




8 Gestión de archivos y gestores de contexto.

Objetivo. ...

Funciones de Python: ...

[MACTI-Algebra_Lineal_01](#) by [Luis M. de la Cruz](#) is licensed under [Attribution-ShareAlike 4.0 International](#) 

9 Gestión de archivos

La función para gestionar archivos es `open()`. Toma dos parámetros: el nombre del archivo y el modo. Existen cuatro diferentes modos:

- `"r"` - Read - Default value. Abre el archivo para lectura. Se produce un error si el archivo no existe.
- `"a"` - Append - Abre el archivo para agregar información. Si el archivo no existe, lo crea.
- `"w"` - Write - Abre el archivo para escritura. Si el archivo no existe, lo crea. Si el archivo existe, lo sobrescribe.
- `"x"` - Create - Crea el archivo, regresa un error si el archivo existe.

Adicionalmente se puede especificar si el archivo se abre en modo texto o binario:

- `"t"` - Text - Valor por omisión.
- `"b"` - Binary

```
# Abrimos el archivo gatos.txt en modo escritura
# Ojo: si el archivo existe, lo sobrescribe.
f = open('gatos.txt', 'w')
```

```
# Construimos un diccionario con información
gatos = ['Persa', 'Sphynx', 'Ragdoll', 'Siamés']
peso_minimo = [2.3, 3.5, 5.4, 2.5]
dicc = dict(zip(gatos, peso_minimo))
print(dicc)
```

```
{'Persa': 2.3, 'Sphynx': 3.5, 'Ragdoll': 5.4, 'Siamés': 2.5}
```

```
# Guardamos cada elemento del diccionario en el archivo
for i in dicc:
    f.write('{:>10} {:>10} \n'.format(i, dicc[i]))

# Es importante siempre cerrar el archivo cuando ya no se use.
f.close()
```

```
# Ahora abrimos el archivo en modo lectura
# El archivo debe existir, de otro modo se genera un error.
```

```
f = open('gatos.txt', 'r')
for i in f:
    print(i)

f.close()
```

Persa	2.3
Sphynx	3.5
Ragdoll	5.4
Siamés	2.5

10 Gestores de contexto

Permiten asignar y liberar recursos justo en el momento requerido. Se usa mayormente con `with`, veamos un ejemplo:

```
with open('contexto_ejemplo', 'w') as archivo_abierto:
    archivo_abierto.write('En este ejemplo, todo se realiza con un gestor de cont
```

Lo que realiza la operación anterior es: 1. Abre el archivo `contexto_ejemplo`. 2. Escribe algo en el archivo, 3. Cierra el archivo. Si ocurre algún error mientras se escribe en el archivo, entonces se intenta cerrar el archivo.

El código es equivalente a:

```
archivo_abierto = open('contexto_ejemplo', 'w')
try:
    archivo_abierto.write('En este ... contexto!')
finally:
    archivo_abierto.close()
```

Es posible implementar nuestros propios gestores de contexto. Para ello se requiere un conocimiento más profundo de Programación Orientada a Objetos.