

## WAV Audio File Header Parser

Given a WAV audio file, parse the file header and extract important audio properties. WAV files contain a header with metadata about the audio data stored in the file.

Your task is to read the binary header information and return a dictionary containing the extracted properties.

### Input Format

A `BufferedReader` object `f` representing an open WAV file in binary read mode.

### Output Format

A Python dictionary with the following keys:

- "fmt": Audio format code (integer)
- "file\_size": Total file size in bytes (integer)
- "channels": Number of audio channels (integer)
- "sample\_rate": Sample rate in Hz (integer)
- "bit\_rate": Bit rate in bits per second (integer)
- "block\_align": Block alignment in bytes (integer)
- "bit\_per\_sample": Bits per sample (integer)
- "data\_size": Size of audio data in bytes (integer)
- "duration": Duration of audio in seconds (float)

### Constraints

- The input file is a valid WAV file with standard RIFF header
- Audio format is PCM (uncompressed)

## Sample Input

```
example.wav
```

## Sample Output

```
=====
Audio Format: PCM
File size: 5289194
Channels: 2
Sample Rate: 44100
Bit rate: 1411200
Block align: 4
Bit per sample: 16
Data Size: 5289012
Duration(s): 29.983061224489795
=====
```

## Implementation

**Goal:** Fill in the following function:

```
from io import BufferedReader

def extract_wav_header(f: BufferedReader):
    prop = {}
    ...
    return prop
if __name__ == "__main__":
    filename = "example.wav"
    with open(filename, "rb") as f:
        audio_prop = extract_wav_header(f)
    print("=" * 50)
    print("Audio Format:", "PCM" if audio_prop["fmt"] == 1 else "Other")
    print("File size:", audio_prop["file_size"])
    print("Channels:", audio_prop["channels"])
    print("Sample Rate:", audio_prop["sample_rate"])
    print("Bit rate:", audio_prop["bit_rate"])
    print("Block align:", audio_prop["block_align"])
    print("Bit per sample:", audio_prop["bit_per_sample"])
    print("Data Size:", audio_prop["data_size"])
    print("Duration(s):", audio_prop["duration"])
    print("=" * 50)
```

## Reference

For details on the WAV file format and header structure, see:

<https://ccrma.stanford.edu/courses/422-winter-2014/projects/WaveFormat/>

## Hint

- Use `f.read(n)` to read `n` bytes from the file
- Use `int.from_bytes` to convert bytes to integers
- Be aware of the byte order (little-endian or big-endian)
- The WAV header follows the RIFF format specification

## Warnings

- You are **NOT ALLOWED** to use libraries like `wave`, `numpy`, or `struct` —  
you **MUST** parse the binary header manually