

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

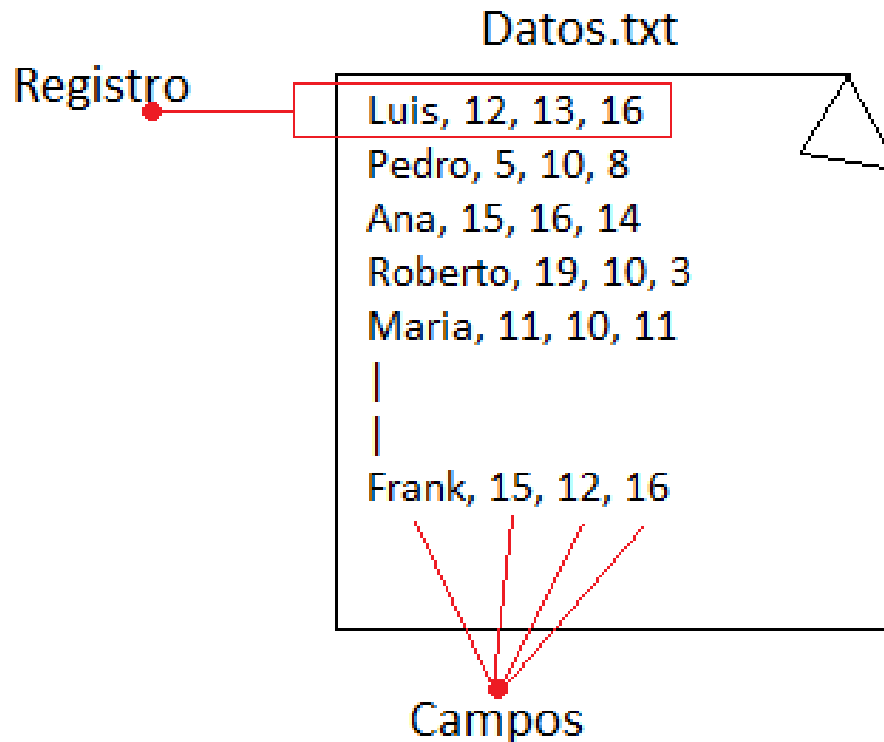
Archivos de texto

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 la 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 terminada 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;

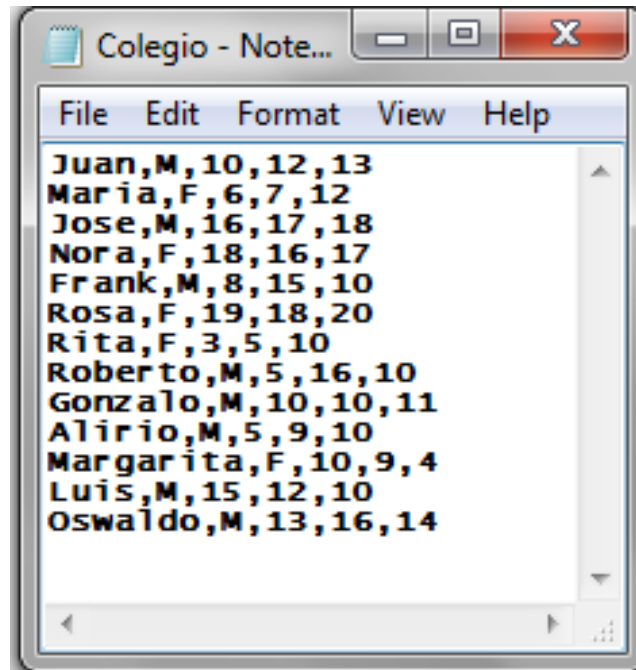
Print "{0:10}".format(nombre), imprimirá un nombre reservando para el mismo un campo de impresión de diez caracteres.

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:



```
Colegio - Note...
File Edit Format View Help
Juan,M,10,12,13
María,F,6,7,12
Jose,M,16,17,18
Nora,F,18,16,17
Frank,M,8,15,10
Rosa,F,19,18,20
Rita,F,3,5,10
Roberto,M,5,16,10
Gonzalo,M,10,10,11
Alirio,M,5,9,10
Margarita,F,10,9,4
Luis,M,15,12,10
Oswaldo,M,13,16,14
```

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.

Que variables requerimos:

Variables almacenan valores de los campos:

Guarda nombre alumno: nom → String

Guarda las notas de los parciales: n1, n2, n3 → Enteras

Variables para los cálculos:

Guarda el promedio del alumno: prom → Real

Guarda nota promedio alumnos masculinos: promMR → Real

Guarda porcentaje alumnas aprobadas: promFA → Real

Guarda cantidad alumnas aprobadas : contFA → Real

Guarda la cantidad de alumnas: contF → Entera

Guarda acumulado de notas alumnos reprobados: acumMR → Entera

Guarda la cantidad de alumnos: contM → Entera

Bandera para controlar el primer elemento de la lista: band → Entera

Guarda el nombre del alumno con la mayor nota: mnom → String

Guarda la mayor nota promedio de la lista: mprom → Real

Guarda el registro leído del archivo: registro → String

Guarda el estatus del alumno(Aprobado/Reprobado): estatus → String

Código

```
1  # Inicializar variables
2  nom = genero = estatus = ""
3  n1 = n2 = n3 = 0
4  prom = promMR = porcFA = 0.0
5  contFA = contF = 0
6  acumMR = 0.0
7  contM = 0
8  band = 0
9  mnom = ""
10 mprom = 0.0
11 registro = ""
```

```
12 # Abrimos el archivo
13 archivo = open("Colegio.txt")
14 # Titulos cabeceras de la tabla de salida
15 print("Nombre      Nota Final  Estatus")
16 # Ciclo para leer los registros de un archivo
17 for registro in archivo:
18     campos = registro.split(",") # Fragmentacion del registro en campos
19     nom = campos[0] # Asignacion de los campos a las variables
20     genero = campos[1]
21     n1 = int(campos[2])
22     n2 = int(campos[3])
23     n3 = int(campos[4])
24     # 1. Calculamos promedio del estudiante
25     prom = (n1 + n2 + n3) / 3
26     if prom >= 9.5: # Validamos si el alumno aprobo
27         estatus = "APROBO"
28     else:
29         estatus = "REPROBO"
30     print("{0:12} {1:7.2f} {2:12}".format(nom, prom, estatus))
```

Comentarios:

Línea 13 : procedemos a abrir el archivo “Colegio.txt”

Línea 15: mostramos los títulos de las cabeceras de las columnas de la tabla.

Línea 17: ciclo for que nos permitirá recorrer cada registro del archivo “Colegio.txt”

Línea 18: Fragmentamos el registro en los campos correspondientes

Líneas 20 a la 23 : asignamos los valores de los campos a las variables correspondientes

Línea 25: calculamos la nota promedio del estudiante.

Línea 26 a la 29: Condicional que nos permite determinar el estatus del alumno en función de la nota promedio

Línea 30: Mostramos por consola los datos del alumno procesado , con salida formateada

```
31 # Calculos para todos los alumnos
32 # 2._ Determinar valores calculo porcentaje
33 if genero == "F" or genero == "f":
34     contF += 1 # Cuantas las alumnas
35     if prom >= 9.5:
36         contFA += 1 # Cuenta alumnas aprobadas
37 # 3._ Determinar valores calculo promedio
38 if (genero == "M" or genero == "m") and (prom < 9.5):
39     acumMR += prom
40     contM += 1
41 # 4._ Determinamos alumno con el mayor promedio
42 if band == 0:
43     mnom = nom
44     mprom = prom
45     band = 1
46 elif mprom < prom:
47     mnom = nom
48     mprom = prom
```

Comentarios:

Líneas 33 a la 36: verificamos si el estudiante procesado es una alumna si se cumple contamos a la alumna y dentro de bloque verificamos si esta alumna aprobó, de ser así contamos la alumna aprobada. Estos datos servirán para calcular el porcentaje de alumnas que aprobaron con respecto al total de alumnas.

Líneas 38 a la 40: verificamos si el estudiante es un alumno **y** si reprobó la materia, de ser así se acumula su nota promedio y se cuenta el alumno. Estos datos servirán para calcular la nota promedio de los alumnos reprobados.

Líneas 42 a la 48: esta estructura condicional if/elif tiene como finalidad determinar cual fue el estudiante que obtuvo la mayor calificación, entre las líneas 42 y la 45 verificamos si el valor de la bandera (band) es igual a cero de ser así es porque estamos procesando el primer elemento de la lista, por lo tanto asumimos que este es el alumno con la mayor nota guardando sus datos en la variables correspondientes y cambiamos el valor de la bandera a 1, de no ser así es porque el valor de la bandera ha cambiado y el primer alumno ya fue procesado, por lo tanto procesamos las líneas entre la 46 y 48, en donde en la 46 verificamos si el valor de **mprom**, que guarda la mayor nota promedio encontrada hasta ahora, es menor que la nota promedio procesada en ese momento, de ser así, hemos encontrado un estudiante cuyo promedio es mayor a la mayor nota promedio considerada hasta ahora, por lo tanto efectuamos los cambios en la variables correspondientes.

```
49 print()
50 print("Calculos finales y salida resultados")
51 # Calculamos el porcentaje
52 if contF > 0:
53     porc = contFA / contF * 100
54     print("Porcentaje de alumnas que aprobaron: ", porc, "%")
55 else:
56     print("No hay alumnas en el cuerso")
57 if contM > 0:
58     promMR = acumMR / contM
59     print("Nota promedio alumnos reprobados: ", promMR)
60 else:
61     print("No hay alumnos en el curso")
62 print("Estudiante con la mayor calificacion: ", mnom)
63 print("Fin programa")
```

Comentarios:

Línea 49: cuando alcanzamos esta línea el ciclo **for** a terminado, indicando esto que todos los registro del archivo “Colegio.txt” han sido procesados, la instrucción `print()` provoca un salto de línea en la pantalla para separar la lista mostrada hasta ahora (Tabla) de los datos finales.

Líneas 52 a la 56: verificamos si **contF** es mayor que cero, de ser así, significa que hay alumnas en la lista por lo tanto podemos calcular el porcentaje y mostrar el resultado, evitando la división por cero, de no cumplirse la condición simplemente emitimos un mensaje de que no hay alumnas en el curso.

Líneas 57 a la 61: verificamos si **contM** es mayor que cero, de ser así, significa que hay alumnos en el curso (genero masculino) y por lo tanto podemos proceder al calculo del promedio, evitando la división por cero, de no cumplirse la condición simplemente emitimos un mensaje de que no hay alumnos en el curso.

Línea 62: mostramos el nombre del estudiante que obtuvo la mayor nota promedio del curso.

Línea 63: informamos que le programa se ejecuto satisfactoriamente.

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