

## کار با فایل‌های متنی در پایتون: قدم اول

فرض کنید یک فایل متنی به نام `data.txt` داریم که حاوی اطلاعات زیر است:

```
1 | Name: Alice
2 | Age: 30
3 | City: New York
```

[Copy](#)[Python](#)

### اشاره‌گر فایل: نشانگر موقعیت در فایل

اشاره‌گر فایل (File Pointer) در پایتون، مانند یک نشانگر در یک کتاب، موقعیت فعلی خواندن یا نوشتن در فایل را مشخص می‌کند. وقتی فایلی را باز می‌کنیم، این نشانگر به طور پیش‌فرض در ابتدای فایل قرار دارد.

- تابع `tell()`: این تابع موقعیت فعلی اشاره‌گر فایل را برمی‌گرداند.

```
1 | f = open("data.txt", 'r')
2 | print(f.tell()) # Output: 0 # Prints the current file pointer position
```

- تابع `seek()`: این تابع موقعیت اشاره‌گر فایل را به مکان مشخص شده منتقل می‌کند.

```
1 | f.seek(10) # Moves the file pointer to position 10
```

### خواندن کاراکترها از فایل

- تابع `read()`: این تابع تعداد مشخصی کاراکتر را از فایل می‌خواند و برمی‌گرداند.

```
1 | f = open("data.txt", 'r')
2 | print(f.read(4)) # Output: 'Name' # Reads and prints the first 4 charact
3 | print(f.tell()) # Output: 4 # Prints the updated file pointer position
4 | print(f.read(6)) # Output: ': Alic' # Reads and prints the next 6 charact
```

اگر آرگومان `read()` را خالی بگذاریم، کل محتوای فایل خوانده می‌شود.

مثال: برنامه‌ای که دو عدد `start` و `end` را از ورودی می‌گیرد و کاراکترهای بین این دو موقعیت را از فایل `data.txt` چاپ می‌کند.

```
1 start, end = map(int, input().split())
2 f = open("data.txt", 'r')
3 f.seek(start) # Moves the file pointer to the 'start' position
4 content = f.read(end - start + 1) # Reads characters from start to end
5 print(content)
```

## خواندن خطوط فایل

• تابع `readline()`: این تابع یک خط از فایل را می‌خواند و برمی‌گرداند.

```
1 f = open("data.txt", 'r')
2 print(f.readline()) # Output: 'Name: Alice\n' # Reads and prints the first line
3 print(f.readline()) # Output: 'Age: 30\n' # Reads and prints the second line
```

مثال: برنامه‌ای که خطوط فایل `data.txt` را به ترتیب معکوس چاپ می‌کند.

```
1 f = open("data.txt", 'r')
2 lines = []
3 while True:
4     line = f.readline() # Reads a line from the file
5     if not line: # Checks if the end of the file has been reached
6         break
7     lines.append(line) # Add the line to the list
8 lines.reverse() # Reverse the list of lines
9 for line in lines:
10     print(line, end='') # Print the lines in reverse order
```

نکات کلیدی:

- همیشه پس از اتمام کار با فایل، آن را با استفاده از `close()` ببندید .
- در صورت عدم وجود فایل، خطای `FileNotFoundError` رخ می‌دهد.
- برای خواندن تمام خطوط فایل به صورت یک لیست میتوان از تابع `readlines()` استفاده کرد.