

實用腳本！利用Python 對PDF 進行加密、解密操作，代碼拿走就用！

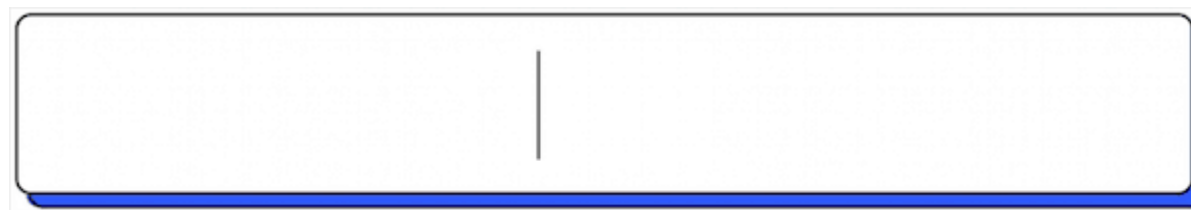
python爬蟲人工智能大數據 昨天

以下文章來源於早起Python，作者陳熹



早起Python

點擊領取Python數據分析200題，回復1



相關導讀

 **01 | Python批量合併PDF**

 **02 | PDF合併、拆分、水印、加密**

 **03 | Python提取PDF表格**

 **04 | Python提取PDF指定內容**



大家好，我是早起。

本文將分享如何利用Python對PDF進行加密和解密操作，主要利用到之前多次介紹過的 `PyPDF2` 模塊。

PDF 加密

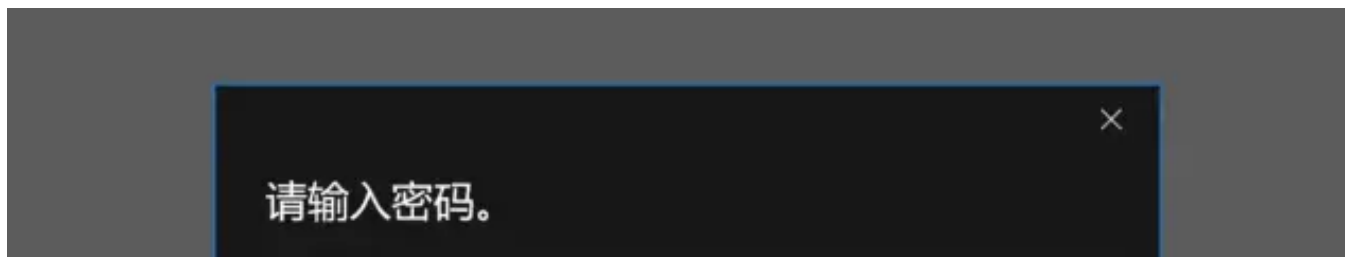
在之前的文章[PDF合併、拆分、水印、加密](#)中簡單提到過加密一個PDF文件的方法，我們先拿自己隨意的一個PDF文件試一下：

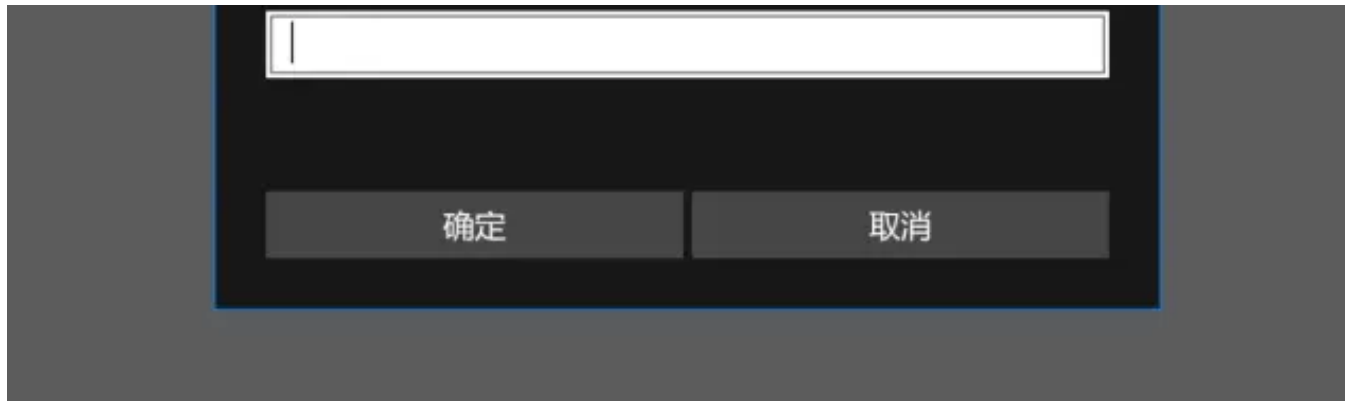
```
from PyPDF2 import PdfFileWriter, PdfFileReader
path = r'C:\xxx' # 这里填写目标 PDF 所在的路径

pdf_reader = PdfFileReader(path + r'\test.pdf')
pdf_writer = PdfFileWriter()

for page in range(pdf_reader.getNumPages()):
    pdf_writer.addPage(pdf_reader.getPage(page))
pdf_writer.encrypt('a123') # 设置密码为 a123
with open(path + r'\test.pdf', 'wb') as out:
    pdf_writer.write(out)
```

上面代碼的運行邏輯是：實例化一個PDF寫入器和讀取器，讀取器讀取完目標PDF文件後，一頁一頁交給寫入器，然後對寫入器設置密碼並輸出。看一下運行結果：





可見這個PDF 文件成功設置上了密碼，如果有多個PDF需要加密，可以寫一個簡單的循環利用上述腳本批量處理，此處不再展開說明。

PDF 已知密碼解密

如果知道密碼的情況下，想直接取消PDF的加密，可以用 `.decrypt`，解密的過程需要讀取器和寫入器共同配合。但區別於加密 `.encrypt`，`.decrypt`是針對讀取器進行解密的，而不是寫入器

```
from PyPDF2 import PdfFileWriter, PdfFileReader
path = r'C:\xxx'

pdf_reader = PdfFileReader(path + r'\test.pdf')
pdf_reader.decrypt('a123') #
pdf_writer = PdfFileWriter()

for page in range(pdf_reader.getNumPages()):
    pdf_writer.addPage(pdf_reader.getPage(page))
with open(path + r'\test.pdf', 'wb') as out:
    pdf_writer.write(out)
```

上面代码的运行逻辑是：首先用读取器读取加密文件，然后直接在读取器上使用 `.decrypt` 进行解密，逐页传到刚实例化的写入器后统一输出，就完成了解密

PDF 暴力解密

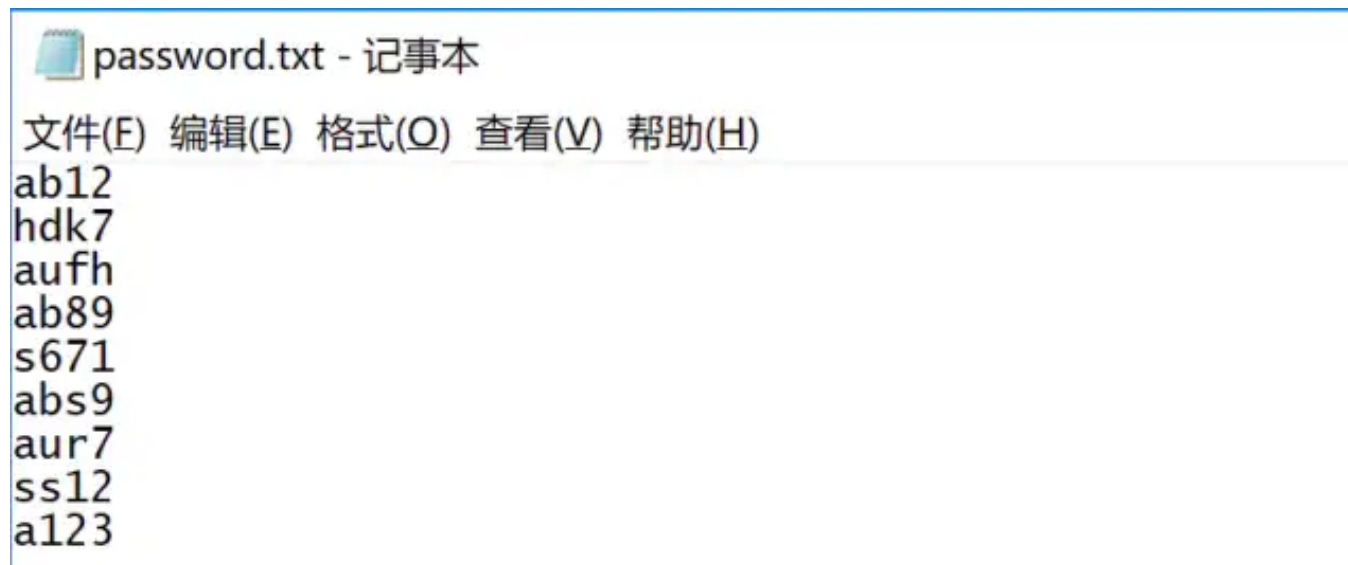
暴力破解，其实就是通过手上已经有的密码库，或者完全通过数字、字母、符号的穷举，将可能的密码都放进去，逐个尝试直到成功。

「目前暴力破解只适用于已知密码位数少，由简单的数字、字母构成」

假设今天要破解的密码就是 `a123`，已知密码 4 位且由数字和小写字母组成。破解又分为两种情况：

1. 手上有密码本

第一种情况，知道大概密码，可以整理成一个 `password.txt`，保证可能的密码一定在其中：



首先读取 `.txt` 文件获取其中所有密码（示例文件只有 9 个密码）：

```
passw = []
path = r'C:\Scientific Research\Python'
file = open(path + r'\password.txt')
for line in file.readlines():
    passw.append(line.strip())
print(passw)
file.close()
```

```
['ab12', 'hdk7', 'aufh', 'ab89', 's671', 'abs9',
 'aur7', 'ss12', 'a123']
```

```
Process finished with exit code 0
```

然后就可以用密码本的密码做暴力破解：

```
from PyPDF2 import PdfFileReader

passw = []
path = r'C:\xxx'
file = open(path + r'\password.txt')
for line in file.readlines():
    passw.append(line.strip())
file.close()

path = r'C:\xxx'
pdf_reader = PdfFileReader(path + r'\test\pdf')
```

```
pdf_reader = PdfFileReader(path + r'(test).pdf')
```

```
for i in passwd:  
    if pdf_reader.decrypt(i):  
        print(f'破解成功，密码为{i}')    else:  
        print(f'破解不成功，密码{i}错误')
```



2. 手术无密码本，完全穷举

首先介绍 `itertools`，可以构建含需要组合的生成器供后续迭代：

```
import itertools
```

```
mylist = ("".join(x) for x in itertools.product("0123456789abcdef", repeat=4))  
print(next(mylist))
```

利用这个方法就可以用 `while` 循环尝试暴力破解（为了减少时间，生成器用"abc123"生成组合）：

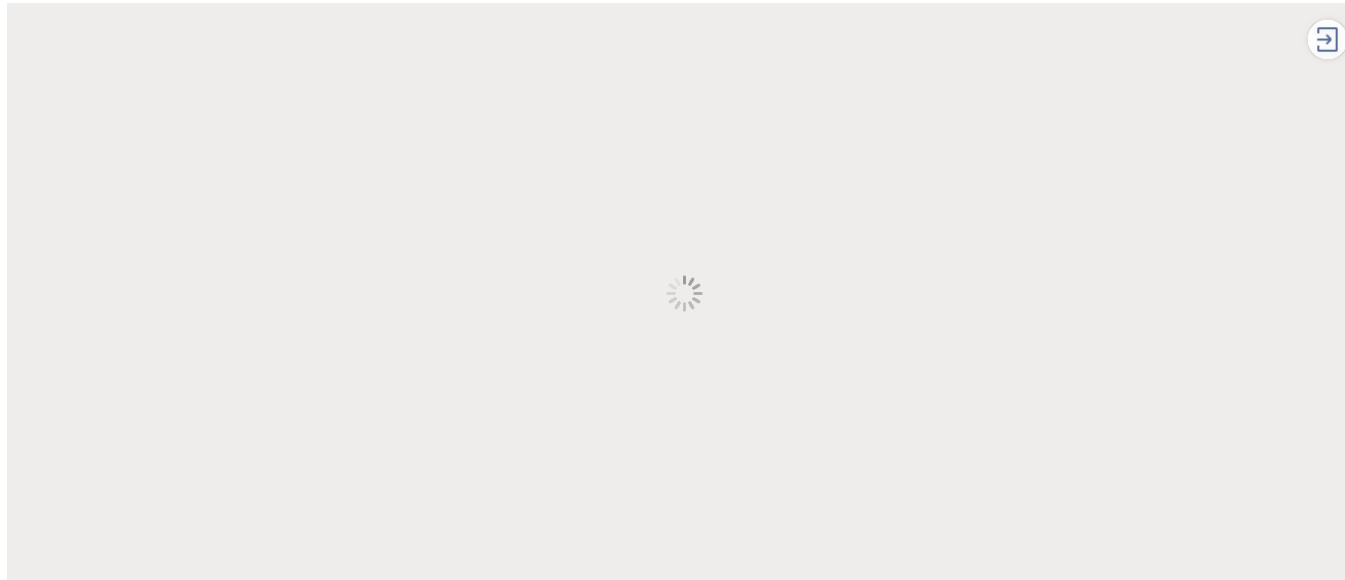
```
import itertools  
from PyPDF2 import PdfFileReader  
  
mylist = ("".join(x) for x in itertools.product("123abc", repeat=4))  
path = r'C:\xxx'  
pdf_reader = PdfFileReader(path + r'\test.pdf')  
  
while True:  
    i = next(mylist)  
    if pdf_reader.decrypt(i):  
        print(f'破解成功 · 密码为{i}')        break  
    else:  
        print(f'破解不成功 · 密码{i}错误')
```



可以看到最后也成功破解了，但需要注意：但即使是这么简单的组合也耗费了一点时间，如果复杂的密码破解难度和破解时间就指数级增长了，**暴力的穷举法就不一定适用**，可以根据需求具体调整规则，缩小范围再进行尝试。

-END-

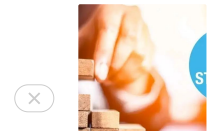
如果你希望学习更多Python办公自动化案例，可以点击下方图片查看！



喜欢此内容的人还喜欢

傻瓜式教程：如何使用“多合一”开发工具STM32CubeIDE

玩轉嵌入式



.NET5 開發手機提詞應用，基於內嵌Web服務器及PowerPoint自動化

dotNET跨平台



