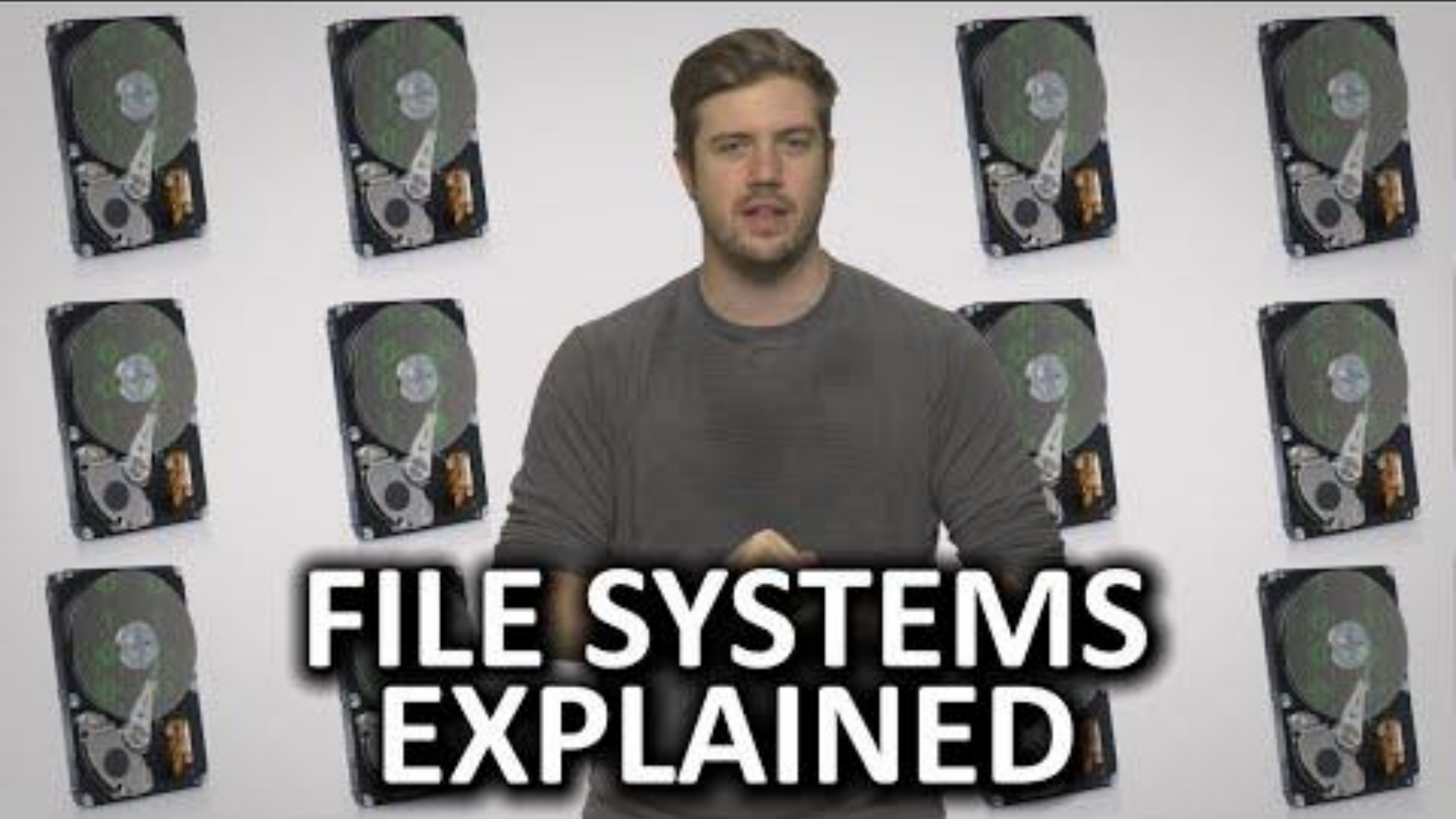
Introduction to File System Module

sebinbenjamin

What we’ll do learn today

- What is a File System ?
- The node File System Module
    - Synchronous methods
    - Asynchronous methods
- Code

What is a File System ?


FILE SYSTEMS
EXPLAINED

File System Module

- The File System is the **way** in which files are named and where they are placed logically for storage and retrieval.
  - Different ways of organizing and storing files on a hard drive, flash drive, or any other storage device
- The fs module provides an API for ***interacting with the file system*** in a manner closely modeled around standard (manufacturer-neutral) POSIX functions .
  - To use the module,

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

- File system operations have synchronous and asynchronous forms.

https://nodejs.org/dist/latest-v10.x/docs/api/fs.html#fs_file_system

Synchronous form

- Synchronous methods are are “blocking” and halts the program until all the resources are available.
- Sync versions of the methods that have proper `-Sync` suffix
    - readFileSync, accessSyn, copyFileSync etc
- **Exceptions** that occur using synchronous operations are thrown immediately and may be handled using **try/catch,** or may be allowed to bubble up.
- Programmers (YOU) are ***strongly encouraged to use the asynchronous versions*** of these calls. The synchronous versions will block the entire process until they complete — halting all connections.

Synchronous Form

```
const file_contents = fs.readFileSync('./pathParams.js', 'utf-8');

// do something with file_contents after read is successful
```

Errors need to be handled using a try/catch block

```
try {
    const data = fs.readFileSync('./some-file.txt', 'utf8');
    res.setHeader('Content-Type', 'text/javascript');
    res.statusCode = 200;
    res.end(data);
} catch (err) {
    console.log('Unable to read file.', err.message);
    res.statusCode = 500;
    res.end(err.message);
}
```

Asynchronous form

- File System module methods take a completion **callback** as **last argument**.
- **First argument/parameter** (err) to this callback function is always reserved for an **exception**.
    - If the operation was completed successfully, then the first argument will be null or undefined.
- There is no guaranteed ordering when using asynchronous methods.

```
fs.readFile('./pathParams.js', 'utf8', (err, file_contents) => {
  // do something with 'file_contents' after read is successful
});
```

Thank you