

디스크파일

2015년 10월 23일에 한국어로 옮겨짐

원문: <https://github.com/torch/torch7/blob/master/doc/diskfile.md>

목차

부모 클래스: [파일](#)

디스크파일은 디스크에 저장된 파일에 대한 기본적인 읽기/쓰기 연산들을 수행할 수 있는 한 특정한 파일입니다. 디스크파일은 [파일](#)에서 설명된 모든 메소드들과 [엔디언 인코딩](#)에 관련된 몇 가지 추가적인 메소드들을 구현합니다.

기본적으로, 디스크파일은 [아스키](#)(ASCII) 모드입니다. 만약 [바이너리](#) 모드로 바뀌면, 기본 엔디언 인코딩은 그 컴퓨터의 기본값을 따릅니다.

디스크파일은 `torch.DiskFile(fileName, mode)`로 입력되는 ("r", "w", 또는 "rw") 파라미터 mode에 따라 읽기, 쓰기, 또는 읽기-쓰기 모드에서 열릴 수도 있습니다.

`torch.DiskFile(fileName, [mode], [quiet])`

인자로 입력된 mode를 사용하여 디스크 상의 fileName을 여는 *생성자*. 유효한 mode에는 "r"(읽기), "w"(쓰기), 또는 "rw"(읽기-쓰기)가 있습니다. 기본값은 읽기 모드입니다.

만약 읽기-쓰기 모드인데 파일이 존재하지 않으면, 그 파일이 새로 *만들어질 것입니다*. 만약 읽기-쓰기 모드인데 파일이 이미 존재하면, 그 파일이 열리고 그 파일의 시작점에 위치하게 될 것입니다.

만약 quiet가 true이면, 그 파일을 여는 데 문제가 있더라도 에러는 생기지 않을 것입니다. 대신 nil이 리턴됩니다.

파일은 기본적으로 [아스키](#)(ASCII) 모드로 열립니다.

`bigEndianEncoding()`

[바이너리](#) 모드에서, 인코딩을 [빅 엔디언](#)으로 강제합니다. ([빅 엔디언](#): 메모리 주소가 증가함에 따라 한 숫자에서 큰 단위에 해당하는 부분이 먼저 기록되는 방식).

[boolean] isBigEndianCPU()

만약 그 컴퓨터의 [중앙처리장치](#)(CPU)가 [빅 엔디언](#)으로 동작하면, true를 리턴합니다. ([빅 엔디언](#): 메모리 주소가 증가함에 따라 한 숫자에서 큰 단위에 해당하는 부분이 먼저 기록되는 방식).

[boolean] isLittleEndianCPU()

만약 그 컴퓨터의 [중앙처리장치](#)(CPU)가 [리틀 엔디언](#)으로 동작하면, true를 리턴합니다. ([리틀 엔디언](#): 메모리 주소가 증가함에 따라 한 숫자에서 작은 단위에 해당하는 부분이 먼저 기록되는 방식).

littleEndianEncoding()

[바이너리](#) 모드에서, 인코딩을 [리틀 엔디언](#)으로 강제합니다. ([리틀 엔디언](#): 메모리 주소가 증가함에 따라 한 숫자에서 작은 단위에 해당하는 부분이 먼저 기록되는 방식).

nativeEndianEncoding()

[바이너리](#) 모드에서, 인코딩을 그 컴퓨터의 기본 엔디언(native endian)으로 강제합니다.

목차

[torch.DiskFile\(fileName, \[mode\], \[quiet\]\)](#)

[bigEndianEncoding\(\)](#)

[\[boolean\] isBigEndianCPU\(\)](#)

[\[boolean\] isLittleEndianCPU\(\)](#)

[littleEndianEncoding\(\)](#)

[nativeEndianEncoding\(\)](#)

[목차](#)

❖ 틀렸거나 보완할 점을 본문에 댓글로 또는 저에게 [이메일](#)로 알려 주시면 감사하겠습니다.