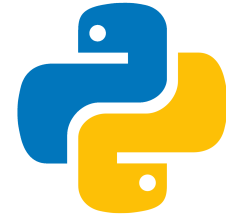


Tractament de fitxers en Python



1. Tractament de fitxers
2. Lectura de fitxers
3. Creació i escriptura
4. Esborrat de fitxers

Tractament de fitxers



open()

- Funció **open()** forma part de la base de Python.
- Té dos paràmetres d'entrada: **nom del fitxer** i **tipus d'obertura del fitxer**. Podem usar rutes relatives o absolutes per especificar fitxer.
- Quatre obertures de fitxer:
 - i. "r" – Mode lectura. Valor per defecte. Mostrarà error si el fitxer no existeix o no accessible.
 - ii. "a" – Mode *append* per afegir contingut al final. En el cas de que no existeixi l'arxiu especificat, l'intenta crear.
 - iii. "w" – Mode escriptura amb sobreescriptura. Intenta crear l'arxiu si no existeix.
 - iv. "x" – Mode creació. Intenta crear un nou fitxer buit amb el nom i la ruta especificades. Susceptible de retornar errors per diversos motius.



open()

- Modificadors d'obertura:
 - i. `"t"` – Especifica que el fitxer conté text.
 - ii. `"b"` – Arxius que NO siguin de tipus text. Es tracten com a binaris.
- L'obertura del fitxer s'assigna a una variable no declarada.
- Exemple:

```
>>> f = open("test.txt", "a")
>>> f = open("test.txt", "wt")
>>> f.write("MP15-RA1")
7
>>> f = open("test.txt", "r")
>>> print(f.read())
MP15-RA1
```

Lectura de fitxers



read()

- Abans de llegir, obrir el fitxer en mode lectura:
open(ruta_fitxer/nom_fitxer, "r") i assignar-ho a una variable.
- El **polimorfisme** i **tipatge dinàmic** de Python permet que la variable on hem obert el fitxer s'haurà creat amb el tipus <class '_io.TextIOWrapper'>
- Usem la funció **read()** per llegir el contingut de l'arxiu
- Podem llegir tot el fitxer amb read() o **un número de files** amb read(n).

```
>>> f = open("test.txt", "r")
>>> type(f)
<class '_io.TextIOWrapper'>
>>> print(f.read())
MP15-UF1
```



readline()

- La funció **readline()** llegeix línia a línia començant per la primera.
- Usa un punter per marcar on es troba dins del fitxer.
- Podem iterar el contingut amb un for-loop:

```
>>> f = open("test.txt", "r")
>>> for x in f:
...     print(x)
...
```



close()

- Tancarem el fitxer per alliberar el recurs del HD.

```
>>> f.close()
```


Creació - escriptura



write()

- Usarem la funció **write()**.
- Aquesta funció requereix passar-li el text a afegir entre cometes dobles.
- Podem usar caràcters no imprimibles tals com tabulador, salt de línia, ... etc escapant els seus valors con \. Per exemple: \n, \t.

Esborrat de fitxers



os.remove()

- Per esborrar un arxiu (o directori) usarem la funció **remove()** de la llibreria **os** (*operating system*) de Python.

- Importarem la llibreria:

```
>>> import os
```

- Usarem el mètode estàtic **remove()** de la classe **os** passant-li la ruta i el nom de l'arxiu per paràmetre:

```
>>> os.remove("test.txt")
```



os.path.exists()

- Si l'arxiu no existeix, rebrem un error:

```
>>> os.remove("test.txt")
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
FileNotFoundError: [WinError 2] The system cannot find the file specified: 'test.txt'
```

- Per evitar-ho, usarem la funció: `os.path.exists()` per comprovar si l'arxiu existeix.

```
>>> if os.path.exists("test.txt"):
```

```
...     os.remove("demofile.txt")
```

```
... else:
```

```
...     print("The file does not exist")
```

```
...
```

```
The file does not exist
```



`os.rmdir()`

- Per esborrar un directori sencer (i el seu contingut), usarem el mètode estàtic `rmdir()` de la classe `os`.

```
>>> os.rmdir("TMP2")
```

- Abans d'esborrar un directori, haurem de comprovar que aquest existeix prèviament.
- **Nota:** *Compte amb les proves que facis amb el tractament de fitxers, ja que Python no pregunta abans d'esborrar un directori sencer. Si s'executa la instrucció, s'esborra immediatament.*



*“Un gran **poder**
comporta una
gran
responsabilitat”*





“Ser lliure és no seguir a ningú”

Kilian Jornet

Raimon Izard

