

مدیریت فایل‌ها در پایتون

مدیریت فایل‌ها یکی از مهم‌ترین مهارت‌های برنامه‌نویسی است، به‌ویژه در مواقعی که نیاز به ذخیره‌سازی یا پردازش داده‌های متنی و باینری داریم.

در این بخش، روش‌های مختلف خواندن، نوشتن، و مدیریت فایل‌ها در پایتون را بررسی می‌کنیم.

۱. باز کردن فایل با `open()`

تابع `open()` برای باز کردن فایل‌ها استفاده می‌شود و دو آرگومان اصلی دارد:

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

📌 پارامترهای مهم:

- `"filename.txt"` → نام و مسیر فایل
- `"mode"` → حالت باز کردن فایل

۲. حالت‌های مختلف باز کردن فایل

حالت	توضیح
<code>r</code>	خواندن (Read) → فایل باید از قبل وجود داشته باشد.
<code>w</code>	نوشتن (Write) → اگر فایل وجود داشته باشد، محتوای قبلی پاک می‌شود.
<code>a</code>	اضافه کردن (Append) → داده‌ها به انتهای فایل اضافه می‌شوند.
<code>x</code>	ایجاد (Exclusive Create) → فقط اگر فایل وجود نداشته باشد، آن را ایجاد می‌کند.
<code>rb</code>	خواندن باینری
<code>wb</code>	نوشتن باینری
<code>ab</code>	اضافه کردن باینری

۳. خواندن فایل (Mode `r`)

```
file = open("example.txt", "r") # باز کردن فایل در حالت خواندن
content = file.read() # خواندن کل محتوا
print(content)
file.close() # بستن فایل
```

📌 نکات مهم:

- اگر فایل وجود نداشته باشد، خطای `FileNotFoundError` دریافت می‌کنید.
- پس از خواندن، باید فایل را ببندید (`close()`).

✓ خواندن فایل خط به خط با `() readline` و `() readlines`

```
file = open("example.txt", "r")
print(file.readline()) # فقط یک خط می‌خواند
print(file.readlines()) # همه خطوط را به صورت لیست باز می‌گرداند
file.close()
```

۴. نوشتن در فایل (w Mode)

✎ اگر فایل وجود داشته باشد، محتوای قبلی پاک می‌شود!

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

۵. اضافه کردن به فایل (a Mode)

✎ محتوای جدید به انتهای فایل اضافه می‌شود.

```
file = open("example.txt", "a")
file.write("This line is appended.\n")
file.close()
```

۶. استفاده از `with` برای مدیریت خودکار فایل‌ها

✎ بهتر است از `with open` استفاده کنید تا نیازی به `close` نداشته باشید.

```
with open("example.txt", "r") as file:
    content = file.read()
    print(content) # فایل به‌طور خودکار بسته می‌شود
```

✓ مثال: نوشتن در فایل با `with open`

```
with open("example.txt", "w") as file:
    file.write("Hello, Python!\n")
```

✓ مثال: اضافه کردن داده‌ها با `with open`

```
with open("example.txt", "a") as file:
    file.write("Appending new content.\n")
```

۷. خواندن و نوشتن فایل‌های باینری (rb و wb)

✎ برای فایل‌های تصویری، صوتی، و سایر فایل‌های غیرمتنی استفاده می‌شود.

خواندن یک فایل باینری

```
with open("image.jpg", "rb") as file:
    binary_data = file.read()
    print(binary_data[:10]) # نمایش 10 بایت اول
```

نوشتن در یک فایل باینری

```
with open("new_image.jpg", "wb") as file:
    file.write(binary_data)
```

۸. بررسی وجود فایل قبل از باز کردن آن

```
import os

if os.path.exists("example.txt"):
    with open("example.txt", "r") as file:
        print(file.read())
else:
    print("فایل موردنظر وجود ندارد")
```

۹. مدیریت خطاهای فایل با try-except

```
try:
    with open("non_existing_file.txt", "r") as file:
        content = file.read()
except FileNotFoundError:
    print("خطا: فایل وجود ندارد")
```

۱۰. حذف فایل

```
import os

if os.path.exists("example.txt"):
    os.remove("example.txt")
    print("فایل حذف شد.")
else:
    print("فایل موردنظر پیدا نشد")
```

جمع‌بندی

- ✓ `open("file", "mode")` برای باز کردن فایل استفاده می‌شود.
 - ✓ حالت‌های مختلف شامل `r` (خواندن)، `w` (نوشتن)، `a` (اضافه کردن) و `b` (باینری) است.
 - ✓ `with open` استفاده شود تا نیازی به `close` نباشد.
 - ✓ برای مدیریت خطاها از `try-except` استفاده کنید.
 - ✓ می‌توان فایل‌های باینری مانند عکس و ویدیو را نیز خواند و نوشت.
- 🚀 مدیریت فایل‌ها در پایتون به شما کمک می‌کند تا داده‌های خود را به‌صورت دائمی ذخیره کنید! 🔥