

Useful fs Module Functions

Note: You will often see the following “promisified” for use with async/await. The promisified versions will return a Promise that resolves callback’s contents or rejects if there was an error. Remember that you should always handle potential fs function errors (try/catch if async/await, if/else if standard callback).

Example of promisifying callback-version of fs.readFile:

```
fs.readFile("file.txt", "utf8", (err, contents) => {  
  ...  
});
```

// Promisified:

```
const util = require("util");  
const readFile = util.promisify(fs.readFile);
```

```
async function example() {  
  try {  
    let contents = await readFile("file.txt");  
    ...  
  } catch(err) {  
    ...  
  }  
}
```

Function	Description
fs.readFile(filename, "utf8", callback); fs.readFile("file.txt", "utf8", (err, contents) => { ... });	Reads the contents of the file located at relative directory filename . If successful, passes the file contents to the callback as contents parameter. Otherwise, passes error info to callback as error parameter.
fs.writeFile(filename, data, "utf8", callback); fs.writeFile("file.txt", "new contents", "utf8", (err) => { ... });	Writes data string to the file specified by filename , overwriting its contents if it already exists. If an error occurs, error is passed to the callback function.
fs.appendFile(filename, data, "utf8", callback); fs.appendFile("file.txt", "added contents", "utf8", (err) => { ... });	Writes data to the file specified by filename , appending to its contents. Creates a new file if the filename does not exist. If an error occurs, error is passed to the callback function.
fs.existsSync(filename);	Returns true if the given filename exists. <u>This is the only synchronous fs function you may use in CSE154</u> (the asynchronous function is deprecated due to race conditions).
fs.readdir(path, callback); fs.readdir("dir/path", (err, contents) => { ... });	Retrieves all files within a directory located at path . If successful, passes the array of directory content paths (as strings) to the callback as contents parameter. Otherwise, passes error info to callback as error parameter.