Manejo de archivos

March 30, 2022

Ejecuta esta línea para cargar los archivos necesarios para el notebook

```
[ ]: !wget https://raw.githubusercontent.com/cosmolejo/dataRepo/master/Frankenstein.
     ↪txt
```

```
--2022-03-15 22:26:37--
https://raw.githubusercontent.com/cosmolejo/dataRepo/master/Frankenstein.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.108.133, 185.199.109.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.108.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 2941 (2.9K) [text/plain]
Saving to: 'Frankenstein.txt'

Frankenstein.txt    100%[===================>]   2.87K  --.-KB/s    in 0s

2022-03-15 22:26:37 (48.3 MB/s) - 'Frankenstein.txt' saved [2941/2941]
```

1 Manejo de archivos en Python

En este punto del curso, hemos aprendido todo lo esencial para poder programar en Python, variables, condiciones, ciclos, funciones y librerías. Con todas estas herramientas podemos crear programas que ejecuten cualquier tipo de instrucciones para resolver algún problema. Sin embargo, hasta ahora nuestros programas no tienen "memoria" no tenemos forma de recuperar la información procesada al volver a abrir nuestro programa. Aquí entra en juego el concepto de **persistencia**.

La **persistencia** es la acción de preservar la información de un programa de forma permanente (guardado), pero a su vez también se refiere a poder recuperar la información del mismo (lectura) para que pueda ser nuevamente utilizado. En las aplicaciones profesionales y modernas es muy común que este proceso de persistencia se maneje a partir de bases de datos, estas herramientas se estudiarán en los próximos ciclos, por lo pronto (y para conocer los conceptos de los cuales parten dichas bases) estudiaremos una forma más sencilla de persistencia, el manejo de archivos.

1.1 Control genérico de archivos

Iniciemos estudiando cómo Python nativamente puede operar con archivos. Python trata los archivos de forma diferente, como texto o binario y esto es importante. Cada línea de código

incluye una secuencia de caracteres y forman un archivo de texto. Cada línea de un archivo se termina con un carácter especial, llamado `EOL` o caracteres de fin de línea como la coma `,` o el carácter de nueva línea `\n`. Este termina la línea actual y le indica al intérprete que ha comenzado una nueva. Comencemos con la lectura y escritura de archivos.

1.1.1 Apertura de archivos con `open`

Antes de realizar cualquier operación en el archivo como leer o escribir, primero tenemos que abrir ese archivo. Para ello, debemos utilizar la función incorporada de Python `open()`

Pero en el momento de la apertura, tenemos que especificar el modo, que representa el propósito de la apertura del archivo.

```
f = open("nombre_archivo", "modo")
```

Donde tenemos los siguientes modos de apertura de archivo, si no especificamos un modo, Python usará `r` por defecto.

Modo	Comportamiento
`r`	abrir un archivo existente para una operación de lectura.
`w`	abrir un archivo existente para una operación de escritura. Si el archivo ya contiene algunos datos, se sobre escribirá.
`a`	abrir un archivo existente para una operación de adición. No eliminará los datos existentes.
`r+`	Para leer y escribir datos en el archivo. Los datos anteriores en el archivo no serán borrados.
`w+`	Para escribir y leer datos. Anulará los datos existentes.
`a+`	Para añadir y leer datos del archivo. No anulará los datos existentes.

Abramos el archivo que cargamos en la primera celda.

[ ]:
```
file = open('Frankenstein.txt','r+')
print(file, type(file))
```

```
<_io.TextIOWrapper name='Frankenstein.txt' mode='r+' encoding='UTF-8'> <class
'_io.TextIOWrapper'>
```

Como vemos en el `print`, el comando `open` crea un objeto donde almacena toda la información existente en el archivo. ¿Cómo podemos leer los datos?

1.1.2 Lectura de archivos

Hay más de una forma de leer un archivo en Python. Si necesitamos extraer una cadena que contenga todos los caracteres del archivo entonces podemos usar la función interna `read`. El código completo funcionaría así:

[ ]:
```
print(file.read())
```

```
Frankenstein;
```

or, the Modern Prometheus

by Mary Wollstonecraft (Godwin) Shelley

Chapter 1

I am by birth a Genevese, and my family is one of the most distinguished of that republic. My ancestors had been for many years counsellors and syndics, and my father had filled several public situations with honour and reputation. He was respected by all who knew him for his integrity and indefatigable attention to public business. He passed his younger days perpetually occupied by the affairs of his country; a variety of circumstances had prevented his marrying early, nor was it until the decline of life that he became a husband and the father of a family.

As the circumstances of his marriage illustrate his character, I cannot refrain from relating them. One of his most intimate friends was a merchant who, from a flourishing state, fell, through numerous mischances, into poverty. This man, whose name was Beaufort, was of a proud and unbending disposition and could not bear to live in poverty and oblivion in the same country where he had formerly been distinguished for his rank and magnificence. Having paid his debts, therefore, in the most honourable manner, he retreated with his daughter to the town of Lucerne, where he lived unknown and in wretchedness. My father loved Beaufort with the truest friendship and was deeply grieved by his retreat in these unfortunate circumstances. He bitterly deplored the false pride which led his friend to a conduct so little worthy of the affection that united them. He lost no time in endeavouring to seek him out, with the hope of persuading him to begin the world again through his credit and assistance.

Beaufort had taken effectual measures to conceal himself, and it was ten months before my father discovered his abode. Overjoyed at this discovery, he hastened to the house, which was situated in a mean street near the Reuss. But when he entered, misery and despair alone welcomed him. Beaufort had saved but a very small sum of money from the wreck of his fortunes, but it was sufficient to provide him with sustenance for some months, and in the meantime he hoped to procure some respectable employment in a merchant's house. The interval was, consequently, spent in inaction; his grief only became more deep and rankling when he had leisure for reflection, and at length it took so fast hold of his mind that at the end of three months he lay on a bed of sickness, incapable of any exertion.

His daughter attended him with the greatest tenderness, but she saw with despair that their little fund was rapidly decreasing and that

```
there was no other prospect of support. But Caroline Beaufort
possessed a mind of an uncommon mould, and her courage rose to support
her in her adversity. She procured plain work; she plaited straw and
by various means contrived to earn a pittance scarcely sufficient to
support life.
```

Este método no es muy eficiente si queremos extraer línea por línea la información, en este caso, podemos separar el archivo por líneas usando la función `readlines`, esta función separa las líneas dentro de una lista:

[ ]:
```
#la función read extrae toda la información del archivo
#por este motivo debemos volverlo a abrir, de lo contrario
#file.readlines() nos entregará una lista vacía
file = open('Frankenstein.txt','r+')
lineas = file.readlines()
print(type(lineas),lineas)
```

```
<class 'list'> ['\ufeffFrankenstein;\n', '\n', 'or, the Modern Prometheus\n',
'\n', 'by Mary Wollstonecraft (Godwin) Shelley\n', '\n', '\n', 'Chapter 1\n',
'\n', '\n', 'I am by birth a Genevese, and my family is one of the most\n',
'distinguished of that republic. My ancestors had been for many years\n',
'counsellors and syndics, and my father had filled several public\n',
'situations with honour and reputation. He was respected by all who\n', 'knew
him for his integrity and indefatigable attention to public\n', 'business. He
passed his younger days perpetually occupied by the\n', 'affairs of his country;
a variety of circumstances had prevented his\n', 'marrying early, nor was it
until the decline of life that he became a\n', 'husband and the father of a
family.\n', '\n', 'As the circumstances of his marriage illustrate his
character, I cannot\n', 'refrain from relating them. One of his most intimate
friends was a\n', 'merchant who, from a flourishing state, fell, through
numerous\n', 'mischances, into poverty. This man, whose name was Beaufort, was
of a\n', 'proud and unbending disposition and could not bear to live in
poverty\n', 'and oblivion in the same country where he had formerly been\n',
'distinguished for his rank and magnificence. Having paid his debts,\n',
'therefore, in the most honourable manner, he retreated with his\n', 'daughter
to the town of Lucerne, where he lived unknown and in\n', 'wretchedness. My
father loved Beaufort with the truest friendship and\n', 'was deeply grieved by
his retreat in these unfortunate circumstances.\n', 'He bitterly deplored the
false pride which led his friend to a conduct\n', 'so little worthy of the
affection that united them. He lost no time in\n', 'endeavouring to seek him
out, with the hope of persuading him to begin\n', 'the world again through his
credit and assistance.\n', '\n', 'Beaufort had taken effectual measures to
conceal himself, and it was ten\n', 'months before my father discovered his
abode. Overjoyed at this discovery,\n', 'he hastened to the house, which was
situated in a mean street near the\n', 'Reuss. But when he entered, misery and
despair alone welcomed him. Beaufort\n', 'had saved but a very small sum of
money from the wreck of his fortunes, but\n', 'it was sufficient to provide him
```

```
with sustenance for some months, and in\n', 'the meantime he hoped to procure
some respectable employment in a\n', 'merchant's house. The interval was,
consequently, spent in inaction;\n', 'his grief only became more deep and
rankling when he had leisure for\n', 'reflection, and at length it took so fast
hold of his mind that at the end\n', 'of three months he lay on a bed of
sickness, incapable of any exertion.\n', '\n', 'His daughter attended him with
the greatest tenderness, but she saw\n', 'with despair that their little fund
was rapidly decreasing and that\n', 'there was no other prospect of support. But
Caroline Beaufort\n', 'possessed a mind of an uncommon mould, and her courage
rose to support\n', 'her in her adversity. She procured plain work; she plaited
straw and\n', 'by various means contrived to earn a pittance scarcely sufficient
to\n', 'support life.\n']
```

Si quisiéramos separar cada línea en sus palabras, podemos usar el comando `split`que vimos en un ejemplo anteriormente, su resultado lo podríamos almacenar en una lista de listas para futuras operaciones.

[ ]:
```
stop = 0 #usaremos una variable para contar las líneas e interrumpir el ciclo
for l in lineas:
    print(l.split(" ")) #creamos una lista por cada línea cortando por los␣
 ↪espacios: " "
    stop += 1
    if stop==5: # Romperemos la ejecución luego de 5 iteraciones
        break   # para hacer más simple la visualización de la salida
                # ¿Qué pasaría si eliminas este condicional?
```

```
['\ufeffFrankenstein;\n']
['\n']
['or,', 'the', 'Modern', 'Prometheus\n']
['\n']
['by', 'Mary', 'Wollstonecraft', '(Godwin)', 'Shelley\n']
```

1.1.3 Cierre de archivos

Una vez terminemos de trabajar con nuestro archivo, es necesario usar la función `close` para que python libere la memoria, esto nos permitirá reabrir un archivo que teníamos de escritura para lectura o simplemente para mejorar el rendimiento de nuestro programa en las instrucciones restantes. Siempre es una buena práctica cerrar los archivos que abramos.

[ ]:
```
file.close() #liberamos la memoria
```

1.1.4 Creación de un archivo utilizando el modo escritura

Es posible también crear archivos de cero en caso de que no existan dentro de nuestro ordenador, en el siguiente ejemplo crearemos un archivo y escribiremos en él.

[ ]:
```
archivo = open('numeros.txt','w')
archivo.write("cuadrados y cubos de los números del 1 al 15\n")
```

```
for i in range(1,16): #es importante añadir el salto de línea al final de cada write
    archivo.write(f'{i}^2 = {i**2}, {i}^3 = {i**3}\n')
archivo.close()
```

Leamos el archivo que creamos

[ ]:
```
resultado = open('numeros.txt','r')
print(resultado.read())
resultado.close()
```

```
cuadrados y cubos de los números del 1 al 15
1^2 = 1, 1^3 = 1
2^2 = 4, 2^3 = 8
3^2 = 9, 3^3 = 27
4^2 = 16, 4^3 = 64
5^2 = 25, 5^3 = 125
6^2 = 36, 6^3 = 216
7^2 = 49, 7^3 = 343
8^2 = 64, 8^3 = 512
9^2 = 81, 9^3 = 729
10^2 = 100, 10^3 = 1000
11^2 = 121, 11^3 = 1331
12^2 = 144, 12^3 = 1728
13^2 = 169, 13^3 = 2197
14^2 = 196, 14^3 = 2744
15^2 = 225, 15^3 = 3375
```

También, si queremos, podemos visualizarlo línea por línea usando el ciclo `for`

[ ]:
```
texto = open('numeros.txt','r')
for l in texto.readlines():
    print(l)
texto.close()
```

```
cuadrados y cubos de los números del 1 al 15

1^2 = 1, 1^3 = 1

2^2 = 4, 2^3 = 8

3^2 = 9, 3^3 = 27

4^2 = 16, 4^3 = 64

5^2 = 25, 5^3 = 125
```

```
6^2 = 36, 6^3 = 216

7^2 = 49, 7^3 = 343

8^2 = 64, 8^3 = 512

9^2 = 81, 9^3 = 729

10^2 = 100, 10^3 = 1000

11^2 = 121, 11^3 = 1331

12^2 = 144, 12^3 = 1728

13^2 = 169, 13^3 = 2197

14^2 = 196, 14^3 = 2744

15^2 = 225, 15^3 = 3375
```

1.2 Uso de la sentencia `with`

La sentencia `with` nos permite crear un bloque de código donde se puedan ejecutar algunos comandos y liberar la memoria luego de terminar su ejecución, en el caso del manejo de archivos, esto nos permite abrir temporalmente un archivo y cerrarlo tan pronto como se acabe el bloque de código, esto nos ahorra el uso de la función `close`

```
with operación as variable:
    instrucciones
```

la sentencia `as` nos permite asignarle una variable a la operación que invocamos, por ejemplo, un `open`. El siguiente bloque de código es equivalente al ejemplo anterior.

[ ]:
```
with open('numeros.txt','r') as texto:
    for l in texto.readlines():
        print(l)
```

```
cuadrados y cubos de los números del 1 al 15

1^2 = 1, 1^3 = 1

2^2 = 4, 2^3 = 8

3^2 = 9, 3^3 = 27

4^2 = 16, 4^3 = 64

5^2 = 25, 5^3 = 125
```

```
6^2 = 36, 6^3 = 216

7^2 = 49, 7^3 = 343

8^2 = 64, 8^3 = 512

9^2 = 81, 9^3 = 729

10^2 = 100, 10^3 = 1000

11^2 = 121, 11^3 = 1331

12^2 = 144, 12^3 = 1728

13^2 = 169, 13^3 = 2197

14^2 = 196, 14^3 = 2744

15^2 = 225, 15^3 = 3375
```