

## کار با فایل‌ها در پایتون:

پایتون امکانات گسترده‌ای برای کار با فایل‌ها فراهم می‌کند که شامل خواندن، نوشتن، و مدیریت انواع مختلف فایل‌ها است.

---

## باز کردن فایل در پایتون (تابع open):

برای کار با فایل‌ها در پایتون از تابع `open()` استفاده می‌شود که قالب کلی آن به شکل زیر است:

```
file = open("filename.txt", "mode")
```

code block

```
file.close()
```

**نکته:** بعد از باز کردن یک فایل باید آن را ببندیم تا به مشکل نخوریم

پس اول با تابع `open` یک فایل رو باز میکنیم کارهای مورد نیاز رو انجام میدیم در انتها با استفاده از متد `close` فایل را می‌بندیم

```
with open("filename.txt", "mode") as file:
```

code block

**نکته:** بعد از باز کردن فایل به شکل بالا دیگر نیازی به بستن فایل نداریم زیرا که خود دستور with بستن فایل را هندل میکند و اینکه کدهای مربوط به فایل رو با رعایت تو رفتگی باید بنویسیم

---

### آرگومان‌ها:

"filename.txt": نام یا آدرس فایل مورد نظر

"mode": مشخص کننده نوع عملیاتی که روی فایل انجام میشود

### مدهای (modes) باز کردن فایل:

مد	توضیح
"r"	خواندن - (Read) فایل را فقط برای خواندن باز می کند (پیش فرض). اگر فایل وجود نداشته باشد، خطا می دهد.
"w"	نوشتن - (Write) فایل را برای نوشتن باز می کند. اگر فایل وجود نداشته باشد، یک فایل جدید ایجاد می کند. اگر فایل وجود داشته باشد، محتوای آن را پاک می کند.
"a"	افزودن - (Append) فایل را برای افزودن محتوا باز می کند. اگر فایل وجود نداشته باشد، یک فایل جدید ایجاد می شود.

"x"	ایجاد - (Exclusive Creation) اگر فایل وجود نداشته باشد، یک فایل جدید ایجاد می کند. اگر فایل وجود داشته باشد، خطا می دهد.
"b"	باینری - (Binary Mode) فایل را در حالت باینری باز می کند. (مثلاً برای تصاویر، ویدئوها و غیره)
"t"	متنی - (Text Mode) فایل را در حالت متنی باز می کند (پیش فرض).
"r+"	خواندن و نوشتن در فایل بدون حذف محتوای قبلی.
"w+"	نوشتن و خواندن، اما محتوای فایل پاک می شود.
"a+"	خواندن و افزودن محتوا بدون حذف محتوای قبلی.

**نکته:** می توان مدهای باینری را با سایر مدها ترکیب کرد، مثلاً ( "rb" خواندن باینری ) ( "wb" نوشتن باینری )

### مدهای رایج کار با فایل ها:

پس از باز کردن فایل، می توان از متدهای زیر برای خواندن، نوشتن و مدیریت آن استفاده کرد.

### خواندن از فایل

```
file = open("example.txt", "r")
```

```
content = file.read() # خواندن کل محتوا
```

```
print(content)
```

```
file.close() # بستن فایل
```

### متدهای خواندن:

متد	توضیح
file.read(size)	خواندن کل محتوا یا تعداد size بایت کاراکتر.
file.readline()	خواندن یک خط از فایل.
file.readlines()	خواندن تمام خطوط و بازگرداندن آنها به عنوان لیست.

### مثال خواندن خط به خط:

```
file = open("example.txt", "r")
```

```
for line in file:
```

```
    print(line.strip()) # حذف کاراکترهای اضافی مثل '\n'
```

```
file.close()
```

---

### نوشتن در فایل:

# باز کردن فایل برای نوشتن (پاک شدن محتوای قبلی)

```
file = open("example.txt", "w")
file.write("Hello, World!\n")
file.write("This is a test.\n")
file.close()
```

### متدهای نوشتن:

متد	توضیح
file.write(string)	نوشتن یک رشته در فایل.
file.writelines(lines)	نوشتن یک لیست از خطوط.

### افزودن محتوا بدون حذف محتوای قبلی:

```
file = open("example.txt", "a")
file.write("Appending a new line.\n")
file.close()
```

---

### کار با فایل‌های باینری:

برای خواندن یا نوشتن فایل‌های باینری مانند تصاویر و ویدئوها از مد "b" استفاده می‌شود.

### مثال: کپی کردن یک تصویر

with open("image.jpg", "rb") as source:

with open("copy.jpg", "wb") as destination:

destination.write(source.read())

---

### نکات مهم:

- ✓ با مدهای مناسب استفاده کنید open() برای کار با فایل‌ها از ✓
  - ✓ را بشناسید readlines(), readline(), write(), read() متدهای ✓
  - ✓ برای مدیریت خود کار فایل‌ها استفاده کنید with از ✓
  - ✓ استفاده کنید "b" برای کار با فایل‌های باینری از ✓
- 

### متد seek :

متد seek() در پایتون برای تغییر موقعیت مکان‌نمای (cursor) فایل استفاده می‌شود.

### متد seek(offset, whence)

file.seek(offset, whence)

آرگومان	توضیح
offset	تعداد بایت‌هایی که مکان‌نما باید جابجا شود.
whence	مشخص‌کننده‌ی نقطه‌ی شروع جابجایی (0 برای ابتدا، 1 از موقعیت فعلی، 2 از انتها)

### مثال: تغییر مکان‌نما به موقعیت خاص

```
file = open("example.txt", "r")
file.seek(5) # حرکت مکان‌نما به بایت پنجم
print(file.read()) # خواندن از موقعیت جدید
file.close()
```

### مثال: خواندن از انتهای فایل با seek()

```
file = open("example.txt", "rb")
file.seek(-10, 2) # حرکت ۱۰ بایت به عقب از انتهای فایل
print(file.read()) # خواندن آخرین ۱۰ بایت
file.close()
```

---

## متد tell() :

پرای مشاهده موقعیت فعلی مکان نما

```
file = open("example.txt", "r")
```

```
print(file.tell()) # نمایش موقعیت مکان نما
```

```
file.read(5)
```

```
print(file.tell()) # موقعیت جدید مکان نما بعد از خواندن ۵ بایت
```

```
file.close()
```