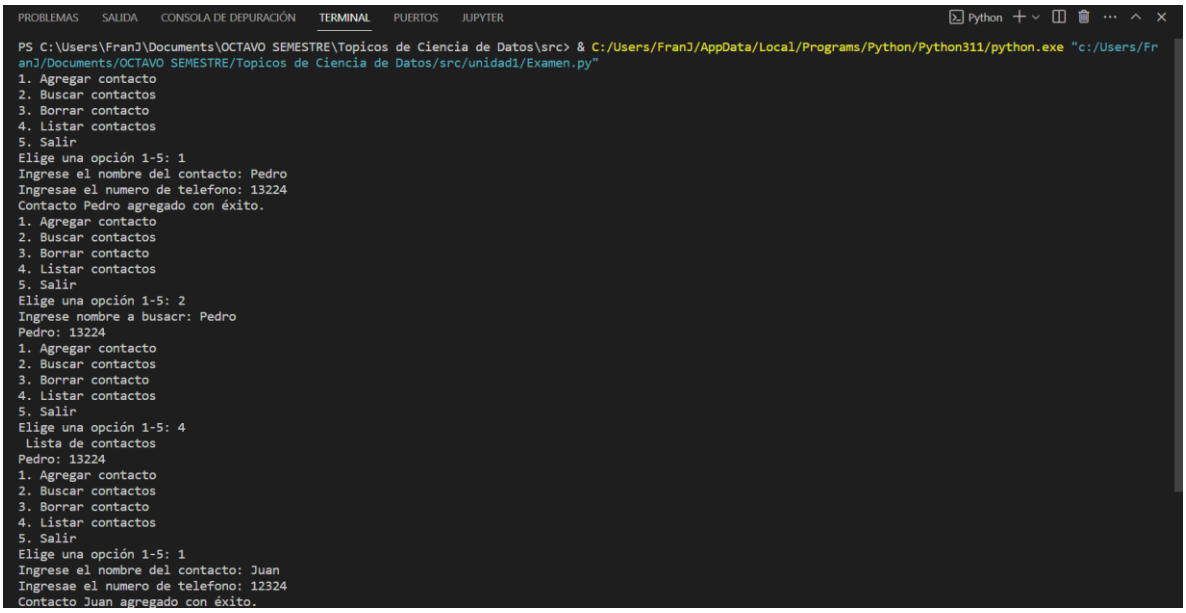


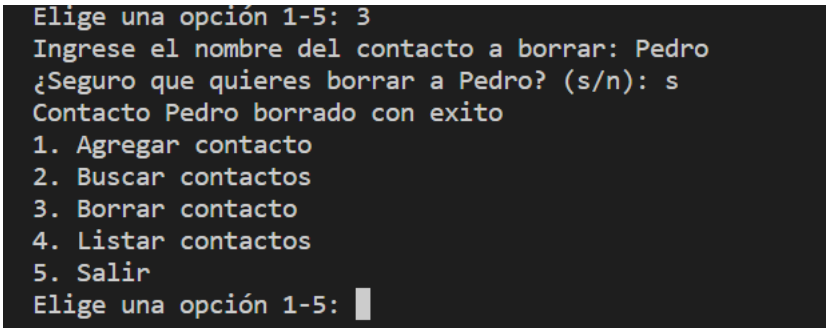
```python
agenda = {}  # Creamos un diccionario para almacenar los contactos

for i in range(4):
    print("1. Agregar contacto")
    print("2. Buscar contactos")
    print("3. Borrar contacto")
    print("4. Listar contactos")
    print("5. Salir")

    opcion = input("Elige una opción 1-5: ")

    if opcion == "1":
        nombre = input("Ingrese el nombre del contacto: ")
        telefono = input("Ingresae el numero de telefono: ")
        agenda[nombre] = telefono
        print(f"Contacto {nombre} agregado con éxito.")

    elif opcion == "2":
        buscar = input("Ingrese nombre a busacr: ")
        for nombre, telefono in agenda.items():
            if nombre.lower().startswith(buscar.lower()):
                print(f"{nombre}: {telefono}")

    elif opcion == "3":
        borrar = input("Ingrese el nombre del contacto a borrar: ")
        if borrar in agenda:
            confirmar = input(f"¿Seguro que quieres borrar a {borrar}? (s/n): ")
            if confirmar.lower() == "s":
                del agenda[borrar]
                print(f"Contacto {borrar} borrado con exito")
            else:
                print("No se borro el contacto",borrar)
        else:
            print(" El conctacto no exite" ,borrar)

    elif opcion == "4":
        print(" Lista de contactos")
        for nombre, telefono in agenda.items():
            print(f"{nombre}: {telefono}")

    elif opcion == "5":
        break
```



```python
    if opcion == "1":
        nombre = input("Ingrese el nombre del contacto: ")
        telefono = input("Ingresae el numero de telefono: ")
        agenda[nombre] = telefono
        print(f"Contacto {nombre} agregado con éxito.")

    elif opcion == "2":
        buscar = input("Ingrese nombre a busacr: ")
        for nombre, telefono in agenda.items():
            if nombre.lower().startswith(buscar.lower()):
                print(f"{nombre}: {telefono}")

    elif opcion == "3":
        borrar = input("Ingrese el nombre del contacto a borrar: ")
        if borrar in agenda:
            confirmar = input(f"¿Seguro que quieres borrar a {borrar}? (s/n): ")
            if confirmar.lower() == "s":
                del agenda[borrar]
                print(f"Contacto {borrar} borrado con exito")
            else:
                print("No se borro el contacto",borrar)
        else:
            print(" El conctacto no exite" ,borrar)

    elif opcion == "4":
        print(" Lista de contactos")
        for nombre, telefono in agenda.items():
            print(f"{nombre}: {telefono}")

    elif opcion == "5":
        break

    else:
        print("Opcion invalida")
```

Ejecución


PROBLEMAS
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
PUERTOS
JUPYTER
Python
PS C:\Users\FranJ\Documents\OCTAVO SEMESTRE\Topicos de Ciencia de Datos\src> & C:/Users/FranJ/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/FranJ/Documents/OCTAVO SEMESTRE/Topicos de Ciencia de Datos/src/unidad1/Examen.py"
1. Agregar contacto
2. Buscar contactos
3. Borrar contacto
4. Listar contactos
5. Salir
Elige una opción 1-5: 1
Ingrese el nombre del contacto: Pedro
Ingresae el numero de telefono: 13224
Contacto Pedro agregado con éxito.
1. Agregar contacto
2. Buscar contactos
3. Borrar contacto
4. Listar contactos
5. Salir
Elige una opción 1-5: 2
Ingrese nombre a busacr: Pedro
Pedro: 13224
1. Agregar contacto
2. Buscar contactos
3. Borrar contacto
4. Listar contactos
5. Salir
Elige una opción 1-5: 4
Lista de contactos
Pedro: 13224
1. Agregar contacto
2. Buscar contactos
3. Borrar contacto
4. Listar contactos
5. Salir
Elige una opción 1-5: 1
Ingrese el nombre del contacto: Juan
Ingresae el numero de telefono: 12324
Contacto Juan agregado con éxito.


Elige una opción 1-5: 3
Ingrese el nombre del contacto a borrar: Pedro
¿Seguro que quieres borrar a Pedro? (s/n): s
Contacto Pedro borrado con exito
1. Agregar contacto
2. Buscar contactos
3. Borrar contacto
4. Listar contactos
5. Salir
Elige una opción 1-5:

Código

```
agenda = {}

for i in range(4):
    print("1. Agregar contacto")
    print("2. Buscar contactos")
    print("3. Borrar contacto")
    print("4. Listar contactos")
    print("5. Salir")

    opcion = input("Elige una opción 1-5: ")

    if opcion == "1":
        nombre = input("Ingrese el nombre del contacto: ")
        telefono = input("Ingresae el numero de telefono: ")
        agenda[nombre] = telefono
        print(f"Contacto {nombre} agregado con éxito.")

    elif opcion == "2":
        buscar = input("Ingrese nombre a busacr: ")
        for nombre, telefono in agenda.items():
            if nombre.lower().startswith(buscar.lower()):
                print(f"{nombre}: {telefono}")

    elif opcion == "3":
        borrar = input("Ingrese el nombre del contacto a borrar: ")
        if borrar in agenda:
            confirmar = input(f"¿Seguro que quieres borrar a {borrar}?
(s/n): ")
            if confirmar.lower() == "s":
                del agenda[borrar]
                print(f"Contacto {borrar} borrado con exito")
            else:
                print("No se borro el contacto",borrar)
        else:
            print(" El conctacto no exite" ,borrar)

    elif opcion == "4":
        print(" Lista de contactos")
        for nombre, telefono in agenda.items():
            print(f"{nombre}: {telefono}")

    elif opcion == "5":
            break
```

```
    else:
        print("Opcion invalida")
```