

# کار با فایل

چند راه مختلف برای کار با فایل در پایتون وجود دارد که ساده‌ترین راه آن در این درسنامه توضیح داده شده است.

## کلمه‌ی کلیدی with

with کلمه ای کلیدی است که جلوی خود، یک متغیر یا منبع را باز می کند و پس از تمام شدن بلوک خود آن را می‌بندد و از حافظه پاک می‌کند. هر شیئی با پیاده سازی دو متد `__enter__` و `__exit__` می‌تواند جلوی with استفاده شود. در حقیقت برای مدیریت راحت‌تر منابع از with استفاده می‌شود.

```
1 # file handling
2
3 # 1) without using with statement
4 file = open('file_path', 'w')
5 file.write('hello world !')
6 file.close()
7
8 # 2) without using with statement
9 file = open('file_path', 'w')
10 try:
11     file.write('hello world')
12 finally:
13     file.close()
14 # using with statement
15 with open('file_path', 'w') as file:
16     file.write('hello world !')
```

- کد بالا ۳ روش برای مدیریت منابع هنگام باز کردن فایل را نشان می‌دهد. استفاده از with باعث می‌شود مدیریت منابع راحت‌تر شود. پس در ادامه از آن استفاده می‌کنیم.

## باز کردن فایل

`open(path, mode)`

تابع `open` مسیر و حالت باز کردن را می‌گیرد و شی فایل را برمی‌گرداند. آدرس می‌تواند **نسبی** یا **مطلق** باشد ؛ چند حالت برای باز کردن فایل وجود دارد که در زیر توضیح داده شده است.

حالت	توضیحات
r	این حالت پیش فرض است. فایل را برای خواندن باز می‌کند.
w	فایل را برای نوشتن باز می‌کند. اگر فایل وجود داشته باشد، آن را خالی می‌کند و اگر وجود نداشته باشد آن را ایجاد می‌کند.

توضیحات	حالت
فایل جدیدی ایجاد می‌کند. اگر فایل وجود داشته باشد عملیات شکست می‌خورد.	x
برای اضافه کردن به انتهای فایل است. اگر فایل وجود نداشته باشد، یک فایل جدید درست می‌کند.	a
فایل را به صورت متنی باز می‌کند.	t
فایل را به صورت دودویی (binary) باز می‌کند.	b
فایل را برای خواندن و نوشتن (به روز رسانی) باز می‌کند.	+

## خواندن فایل

```
1 | with open('myfile1.txt', "r+") as file:
2 |     contents = file.read()
```

متد `read` محتوای فایل را به صورت یک رشته برمی‌گرداند.

```
1 | with open("myfile.txt") as f:
2 |     for line in f:
3 |         print(line)
```

- با قرار دادن فایل در حلقه‌ی `for` می‌توان خط به خط فایل را خواند.

## نوشتن در فایل

```
1 | contents = {"aa": 12, "bb": 21}
2 | with open("myfile1.txt", "w") as file:
3 |     file.write(str(contents))
```

متد `write` یک رشته را در فایل می‌نویسد. برای نوشتن در فایل از حالت `w` استفاده شده است.

## کار با JSON

پایتون به خودی خود، کتابخانه‌ای برای کار با `JSON` دارد ولی باید در ابتدا به برنامه وارد (`import`) شود.

```
1 | import json
```

```
1 | with open("myfile2.txt", "w+") as file:
2 |     file.write(json.dumps(contents))
```

متد dumps یک آرایه یا دیکشنری یا nested list یا nested dictionary را می‌گیرد و رشته‌ی JSON آن را برمی‌گرداند.

- دیکشنری‌های پایتون در یکی از درسنامه‌های بعدی توضیح داده شده‌است.

```
1 | with open('myfile2.txt', "r+") as file:
2 |     contents = json.load(file)
```

متد load یک فایل را می‌گیرد و محتوی آن که یک JSON بوده را به صورت یک دیکشنری یا لیست پایتون برمی‌گرداند. این لیست یا دیکشنری‌ها می‌توانند تو در تو باشند.

## کار با csv

یکی از رایج‌ترین فرمت‌ها برای ذخیره‌ی جداول داده، csv (*comma separated values*) است. در خط اول نام ستون‌های داده که با ، از هم جدا شده می‌آید. و بعد از آن در هر خط یک رکورد از داده است که داده‌ها در آن به ترتیب ستون‌ها آمده‌اند و با ، از هم جدا شده‌اند.

مثالی از محتوای یک فایل csv

```
Lead,Title,Phone,Notes
Jim Grayson,Senior Manager,(555)761-2385,"Spoke Tuesday, he's interested"
Prescilla Winston,Development Director,(555)218-3981,said to call again next week
Melissa Potter,Head of Accounts,(555)791-3471,"Not interested, gave referral"
```

- **نکته:** کاماهای داخل " " برای جدا کردن به حساب نمی‌آیند.

در اینجا به قالب کلی فایل csv بسنده می‌شود. در قسمت‌های بعدی این فصل بیشتر با فایل csv کار خواهید کرد.