

Camino al Certamen 3: Resumen de Contenidos

Procesamiento de Texto		
Descripción	Comando	Ejemplo
Reemplazar s1 por s2 en un string	<code>string.replace(s1, s2)</code>	<code>"Programacion".replace("a", "x") => "Progrxmxcion"</code>
Cortar un string utilizando un <i>separador</i>	<code>string.split(separador)</code>	<code>"Programacion".split("a") => ["Progr", "m", "cion"]</code>
Unir lista de strings utilizando <i>pegamento</i>	<code>pegamento.join(lista)</code>	<code>"a".join(["Progr", "m", "cion"]) => "Programacion"</code>
El string tiene sólo mayúsculas?	<code>string.isupper()</code>	<code>"Programacion".isupper() => False</code>
El string tiene sólo minúsculas?	<code>string.islower()</code>	<code>"Programacion".islower() => False</code>
Convertir todo un string a mayúsculas	<code>string.upper()</code>	<code>"Programacion".upper() => "PROGRAMACION"</code>
Convertir todo un string a minúsculas	<code>string.lower()</code>	<code>"Programacion".lower() => "programacion"</code>
Intercambiar minúsculas y mayúsculas	<code>string.swapcase()</code>	<code>"Programacion".swapcase() => "pROGRAMACION"</code>
El <i>string</i> comienza con s1?	<code>string.startswith(s1)</code>	<code>"Programacion".startswith("Progra") => True</code>
El <i>string</i> termina con s1?	<code>string.endswith(s1)</code>	<code>"Programacion".endswith("ion") => True</code>
s1 está contenido dentro de s2?	<code>s1 in s2</code>	<code>"ama" in "Programacion" => True</code>
Eliminar saltos de línea de un <i>string</i>	<code>string.strip()</code>	<code>"Programacion\n".strip() = "Programacion"</code>

Manejo de Archivos		
Descripción	Comando	Explicación
Abrir un archivo	<code>archivo = open(nombre, m)</code>	La variable <i>archivo</i> hará de "puente" entre Python y nuestro archivo. Si el nombre de archivo no existe, arroja error. La variable m denota el modo de apertura: 'w' para escritura desde el inicio del archivo (sobrescribe todo), 'a' para escritura desde el fin del archivo (agrega al final de este), vacío para lectura.
Cerrar un archivo	<code>archivo.close()</code>	Paso opcional (Python cierra solo el archivo), pero recomendado.
Leer un archivo	<code>for linea in archivo</code>	Toma cada línea del archivo, y la deja en la variable linea. Ojo con los saltos de línea! Elimínalos antes de empezar a trabajar.
Escribir un archivo	<code>archivo.write(linea)</code>	Escribe la <i>linea</i> en el <i>archivo</i> especificado. El comando no agrega saltos de línea, así que ojo cuando desees escribir múltiples líneas!

Creado por Camilo Saldías

<https://github.com/csaldias/python-usm>

Recetario para el Certamen 3

Abrir y leer un archivo

```
Archivo = open(nombre_archivo)
For linea in archivo:
    datos = linea.strip().split(caracter)]
```

Abrir un archivo y agregar datos al final

```
Archivo = open(nombre_archivo, 'a')
Archivo.write(datos+'\n')
#datos (string) se agrega al final de archivo
```

Crear un archivo y guardar datos

```
Archivo = open(nombre_archivo, 'w')
Archivo.write(datos+'\n')
#datos debe ser un string!
```

Desempaquetar una línea con múltiples separadores

```
#Asumiremos linea = "dato1;dato2:dato3,dato4"
dato1, resto = linea.split(";", 1)
dato2, resto = resto.split(":", 1)
dato3, dato4 = resto.split(",", 1)
```

Consideraciones especiales

- Recuerda que `.write()` no agrega el salto de línea por defecto! Si deseas que el dato siguiente quede en la siguiente línea (que es como apretar Enter), agrega `'\n'` al final del string que vas a grabar.
- Si estás leyendo una línea, no puedes volver a leer la anterior (osea, no puedes devolverte), porque la lectura se hace de forma secuencial. Para eso, debes cerrar el archivo y volverlo a abrir.
- Todas las funciones de procesamiento de texto devuelven un nuevo string como resultado. En ningún caso modifican el string original.
- La función `.strip(caracter)` permite eliminar caracteres en los "bordes" de un string, no sólo saltos de línea. Por ejemplo, `"asdf1234asdf".strip("asdf") -> "1234"`
- Si deseas hacer un número determinando de cortes a un string, puedes especificarle el número máximo de cortes a la función `split()`. Quedaría entonces `split(separador, max_cortes)`. Con eso, la función sólo corta basado en los primeros *max_cortes* separadores, de izquierda a derecha.

Creado por Camilo Saldías

<https://github.com/csaldias/python-usm>