

A thick black L-shaped frame surrounds the text. It starts at the top left, goes right, then down, then right again at the bottom right.

# INTRODUCCIÓN A LA PROGRAMACIÓN ARCHIVOS

Rodrigo López A.

[rilopez3@uc.cl](mailto:rilopez3@uc.cl)

# Archivos

- En la medida que trabajamos un proyecto en Python, comúnmente nos encontraremos con la necesidad de trabajar con grandes cantidades de datos, los cuales necesitamos guardar en un repositorio, como un archivo, para posteriormente leerlos y utilizarlos.
- En esta unidad trabajaremos con archivos para leer y guardar datos.

- En Python, para abrir un archivo usaremos la función `open`, que recibe el nombre del archivo a abrir y le debemos indicar la opción de lectura 'r' (existen otras opciones que veremos la próxima clase).

```
archivo = open("archivo.txt", 'r')
```

```
[3] 1 archivo = open('file_01.txt', 'r')
```

- Esta función intentará abrir el archivo con el nombre indicado. Si tiene éxito, devolverá una variable que nos permitirá manipular el contenido del archivo.
- Al terminar de trabajar con un archivo, es recomendable cerrarlo, en algunos sistemas los archivos sólo pueden ser abiertos de a un programa por la vez; en otros, lo que se haya escrito no se guardará realmente hasta no cerrar el archivo; o el límite de cantidad de archivos que puede manejar un programa puede ser bajo, etc.
- Para cerrar un archivo simplemente se debe llamar a:

```
archivo.close()
```

```
1 archivo.close()
```

```
1 archivo.readline()
```

-----  
ValueError Traceback (most recent call last)

<ipython-input-8-8a20e031b507> in <module>()  
----> 1 archivo.readline()

ValueError: I/O operation on closed file.

SEARCH STACK OVERFLOW

- Es posible leer un archivo línea por línea de la siguiente forma:

```
línea=archivo.readline()
```

```
while línea != "":
```

```
    línea=archivo.readline()
```

- Esto funciona ya que cada archivo que se encuentre abierto tiene una posición asociada, que indica el último punto que fue leído.
- Cada vez que se lee una línea, avanza esa posición. Es por ello que readline() devuelve cada vez una línea distinta y no siempre la misma.

```
1  línea = archivo.readline()
2  print(línea)
3  línea = archivo.readline()
4  print(línea)
```

Darth

Luke

- Es posible, además, obtener todas las líneas del archivo utilizando una sola llamada a función:

```
lineas = archivo.readlines()
```

- En este caso la variable líneas es un lista con todas las líneas almacenadas en su interior.

✓  
0 s



```
1 lineas = archivo.readlines()
2 print(lineas)
```

```
['Darth\n', 'Leia\n', 'Darth\n', 'Leia\n', 'Leia\n', 'Luke\n', 'Darth\n', 'Leia\n', 'Darth\n', 'Darth\n', 'Leia\n', 'Leia\n', 'Darth\n', 'Leia\n']
```

```
[ ] 1 len(lineas)
```

```
100
```

```
[ ] 1 print(lineas[0])
```

```
Darth
```

- Es bastante común que las líneas tengan caracteres de salto de línea al terminar, o pueden tener otro carácter que no necesitamos: espacios, puntos...
- En estos casos resulta útil el método de strings `STRIP`, el cual eliminara al comienzo y termino de un string el conjunto de caracteres que indiquemos (si no se define eliminara los espacios).

```
1 lineas = archivo.readlines()
2 print(lineas)
```

```
['Darth\n', 'Leia\n', 'Darth\n', 'Leia\n', 'Leia\n', 'Luke\n', 'Darth\n', 'Leia\n', 'Darth\n', 'Darth\n', 'Leia\n', 'Leia\n', 'Darth\n', 'Leia\n']
```

```
1 for i in range(len(lineas)):
2     lineas[i] = lineas[i].strip('\n')
3 print(lineas)
```

```
['Darth', 'Leia', 'Darth', 'Leia', 'Leia', 'Luke', 'Darth', 'Leia', 'Darth', 'Darth', 'Leia', 'Leia', 'Darth', 'Leia', 'Darth', 'Leia',
```

# Escritura de archivos

- Para escribir, crear o agregar contenido a un archivo, es necesario seguir un proceso similar al que realizamos para leerlos.
- Abrir un archivo  
`archivo = open('archivo.txt', opcion )`
- Escribir en el archivo  
`archivo.write('Test')`
- Cerrar el archivo  
`archivo.close()`
- Para escribir en un archivo se utilizan 2 opciones:
  - *w: permite sobre escribir en el archivo*
  - *a: permite escribir al final del archivo*
- En ambos casos si el archivo no existe es creado

# Escribiendo con .write(texto)

- El método write() permite escribir sobre el archivo objetivo
- Debe estar el archivo abierto
- Solo permite escribir strings
- Recuerda cerrar el archivo o no veras lo que has escrito

```
1 archivo = open('ejemplo.txt','w')
2 archivo.write('Hola')
3 archivo.write('Mundo')
4 archivo.write('\n')
5 archivo.write('Como')
6 archivo.write(' va')
7 archivo.write(' todo.')
8 archivo.close()
```

ejemplo.txt X

```
1 HolaMundo
2 Como va todo.
```



# Opción w versus opción a

```
1 archivo = open('ejemplo.txt','w')
2 archivo.write('Hola')
3 archivo.write('Mundo')
4 archivo.close()
5
6 archivo = open('ejemplo.txt','w')
7 archivo.write('Como')
8 archivo.write(' va')
9 archivo.write(' todo.')
10 archivo.close()
```

ejemplo.txt X

```
1 Como va todo.
```

```
1 archivo = open('ejemplo2.txt','a')
2 archivo.write('Hola')
3 archivo.write('Mundo')
4 archivo.close()
5
6 archivo = open('ejemplo2.txt','a')
7 archivo.write('Como')
8 archivo.write(' va')
9 archivo.write(' todo.')
10 archivo.close()
```

ejemplo.txt X ejemplo2.txt X

```
1 HolaMundoComo va todo.
```

# Escribiendo con print... WHATT

- Es posible escribir con la función `print()`, agregando el parámetro extra `file`
- El `file` debe ser un archivo abierto
- Agrega de forma automática saltos de línea
- Solo permite escribir strings

```
1  archivo = open('ejemplo3.txt','a')
2  print('Hola', file = archivo)
3  print('Mundo', file = archivo)
4  archivo.close()
5
6  archivo = open('ejemplo3.txt','a')
7  print('Como', file = archivo)
8  print('va', file = archivo)
9  print('todo.', file = archivo)
10 archivo.close()
```

```
1  Hola
2  Mundo
3  Como
4  va
5  todo.
6
```

# .write() versus print()

```
1  archivo = open('ejemplo2.txt','a')
2  archivo.write('Hola')
3  archivo.write('Mundo')
4  archivo.close()
5
6  archivo = open('ejemplo2.txt','a')
7  archivo.write('Como')
8  archivo.write(' va')
9  archivo.write(' todo.')
10 archivo.close()
```

ejemplo.txt X ejemplo2.txt X

```
1  HolaMundoComo va todo.
```

```
1  archivo = open('ejemplo3.txt','a')
2  print('Hola', file = archivo)
3  print('Mundo', file = archivo)
4  archivo.close()
5
6  archivo = open('ejemplo3.txt','a')
7  print('Como', file = archivo)
8  print('va', file = archivo)
9  print('todo.', file = archivo)
10 archivo.close()
```

```
1  Hola
2  Mundo
3  Como
4  va
5  todo.
6
```