

# File\_IO

## عمليات الملفات

الملفات عبارة عن أماكن مُسماة على ذاكرة التخزين ، بداخلها يتم حفظ البيانات لإسترجاعها حسب الحاجة . يتم استخدامها لحفظ البيانات بشكل دائم .

لغة البايثون تتيح لنا ادوات للتعامل مع الملفات كالكتابة و القراءة . لتنفيذ عملية على ملف في البايثون ، نمر بثلاث خطوات :

- فتح الملف
- تنفيذ العملية (قراءة أو كتابة أو انشاء)
- اغلاق الملف

### فتح ملف

لفتح ملف نستخدم الدالة `open` . هذه الدالة تأخذ مسار الملف (`path`) المراد فتحه كُعطى بالإضافة الى معطيات اختيارية اخرى . ستقوم الدالة بإرجاع كيان ملف `file object` . بحيث بعدها يمكننا استخدامه لتنفيذ العمليات المطلوبة على الملف .

```
file = open("test.txt") # open file in current directory
file = open("Users/user-name/Documents/test.txt") # or provide full path
```

### أنماط فتح الملف

عند فتح ملف ما ، فهناك عدة أنماط للفتح من اجل تحديد نوع العملية المراد تنفيذها .

- `r` - النمط الافتراضي : القراءة

- نمط القراءة - يتم انشاء الملف في حال عدم وجوده ، او الكتابة عليه ان وجد : w
- نمط الإنشاء - يستخدم فقط لإنشاء الملف ، يفشل هذا النمط في حال كان الملف موجود مسبقاً : x
- نمط الإضافة - يفتح الملف للإضافة ، يتم انشاء الملف في حال عدم وجوده : a
- (النمط البيناري (الرقمي الثنائي : b
- يفتح الملف للقراءة و الكتابة ، موقع المؤشر في البداية : r+
- يفتح الملف للكتابة و القراءة ، يتم الكتابة على المحتويات الموجودة مسبقاً : w+
- يفتح ملف للإضافة و القراءة ، موقع المؤشر بآخر المحتوى : a+

## قراءة محتوى الملف

لقراءة محتوى الملف نستخدم الدالة `read` ، و التي ستقرأ المحتوى كاملاً و ترجعه لك كنص . اختياريًا يمكننا ان نزودها بمعطى لتحديد القدر المطلوب قراءتها من البايتات . يجب ان نكون في نمط القراءة لإستخدام هذه الدالة بنجاح.

```
file = open("test.txt", "r", encoding="utf-8")
content = file.read()
file.close()
```

## كتابة محتوى لملف

لكتابة محتوى لملف ، نستخدم الدالة `write` و التي تأخذ نص `str` كمعطى (المحتوى المراد كتابته) . يجب أن نكون في نمط الكتابة أو الإضافة لإستخدام هذه الدالة بنجاح .

```
file = open('test.txt', "w", encoding="utf-8")
file.write("hey how are you ?")
file.close()
```

يتم استخدام هذه الدوال من اجل معرفة موقع المؤشر بداخل الملف، أو من اجل التحكم بموقعه .

لمعرفة موقع المؤشر داخل الملف نستخدم `tell`

```
file.tell()
```

للتحكم بموقع المؤشر نستخدم `seek` و نعطيه الموقع الذي نريده فيه

```
file.seek(4)
```

## عمليات اخرى على الملفات

- `readline()` : يقرأ سطر واحد من الملف و يرجعه كنص
- `readlines()` : يقرأ الملف كقائمة من السطور و يرجعها
- `writelines(lines)`: دالة تستقبل قائمة من النصوص ، و من ثم تكتبها للملف

## عمليات المجلدات

ما هو المجلد ؟

المجلد عبارة مجموعة من الملفات و المجلدات الثانوية . نستخدم المجلدات لترتيب الملفات و الأصول داخل المشروع . من أجل أن ننفذ عملية ، يجب اولاً أن نستورد الوحدة `os`

معرفة المجلد العملي الحالي

```
import os
```

```
os.getcwd()
```

تغيير المجلد العملي

```
# give the path to the directory  
os.chdir('anotherDirectory')
```

لعرض محتويات المجلد

```
# list the content in the current working directory  
os.listdir()  
os.listdir('path/to/directory')
```

إنشاء مجلد جديد

```
os.mkdir("dir_name")
```

لإعادة تسمية ملف أو مجلد

```
os.rename('old_name','new_name')
```

حذف ملف

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

حذف مجلد (خال من المحتويات)

```
os.rmdir("folder_path")
```

حذف مجلد إذا كان بداخله محتويات

```
# import shutil first
import shutil
shutil.rmtree("folder_path")
```