IIC1103 - Sección 2

Clase 18: Archivos 2

Prof. Francisca Cattan

Universidad Católica de Chile

20 de octubre, 2022

Repaso corto Strings: remover en extremos

- `s.strip(c)`: Retorna `s` sin `c` sólo a ambos costados.
- `s.lstrip(c)`: Retorna `s` sin `c` a la izquierda.
- `s.rstrip(c)`: Retorna `s` sin `c` a la derecha.

```
1 s = "poder\tes poder\n"
2
3 print(s.strip())     # quita el \n final
4 print(s.strip("poder")) # quita la 1ra aparicion de 'poder'
5 print(s.rstrip()) # quita \n \t y espacios al final
6 print(s)
```

```
poder   es poder
  es poder


poder   es poder
poder   es poder
```

Repaso corto Strings: reemplazar

- `s.replace(`c_1`,` c_2`)`: Retorna un string igual a `s`, pero reemplazando cada ocurrencia de c_1 por c_2.

```
1 s = "Primera, frase.\nSegunda frase.\nTercera frase."
2
3 # cambia una frase por otra
4 print(s.replace("\n", ""))
5
6 # quita los espacios
7 print(s.replace(" ", ""))
```

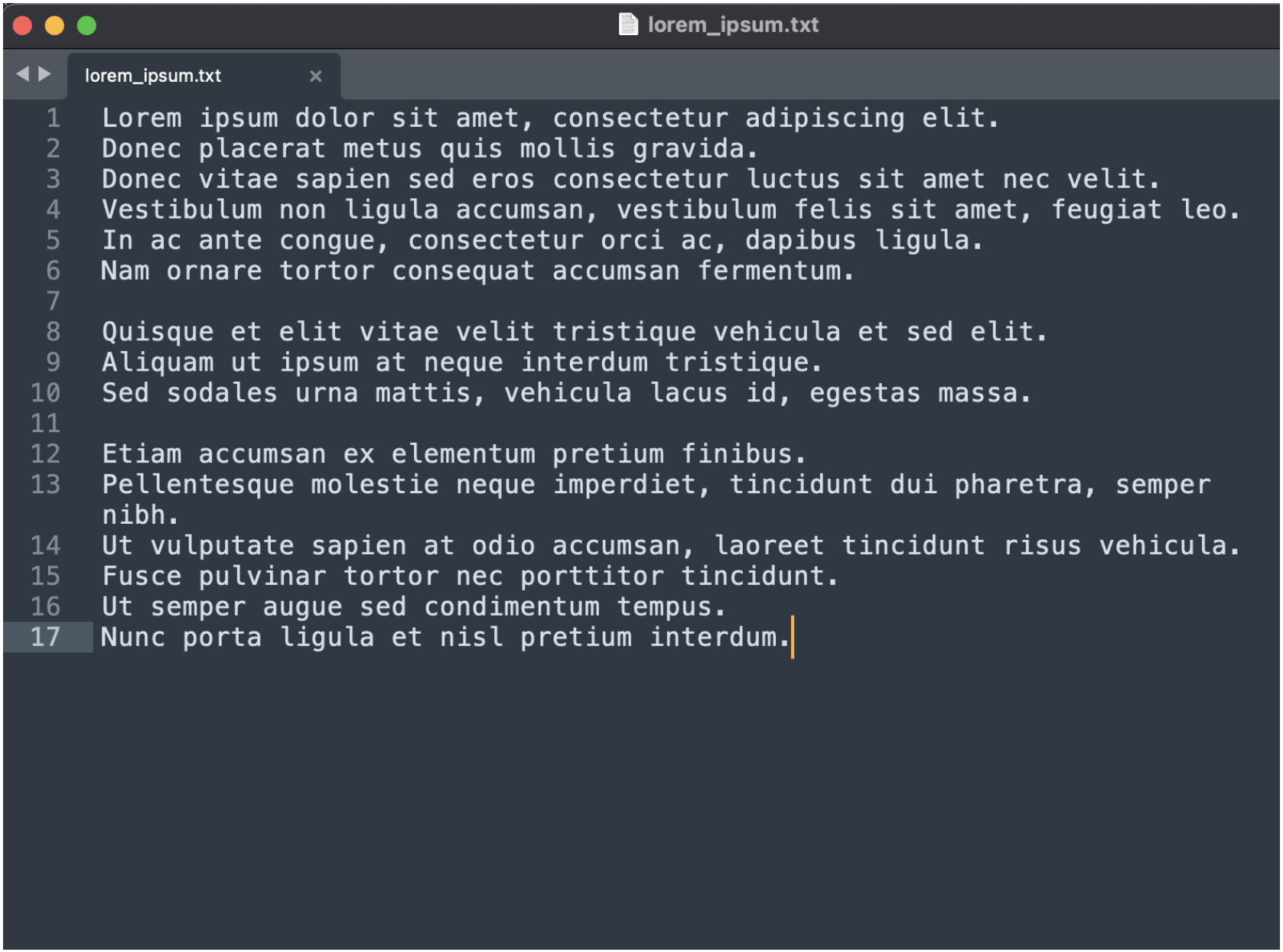
```
Primera, frase.Segunda frase.Tercera frase.
Primera,frase.
Segundafrase.
Terceraf rase.
```

Archivos

Archivos

- Lectura de archivos.
- Escritura de archivos.

Archivos: lectura de archivos



Archivos: lectura de archivos

Antes de cualquier acción sobre un archivo debemos abrirlo.

- `f = open(p)`: Función que recibe una ruta a un archivo y retorna un objeto `f` del tipo `io.TextIOWrapper`.

Una vez que está abierto podemos ejecutar métodos sobre ese objeto:

- `f.readline()`: Lee la siguiente línea del archivo y retorna esa línea en un string.
- `f.readlines()`: Retorna una lista con todas las líneas.

Al finalizar nuestro código SIEMPRE debemos cerrar el/los archivo/s.

- `f.close()`: Cierra un archivo.

Archivos: lectura de archivos

Que hace el comando **readline()**:

- Lee hasta el siguiente '\n'.
- Avanza automáticamente a la siguiente línea. Tiene un contador interno.
- La línea retornada incluye un '\n' al final.
- Final del archivo es un string vacío ''.

Que hace el comando **readlines()**:

- Retorna en una variable del tipo lista todo el contenido del archivo.
- En ambos casos, hay que tener consideración de los '\n' al final del archivo.

Archivos: lectura de archivos

```
1 f = open('./lorem_ipsum.txt')
2 lineas  = f.readlines()
3 f.close()
4 print(lineas)
```

```
[['Lorem ipsum dolor sit amet, consectetur adipiscing elit.\n',
'Donec placerat metus quis mollis gravida.\n', 'Donec vitae
sapien sed eros consectetur luctus sit amet nec velit.\n',
'Vestibulum non ligula accumsan, vestibulum felis sit amet,
feugiat leo.\n', 'In ac ante congue, consectetur orci ac,
dapibus ligula.\n', 'Nam ornare tortor consequat accumsan
fermentum.\n', '\n', 'Quisque et elit vitae velit tristique
vehicula et sed elit.', 'Aliquam ut ipsum at neque interdum
tristique.\n', 'Sed sodales urna mattis, vehicula lacus id,
egestas massa.\n', '\n', 'Etiam accumsan ex elementum pretium
finibus.\n', 'Pellentesque molestie neque imperdiet, tincidunt
dui pharetra, semper nibh.\n', 'Ut vulputate sapien at odio
accumsan, laoreet tincidunt risus vehicula.\n', 'Fusce pulvinar
tortor nec porttitor tincidunt.\n', 'Ut semper augue sed
```

Archivos: escritura de archivos

`f = open(p,m)`: Función que recibe un archivo en modo `m`.

- `"r"` – Modo lectura (por defecto).
- `"w"` – Modo escritura.
- `"a"` – Modo append.

Dado eso, si trabajamos en el modo escritura, podemos hacer más cosas con el archivo.

- `f.write(s)`: Escribe `s` en el archivo (sin fin de línea).
- `print(s, file=f)`: Escribe `s` en el archivo.

Archivos: escritura de archivos

¿Cual es la diferencia entre modo escritura y modo append?

- `f = open(p,'w')`: Crea un nuevo archivo en la ruta `p`. Si el archivo ya existe, borra su contenido.
- `f = open(p,'a')`: Crea un nuevo archivo en la ruta `p`. Si el archivo ya existe, lo lee y agrega su contenido al final.

Archivos: escritura de archivos

Ejemplo: Escribir un archivo con números del 1 al 5.

```
1 f = open('./number_test.txt', 'w')
2 for i in range(1,6):
3     f.write(str(i) + "\n")
4 f.close()
```

Obs: A diferencia de `print()`, el método `write()` no hace un salto de línea automático. Es por eso que en este código lo estamos agregando al escribir cada línea.

Archivos: escritura de archivos

Ejemplo: Escribir un archivo con números del 1 al 5.

```
1 f = open('./number_test.txt', 'w')
2 for i in range(1,6):
3     print(i, file=f)
4 f.close()
```

Si usamos la instrucción `print()` con este segundo parámetro, en vez de mostrar el contenido en consola (temporal) lo va guardando en un archivo.

Archivos: escritura de archivos

Resultado luego de ejecutarlo en modo 'w'.



Archivos: escritura de archivos

¿Qué ocurre si hacemos lo mismo a continuación, pero en modo append? Consideremos que el archivo ya está creado.

```
1 f = open('./number_test.txt', 'a')
2 for i in range(1,6):
3     print(i, file=f)
4 f.close()
```

Archivos: escritura de archivos

Resultado luego de ejecutarlo en modo 'a'.

