



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Lenguaje de Programación Python

Actividad 10:

Diccionario

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 29 de Octubre del 2023

co

RMBA_PY_ACT10.ipynb

☆

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

ComentarioCompartirB

+ Código+ Texto

Conectado

ACTIVIDAD 10

```
[ ] from IPython.display import clear_output
import pandas as pd
import random
!pip install pyyaml
import json
import pprint
import yaml
```

Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (6.0.1)

```
[ ]
def validar_num(msge,msgeError="ERROR: Deberas ingresar solamente un numero"):
    while True:
        try:
            num = int(input(f"{msge}: "))
        except:
            print (msgeError)
            continue
        else:
            break
    return num
```

```
[ ]
def auto(dicc):
```

Conectado a del backend de Google Compute Engine que utiliza Python 3

co

RMBA_PY_ACT10.ipynb

☆

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

ComentarioCompartirB

+ Código+ Texto

RAMDisco

```
def auto(dicc):

    listadicc=[]
    clave=["ID","Nombre","Apellido","Sexo","Edad","Puesto"]
    id=[98345,63854,17287,37618,70134,28462,48267]
    nombre=["Maria","Laura","Sara","Juan","Pedro","Andres","Brayan"]
    apellido=["Garcia","Rodriguez","Vargas","Lopez","Sanchez","Perez","Rocha"]
    sexo=["M","H"]
    edad=[25,23,28,24,33,29,28]
    puesto=["Secretario","Vendedor","Gerente","Vendedor","Contador","Atencion al cliente","Recursos Humanos"]

    dicc=dict(zip(clave,[random.choice(id),random.choice(nombre),random.choice(apellido),random.choice(sexo),random.choice(edad),random.choice(puesto)]))
    listadicc.append(dicc)

    print("Se agrego la persona de manera correcta")
```

```
[ ]

def manual(dicc):
    clave=["ID","Nombre","Apellido","Sexo","Edad","Puesto"]
    id=[]
    nombre=[]
    apellido1=[]
    apellido2=[]
    sexo=[]
    edad=[]
    puesto=[]
```

Conectado a del backend de Google Compute Engine que utiliza Python 3

CO

RMBA_PY_ACT10.ipynb ☆

ComentarioCompartirB

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

+ Código+ Texto

RAMDisco

{x}

<>

```
sexo=[]
edad=[]
puesto=[]
i=0
opc=1
while opc!=2:
    clear_output()
    print("Quieres agregar un nuevo trabajador a la lista?")
    print("1.- SI")
    opc=validar_num("2.- NO ")
    if opc==2:
        break
    idag=validar_num("Ingresa el ID del trabajador a seleccionar:")
    id.append(idag)
    nomag=input("Ingresa el nombre del trabajador: ")
    nombre.append(nomag)

    ap1ag=input("Ingresa el primer apellido del trabajador: ")
    apellido1.append(ap1ag)

    ap2ag=input("Ingresa el segundo apellido del trabajador: ")
    apellido2.append(ap2ag)

    sexoag=input("Ingresa el sexo del trabajador: ")
    sexo.append(sexoag)

    edag=input("Ingresa la edad del trabajador: ")
    edad.append(edag)

    puestag=input("Ingresa el puesto del trabajador: ")
```

✓ Conectado a del backend de Google Compute Engine que utiliza Python 3

CO

RMBA_PY_ACT10.ipynb ☆

ComentarioCompartirB

ArchivoEditarVerInsertarEntorno de ejecuciónHerramientasAyudaSe han guardado todos los cambios

+ Código+ Texto

RAMDisco

{x}

<>

```
    puestag=input("Ingresa el puesto del trabajador: ")
    puesto.append(puestag)

    i+=1
    input("Presiona enter para continuar")
    global op
    op=0

    oprand=random.randint(0,i)
    dicc=dict(zip(clave,[id[oprand],nombre[oprand],apellido1[oprand],apellido2[oprand],sexo[oprand],edad[oprand],puesto[oprand]]))

    print("Trabajador fue agregado correctamente")

[ ]

def imprimir(dicc):
    op=random.randint(1,3)
    if op==1:
        print(json.dumps(dicc, sort_keys=False, indent=2))
    elif op==2:
        print(yaml.dump(dicc, sort_keys=False, default_flow_style=False))
    elif op==3:
        salida=pd.DataFrame(dicc)
```

✓ Conectado a del backend de Google Compute Engine que utiliza Python 3

co

RMBA_PY_ACT10.ipynb

☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

Comentario

Compartir

⚙️

B

+ Código + Texto

RAM Disco

```
[ ] def buscar(dicc):
    encontrado=[]
    encontrado=False
    busc=validar_num("Ingresa el ID que busca: ")
    for diccionario in dicc:
        if diccionario.get("ID")==busc:
            print(f"ID encontrado en la lista: {busc}")
            encontrado.append(dicc)
            salida=pd.DataFrame(encontrado)
            print(salida)
            encontrado = True
    if not encontrado:
        print(f"No se encontro ningun registro con el ID {busc}")

def eliminar(lista_dic):
    clear_output()
    encontrado = False
    id_eliminar = validar_num("Ingresa el ID de la persona que buscas: ")
    for i, diccionario in enumerate(diccionario):
        if diccionario.get("ID") == id_eliminar:
            print(f"{eliminar} eliminada crrectamente")
            del lista_dic[i]
            encontrado = True
    if not encontrado:
        print(f"No se encontro ningun registro con el ID {id_eliminar}")
```

Conectado a del backend de Google Compute Engine que utiliza Python 3

co

RMBA_PY_ACT10.ipynb

☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

Comentario

Compartir

⚙️

B

+ Código + Texto

RAM Disco

```
dicc=[]
while True:
    print("---MENU---")
    print("1.- Agregar (automatico)")
    print("2.- Agregar (manual)")
    print("3.- Imprimir lista")
    print("4.- Buscar {ID}")
    print("5.- Ordenar")
    print("6.- Eliminar {ID}")
    print("7.- Borrar toda la lista")
    print("0.- Salir")
    op=validar_num("Escoge una opcion: ")
    if op==1:
        auto(dicc)
        input("Presiona ENTER para borrar la pantalla")
        clear_output()
        continue
    elif op==2:
        manual(dicc)
        input("Presiona ENTER para borrar la pantalla")
        clear_output()
        continue
    elif op==3:
        imprimir(dicc)
        input("Presiona ENTER para borrar la pantalla")
        clear_output()
        continue
    elif op==4:
        buscar(dicc)
        input("Presiona ENTER para borrar la pantalla")
```

Conectado a del backend de Google Compute Engine que utiliza Python 3

CO

RMBA_PY_ACT10.ipynb ☆

Comentario

Compartir

⚙️

B

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

+ Código + Texto

RAM

Disco

↑

↓

↻

⚙️

📄

🗑️

⋮

❌

🔍

{x}

📁

<>

📄

📧

```
imprimir(dicc)
input("Presiona ENTER para borrar la pantalla")
clear_output()
continue

elif op==4:
    buscar(dicc)
    input("Presiona ENTER para borrar la pantalla")
    clear_output()
    continue

elif op==5:
    input("Presiona ENTER para borrar la pantalla")
    clear_output()
    continue

elif op==6:
    eliminar(dicc)
    input("Presiona ENTER para borrar la pantalla")
    clear_output()
    continue

elif op==7:
    dicc=[]
    input("Presiona ENTER para borrar la pantalla")
    clear_output()
    continue

elif op==0:
    break

---MENU---
1.- Agregar (automatico)
2.- Agregar (manual)
3.- Imprimir lista
4.- Buscar {ID}
```

✓ Conectado a del backend de Google Compute Engine que utiliza Python 3

CO

RMBA_PY_ACT10.ipynb ☆

Comentario

Compartir

⚙️

B

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado todos los cambios

+ Código + Texto

RAM

Disco

↑

↓

↻

⚙️

📄

🗑️

⋮

❌

🔍

{x}

📁

<>

📄

📧

```
clear_output()
continue

elif op==7:
    dicc=[]
    input("Presiona ENTER para borrar la pantalla")
    clear_output()
    continue

elif op==0:
    break

--MENU--
1.- Agregar (automatico)
2.- Agregar (manual)
3.- Imprimir lista
4.- Buscar {ID}
5.- Ordenar
6.- Eliminar {ID}
7.- Borrar toda la lista
0.- Salir
ERROR: Debes ingresar un numero
ERROR: Debes ingresar un numero
ERROR: Debes ingresar un numero
Escoge una opcion: : Escoge una opcion: : Escoge una opcion: :
Escoge una opcion: :
```

✓ Conectado a del backend de Google Compute Engine que utiliza Python 3

