

Uso de Archivos de texto en Python dar Formato a Listas

Objetivo: leer archivos de texto en Python para el manejo de listas así como dar formato de salida a estas

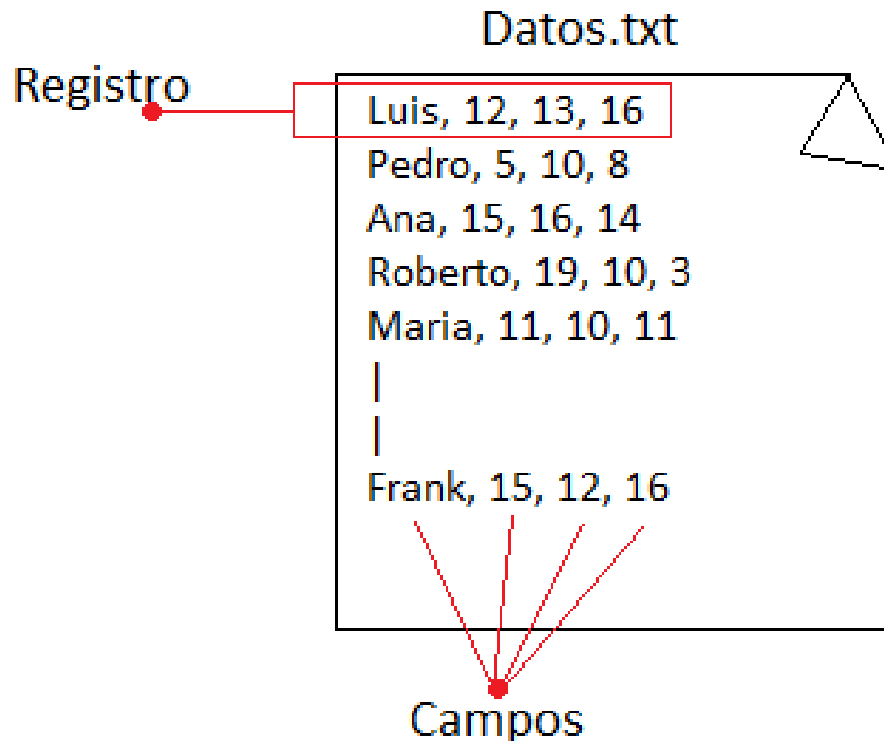
Archivos de texto

Definiciones básicas:

Archivo de texto: archivo que guarda la información en forma de caracteres.

Registro: estructura de datos que almacena datos relacionados o simplemente podríamos decir que un registro es un conjunto de datos relacionados entre sí.

Campo: dato que forma parte de un registro.



Como crear un archivo de texto con el Block de Notas

1. Abrir el Bloc de Notas.
2. Ir colocando los campos del registro separando cada campo por una coma, al final presionar **ENTER** (solo en el caso de que se quieran agregar más registros).
3. En el último registro dejar el cursor al lado del último campo del mismo, **NO PRESIONAR ENTER**.
4. Proceder a grabar el archivo, solo colocar el nombre que tendrá este documento, **NO COLOCAR .TXT**, esto lo coloca la aplicación.

Abrir el archivo de texto en modo de lectura

En Python, para abrir un archivo usaremos la función **OPEN**, que recibe el nombre del archivo que queremos abrir.

```
archivo = open("Nombre_archivo.txt")
```

Esta función intentará abrir el archivo con el nombre indicado en modo de lectura. Si tiene éxito, devolverá una referencia que nos permitirá manipular el archivo de diversas maneras, dicha referencia será almacenada en la variable correspondiente, de tal manera que cada vez que tengamos que usar el archivo simplemente usamos la variable que lo referencia.

Leer los registros de un archivo de texto:

Para leer los registros de un archivo de texto procederemos a leer cada registro del archivo línea por línea, es posible hacerlo de la siguiente forma:

```
for registro in archivo:  
    # procesar registro
```

De esta manera, la variable **registro** irá almacenando una cadena texto correspondientes a cada uno de los registros del archivo hasta leer todo el contenido del archivo.

Como separa los campos de un registro leído de un archivo de texto:

Para dividir los registros leídos de un archivo de texto en campos, debemos separar cada una de las cadenas que forman parte del registro en datos usando la función **Split**, cada vez que el intérprete encuentre el carácter coma (,) en el registro leído. Esto es posible hacerlo de la siguiente manera:

```
campos = registro.split(",")
```

Por ejemplo si el primer registro del archivo es:

Luis, 12, 13, 16

Al ejecutar la primera iteración de **for registro in archivo:**, en la variable registro se almacena la siguiente cadena:

Registro = "Luis, 12, 13, 16"

Al ejecutar sobre esta cadena la función `split()` en la variable campos se crea una lista de la siguiente manera, **campos = registro.split(",")**, se obtiene lo siguiente:

Campos = ["Luis", "12", "13", "16"]

Una vez generada la lista procedemos a obtener la información de cada campo, para ello usamos el identificador de la lista y le asociamos un entero comenzando con cero, de esta manera recuperamos cada una de las cadenas de la lista, así:

Nomb = campos[0] → se le asigna a la variable nombre "Luis"

Nota1 = int(campos[1]) → se le asigna a la variable nota1 el entero 12

Nota2 = int(campos[2]) → se le asigna el entero 13 a nota2

Nota3 = int(campos[3]) → se le asigna el entero 16 a nota3

Una vez terminadas las operaciones con el archivo en uso debemos cerrarlo usando:

archivo.close()

Formato de salida para Lista

Usando la función **cadena.format(argumentos)** podemos dar formato a datos que queremos mostrar por la consola usando el comando print.

Sintaxis

print "{pos:formato_salida}".format(valor)

En donde las { } indican que su contenido es una cadena de formato en la cual pos determina la posición del valor en la lista de argumentos al cual se le aplicara el formato y separado por los dos puntos (:) el formato que se le aplicara al valor, los diferentes tipos de formato son;

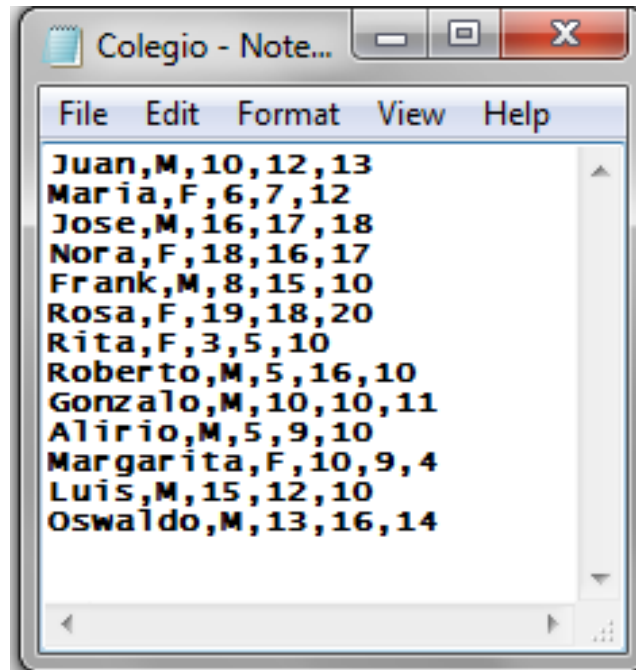
Print "{0:10}".format(nombre), imprimirá un nombre reservando para el mismo un campo de impresión de diez caracteres.

Print "{0:3d}".format(x), imprimirá un numero entero indicado en la cadena de formato con la letra **d** reservando para el mismo un campo de impresión de 3 caracteres.

Print "{0:3.2f}".format(y), imprimirá un número real indicado en la cadena de formato con la letra **f**, reservando para el mismo un campo de impresión de 3 enteros y dos decimales.

Proyecto:

Crear una aplicación en Python que lea el archivo de Colegio.txt cuyo contenido es el siguiente:



El registro del archivo muestra la siguiente información:

Nombre del estudiante, Genero, y la nota obtenida en tres parciales

La aplicación deberá:

- ❖ Leer el contenido del archivo

- ❖ Para cada estudiante:

1. La nota promedio e indicar con un mensaje si “Aprobó” o “Reprobó” la materia

- ❖ Para todos los estudiantes:

2. Porcentaje de alumnas aprobadas con respecto al total de alumnas
3. Nota promedio de los Alumnos que reprobaron la materia.
4. Nombre de alumno que Aprobó con la mayor calificación, en caso de haber mas de un alumno con la misma nota reportar el primero.

Código

```
# Inicializar variables
nom = genero = estatus = ""
n1 = n2 = n3 = 0
prom = promMR = porcFA = 0.0
contFA = contF = 0
acumMR = 0.0
contM = 0
band = 0
mnom = ""
mprom = 0.0
registro = ""
```

```
# Abrimos el archivo
archivo = open("Colegio.txt")
# Ciclo para leer los registros de un archivo
print("Nombre          Nota Final    Estatus")
for registro in archivo:
    campos = registro.split(",")
    nom = campos[0]
    genero = campos[1]
    n1 = int(campos[2])
    n2 = int (campos[3])
    n3 = int (campos[4])
    # 1._Calculo promedio del estudiante
    prom = (n1 + n2 + n3)/3
    if prom >= 9.5:
        estatus = "APROBO"
    else:
        estatus = "REPROBO"
    print("{0:12} {1:7.2f}          |{2:12}".format(nom, prom, estatus))
```

```
#Calculos para todos los alumnos
# 2._ Determinar valores calculo porcentaje
if genero == "F" or genero == "f":
    contF += 1 # Cuantas las alumnas
    if prom >= 9.5:
        contFA += 1 # Cuenta alumnas aprobadas
# 3._ Determinar valores calculo promedio
if (genero == "M" or genero == "m") and (prom < 9.5):
    acumMR += prom
    contM += 1
# 4._ Determinamos alumno con el mayor promedio
if band == 0:
    mnom = nom
    mprom = prom
    band = 1
elif mprom < prom:
    mnom = nom
    mprom = prom
```

```
print()
print ("Calculos finales y salida resultados")
# Calculamos el porcentaje
if contF > 0:
    porc = contFA/contF*100
    print ("Porcentaje de alumnas que aprobaron: ", porc, "%")
else:
    print ("No hay alumnas en el cuerso")
if contM > 0:
    promMR = acumMR/contM
    print ("Nota promedio alumnos reprobados: ", promMR)
else:
    print ("No hay alumnos en el curso")
print ("Estudiante con la mayor calificacion: ", mnom)
print ("Fin programa")
```

Salida por consola

Nombre	Nota Final	Estatus
Juan	11.67	APROBO
Maria	8.33	REPROBO
Jose	17.00	APROBO
Nora	17.00	APROBO
Frank	11.00	APROBO
Rosa	19.00	APROBO
Rita	6.00	REPROBO
Roberto	10.33	APROBO
Gonzalo	10.33	APROBO
Alirio	8.00	REPROBO
Margarita	7.67	REPROBO
Luis	12.33	APROBO
Oswaldo	14.33	APROBO

Calculos finales y salida resultados

Porcentaje de alumnas que aprobaron: 40.0 %

Nota promedio alumnos reprobados: 8.0

Estudiante con la mayor calificacion: Rosa

Fin programa