Manual Técnico

Correo creador: Edwinhndz98@gmail.com

A continuación, se hará una breve introducción al código cuente de Simple SQL, un lenguaje para cargar registros y hacer consultas en consola.

Método Cargar:

```
1.  def Cargar(rutaa):
2.      archivo = open(rutaa)
3.      info = json.load(archivo)
4.      archivo.close()
5.
6.      for element in info:
7.          aux = str(element)
8.          aux = aux.replace("'", "")
9.          aux = aux.replace("{", "")
10.         aux = aux.replace("}", "")
11.         aux = aux.replace(":", "")
12.         aux = aux.replace(" ", "")
13.         arreglo = aux.split(",")
14.         arreglo[0]=arreglo[0].replace("nombre","")
15.         arreglo[1]=arreglo[1].replace("edad","")
16.         arreglo[2]=arreglo[2].replace("activo","")
17.         arreglo[3]=arreglo[3].replace("promedio","")
18.         lista_nombres.append(arreglo[0])
19.         lista_edad.append(arreglo[1])
20.         lista_activo.append(arreglo[2])
21.         lista_promedio.append(arreglo[3])
```

En el método tiene parámetro “ruta” que bien podría ser la ruta directa del archivo que se desea, o bien el nombre teniendo en cuenta que debe guardarse en la misma carpeta que el main.py, el método reemplaza varias cadenas en las cuales están encerradas los datos una vez hecho, se le añaden a una lista predeterminada.

Suma Lista:

```
1.  def sumalista(listaNumeros):
2.      laSuma = 0
3.      for i in listaNumeros:
4.          laSuma = laSuma + float(i)
5.      return laSuma
```

En el método , un bucle for va recorriendo todas las posiciones de una lista dada, esta lista puede ser ya sea la de Edades, o bien la de Promedios, y se suman cada posición. Devolviendo el numero total.

Método Pruebas:

```
def pruebas(valor):
    txt = valor
    txt2=txt
    x = re.search("\A" + " ", txt)
    contador = 0
    while (x):

        x = re.search("\A"+" ", txt)
        txt = txt.replace(" ", "", 1)
        contador = contador + 1
    txt2=txt2.replace(" ","",(contador-1))
    return txt2
```

El método recibe una cadena como parámetro, y mediante la librería regex, elimina solamente los espacios iniciales, sin alterar ni cambiar la cadena mas allá de los espacios al principio de la cadena.

Método Validar:

```
def Validar(lista_atributos):
    bandera = False
    for element in lista_atributos:
        if(element in campos):
            bandera=True
        else:
            bandera=False
            return bandera
    return bandera
```

Este método funciona mediante una lista de atributos obtenidos mediante el input de seleccionar, este compara los atributos dados con los que son validos (nombre, edad, promedio, activo) si alguno de los atributos datos no coincide con los predeterminados devolverá un valor booleano, ya sea True o False.

Método Validar2:

```
def Validar2(atributo, campos):
    if(atributo in campos):
        bandera=True
        return bandera
    else:

        bandera=False
        return bandera
```

Este método al igual que Validar, compara el atributo de la condición dada, es decir, el atributo después de la palabra DONDE, el cual también busca entre los predeterminados si existe el dicho atributo, y regresa un valor booleano, True o False.

Método Info:

```
def info(condicion, atributos):
    data=""
    for i in range(len(lista_nombres)):
        if (condicion == lista_nombres[i]):
            index = i
            for element in atributos:
                if (element == "nombre"):
                    data = data + "Nombre: " + lista_nombres[index] + "\n"
                elif (element == "edad"):
                    data = data + "Edad: " + lista_edad[index] + "\n"
                elif (element == "activo"):
                    data = data + "Activo: " + lista_activo[index] + "\n"
                elif (element == "promedio"):
                    data = data + "Promedio: " + lista_promedio[index] + "\n"
            return data

        elif (condicion == lista_edad[i]):
            index = i
            for element in atributos:
                if (element == "nombre"):
                    data = data + "Nombre: " + lista_nombres[index] + "\n"
                elif (element == "edad"):
                    data = data + "Edad: " + lista_edad[index] + "\n"
                elif (element == "activo"):
                    data = data + "activo: " + lista_activo[index] + "\n"
                elif (element == "promedio"):
                    data = data + "Promedio: " + lista_promedio[index] + "\n"
            return data
        elif (condicion == lista_promedio[i]):
            index = i
            for element in atributos:
                if (element == "nombre"):
                    data = data + "Nombre: " + lista_nombres[index] + "\n"
                elif (element == "edad"):
                    data = data + "Edad: " + lista_edad[index] + "\n"
                elif (element == "activo"):
                    data = data + "Activo: " + lista_activo[index] + "\n"
                elif (element == "promedio"):
                    data = data + "Promedio: " + lista_promedio[index] + "\n"
            return data
        elif (condicion == lista_activo[i]):
            index = i
            for element in atributos:
                if (element == "nombre "):
                    data = data + "Nombre: " + lista_nombres[index] + "\n"
                elif (element == "edad"):
                    data = data + "Edad: " + lista_edad[index] + "\n"
                elif (element == "activo"):
                    data = data + "Activo: " + lista_activo[index] + "\n"
                elif (element == "promedio"):
                    data = data + "Promedio: " + lista_promedio[index] + "\n"
            return data
```

Este método es el principal a la hora de hacer consultas, se le debe dar un arreglo con los atributos que se desea considerar, junto con la condición, recorrerá cada posición para obtener la posición en los arreglos de información. Una vez obtenido,

consulta todos los arreglos donde están los registros y con la posición de la condición, obtiene la información y se va concatenando en una variable. Al final la función regresa la variable con la información.

Método Asterisco:

```
1.  def asterisco():
2.  
3.      for i in range(len(lista_nombres)):
4.          index=i+1
5.          print("")
6.          print(index,".--------------------")
7.          print("nombre: ",lista_nombres[i])
8.          print("edad: ",lista_edad[i])
9.          print("activo: ",lista_activo[i])
10.         print("promedio: ",lista_promedio[i])
11.         print("----------------------")
```

Este método imprime toda la información registrada, obteniendo la información de cada arreglo y concatenando a los print.