

Node.js 파일 시스템 다루기

Node.js에서는 파일 시스템을 손쉽게 다룰 수 있는 다양한 API를 제공합니다. 이를 활용하여 텍스트 파일에서 데이터를 읽고 쓰며, 파일과 폴더를 관리할 수 있습니다.



작성자: 현욱





파일 읽기: `fs.readFile()`

1

파일 경로

`fs.readFile()` 함수는 파일 경로를 매개변수로 받습니다.

2

비동기 처리

파일 읽기는 비동기 작업으로 처리되며, 콜백 함수로 결과를 받습니다.

3

인코딩 설정

콜백 함수에서 인코딩 방식을 지정할 수 있습니다.

파일 쓰기: `fs.writeFile()`

파일 경로

`fs.writeFile()` 함수는 파일 경로와 작성할 데이터를 매개변수로 받습니다.

옵션 설정

인코딩 방식, 모드 등 다양한 옵션을 설정할 수 있습니다.

비동기 처리

파일 쓰기 역시 비동기 작업으로 처리되며, 콜백 함수로 결과를 받습니다.

파일 삭제: `fs.unlink()`

파일 경로

`fs.unlink()` 함수는 삭제할 파일의 경로를 매개변수로 받습니다.

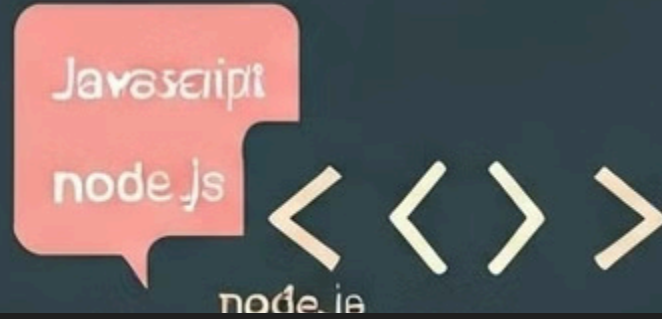
비동기 처리

파일 삭제 역시 비동기 작업으로 처리되며, 콜백 함수로 결과를 받습니다.

오류 처리

콜백 함수에서 오류 처리를 해주어야 합니다.





디렉토리 생성/삭제

디렉토리 생성: **fs.mkdir()**

새로운 디렉토리를 생성할 수 있습니다.

1

2

디렉토리 삭제: **fs.rmdir()**

기존의 디렉토리를 삭제할 수 있습니다.

비동기 처리

디렉토리 생성과 삭제 또한 비동기 작업으로 처리됩니다.

3

파일 정보 확인: `fs.stat()`



크기

파일의 크기를 확인할 수 있습니다.



수정 시간

파일이 마지막으로 수정된 시간을 확인할 수 있습니다.



파일 유형

파일이 일반 파일인지 디렉토리인지 확인할 수 있습니다.



텍스트 파일 **CRUD** 실습

1

파일 생성

`fs.writeFile()`로 새 파일을 생성합니다.

2

파일 읽기

`fs.readFile()`로 파일 내용을 읽어옵니다.

3

파일 수정

`fs.writeFile()`로 파일 내용을 업데이트합니다.

4

파일 삭제

`fs.unlink()`로 파일을 삭제합니다.





요약 및 질의응답

1 핵심 기능

파일 읽기, 쓰기, 삭제, 디렉토리 관리 등 다양한 파일 시스템 기능을 살펴보았습니다.

2 비동기 처리

모든 파일 시스템 작업은 비동기 처리되므로 콜백 함수로 결과를 확인해야 합니다.

3 실습 예시

텍스트 파일을 대상으로 CRUD 작업을 직접 구현해보았습니다.