KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

# 파일 관리하기 - path, File System 모듈

[KB] IT's Your Life

KB 국민은행

# 파일 관리하기 – path, File System 모듈

- path 내장모듈을 이용하여 경로 요소 'some', 'work', 'ex.txt'를 결합하세요.
  - path.js

```
출력]
some\work\ex.txt
```

- 위의 파일에서 현재 파일의 전체 절대경로와 디렉토리 경로를 출력하도록 확장하세요.

```
출력]
some\work\ex.txt
파일 절대 경로: c:\workspace\node\src\03\path.js
경로만: c:\workspace\node\src\03
```

- 위의 파일에서 현재 파일의 파일명(확장명 포함)과 확장명을 제외한 파일명을 출력하도록 확장하세요.

```
출력]
…
파일 이름: path.js
파일 이름(확장자 제외): path
```

# 파일 관리하기 – path, File System 모듈

## path.js

```
const path = require('path');

// 경로 연결하기
const fullPath = path.join('some', 'work', 'ex.txt');
console.log(fullPath);
```

```
some\work\ex.txt
```

# 파일 관리하기 – path, File System 모듈

## path.js

```
const path = require('path');

// 경로 연결하기
const fullPath = path.join('some', 'work', 'ex.txt');
console.log(fullPath);

// 절대 경로
console.log(`파일 절대 경로: ${__filename}`);

// 경로 추출하기
const dir = path.dirname(__filename);
console.log(`경로만: ${dir}`);
```

```
some\work\ex.txt
파일 절대 경로: c:\workspace\node\src\03\path.js
경로만: c:\workspace\node\src\03
```

# 파일 관리하기 – path, File System 모듈

## path.js

```
const path = require('path');

…

// 파일 이름 추출하기
const fn = path.basename(__filename);
const fn2 = path.basename(__filename, '.js');

console.log(`파일 이름: ${fn}`);
console.log(`파일 이름(확장자 제외): ${fn2}`);
```

```
…
파일 이름: path.js
파일 이름(확장자 제외): path
```

# 파일 관리하기 - path, File System 모듈

- **위의 파일에서 현재 파일의 확장명을 출력하도록 확장하세요.**

```
출력]
…
파일 확장자: .js
```

- **위의 파일에서 현재 파일의 경로 요소 전체를 출력하도록 확장하세요.**

```
출력]
…
{
  root: 'c:\\',
  dir: ' c:\\workspace\\node\\src\\03',
  base: 'path.js',
  ext: '.js',
  name: 'path'
}
```

# 파일 관리하기 – path, File System 모듈

## path.js

```
const path = require('path');

…


// 파일 확장자 추출
const ext = path.extname(__filename);
console.log(`파일 확장자: ${ext}`);
```

```
…
파일 확장자: .js
```

# 파일 관리하기 – path, File System 모듈

## path.js

```
const path = require('path');

…

// 경로 분해하기
const parsedPath = path.parse(__filename);
console.log(parsedPath);
```

```
…
{
  root: 'c:\\',
  dir: ' c:\\workspace\\node\\src\\03',
  base: 'path.js',
  ext: '.js',
  name: 'path'
}
```

# 파일 관리하기 - path, File System 모듈

- **현재 작업 디렉토리의 파일 목록을 출력하세요.**
  - 동기함수 호출로 처리하세요.
  - list-1.js

    출력]

    ```
    [ 'package.json', 'node_modules', 'package-lock.json', '01', '03' ]
    ```

  - 비동기함수 호출로 처리하세요.
  - list-2.js

    출력]

    ```
    [ 'package.json', 'node_modules', 'package-lock.json', '01', '03' ]
    ```

# 파일 관리하기 - path, File System 모듈

## list-1.js

```
const fs = require('fs');

let files = fs.readdirSync('./');
console.log(files);
```

```
[ 'package.json', 'node_modules', 'package-lock.json', '01', '03' ]
```

# 파일 관리하기 - path, File System 모듈

## list-2.js

```
const fs = require('fs');

fs.readdir('./', (err, files) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log(files);
});
```

```
[ 'package.json', 'node_modules', 'package-lock.json', '01', '03' ]
```

# 파일 관리하기 - path, File System 모듈

## 다음 조건을 만족하는 프로그램을 작성하세요.

- example.txt

```
Node.js is an open-source, cross-platform JavaScript runtime environment.
Node.js는 Chrome v8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.
```

- example.txt의 내용을 읽어 화면에 출력하세요(동기 함수로 처리)
- read-1.js

  출력]

  Node.js is an open-source, cross-platform JavaScript runtime environment.
  Node.js는 Chrome v8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

- example.txt의 내용을 읽어 화면에 출력하세요(비동기 함수로 처리)
- read-2.js

  출력]

  Node.js is an open-source, cross-platform JavaScript runtime environment.
  Node.js는 Chrome v8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

# 파일 관리하기 - path, File System 모듈

## example.txt

```
Node.js is an open-source, cross-platform JavaScript runtime environment.
Node.js는 Chrome v8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.
```

## read-1.js

```
fs = require('fs');

const data = fs.readFileSync('./example.txt');
console.log(data);
```

```
<Buffer 4e 6f 64 65 2e 6a 73 20 69 73 20 61 6e 20 6f 70 65 6e 2d 73 6f 75 72 63 65 2c 20 63 72
6f 73 73 2d 70 6c 61 74 66 6f 72 6d 20 4a 61 76 61 53 63 72 69 ... 110 more bytes>
```

# 파일 관리하기 - path, File System 모듈

## read-2.js

```
fs = require('fs');

fs.readFile('./example.txt', 'utf-8', (err, data) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log(data);
});
```

```
Node.js is an open-source, cross-platform JavaScript runtime environment.
Node.js는 Chrome v8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.
```

# 파일 관리하기 - path, File System 모듈

- 앞의 example.txt를 읽어서 text-1.txt로 저장하세요.
  - 동기 함수로 작성하세요.
  - write-1.js
  - 비동기 함수로 작성하세요.
  - write-2.js
- 앞의 example.txt를 읽어서 text-1.txt로 저장하는데, 기존에 text-1.txt가 존재하면, 파일이 존재한다고 출력하고, 존재하지 않는 경우에만 작성하세요.(동기 함수 사용)
  - write-3.js

## write-1.js

```
fs = require('fs');

const data = fs.readFileSync('./example.txt', 'utf8');
fs.writeFileSync('./text-1.txt', data);
```

## write-2.js

```
fs = require('fs');

fs.readFile('./example.txt', 'utf8', (err, data) => {
  if (err) {
    console.log(err);
  }

  fs.writeFile('./text-2.txt', data, (err) => {
    if (err) {
      console.log(err);
    }
    console.log('text-2.txt is saved!');
  });
});
```

## write-3.js

```
fs = require('fs');

const data = fs.readFileSync('./example.txt', 'utf8');

if(fs.existsSync('text-1.txt')) {     // text-1.txt 파일이 있다면
  console.log('file already exist');
} else {                              // text-1.txt 파일이 없다면
  fs.writeFileSync('./text-1.txt', data);
}
```

# 파일 관리하기 - path, File System 모듈

- 현재 디렉토리에 test라는 디렉토리가 있는지 검사하고,
  존재하지 않는 경우 디렉토리를 만드세요.
  - dir-1.js

- 현재 디렉토리에 ./test2/test3/test4라는 디렉토리가 있는지 검사하고,
  존재하지 않는 경우 디렉토리를 만드세요.
  - dir-2.js

## dir-1.js

```
fs = require('fs');

if (fs.existsSync('./test')) { // 디렉터리가 있다면
  console.log('folder already exists');
} else {                       // 디렉터리가 없다면
  fs.mkdir('./test', (err) => {
    if (err) {
      return console.error(err);
    }
    console.log('folder created');
  });
}
```

## dir-2.js

```
fs = require('fs');

if (fs.existsSync('./test2/test3/test4')) {  // 디렉터리가 있다면
  console.log('folder already exists');
} else {                                     // 디렉터리가 없다면
  fs.mkdir('./test2/test3/test4', { recursive: true }, (err) => {
    if (err) {
      return console.error(err);
    }
    console.log('folder created');
  });
}
```