

## کار با فایل‌های باینری در پایتون

در درس‌های قبلی، با نحوه کار با فایل‌های متنی آشنا شدیم. اما فایل‌های دیگری نیز وجود دارند که محتوای آن‌ها به صورت متن قابل خواندن نیست. این فایل‌ها، فایل‌های باینری نامیده می‌شوند و شامل تصاویر، فیلم‌ها، فایل‌های اجرایی و ... هستند.

### دودویی (Binary)

همانطور که می‌دانید، داده‌ها در کامپیوتر به صورت بیت‌ها (0 و 1) ذخیره می‌شوند. هر 8 بیت، یک بایت را تشکیل می‌دهد.

### مبنای اعداد

در سیستم اعداد باینری (مبنای 2)، هر رقم با توانی از 2 ارزش‌گذاری می‌شود. برای نمایش راحت‌تر بایت‌ها، از سیستم اعداد هگزادسیمال (مبنای 16) استفاده می‌کنیم.

### باینری در پایتون

پایتون نوع داده bytes را برای کار با داده‌های باینری ارائه می‌دهد.

```
1 # Create a bytes object
2 data = b'\x48\x65\x6c\x6c\x66' # Represents "Hello" in hexadecimal
3
4 # Accessing bytes
5 print(data[0]) # Output: 72 (ASCII value of 'H')
6
7 # Converting bytes to a list of integers
8 print(list(data)) # Output: [72, 101, 108, 108, 111]
9
10 # Creating bytes from ASCII characters
11 data2 = b'Hello'
12 print(list(data2)) # Output: [72, 101, 108, 108, 111]
13
14 # Creating bytes from a list of integers
15
```

```
--  
16 | data3 = bytes([72, 101, 108, 108, 111])  
    | print(data3) # Output: b'Hello'
```

## فایل‌های غیرمتنی در پایتون

برای کار با فایل‌های باینری، از حالت 'rb' (read binary) و 'wb' (write binary) در تابع `open()` استفاده می‌کنیم.

مثال: کپی کردن یک فایل باینری

```
1 | with open('input.bin', 'rb') as input_file:  
2 |     data = input_file.read()  
3 |     with open('output.bin', 'wb') as output_file:  
4 |         output_file.write(data)
```

نکات کلیدی:

- از `readline()` برای فایل‌های باینری استفاده نکنید.
- داده‌های باینری را به صورت `bytes` بخوانید و بنویسید.
- برای مدیریت فایل‌ها، از دستور `with` استفاده کنید.