1. node - כניסה (invokes the Node.js runtime environment. This environment allows you to run JavaScript code outside of a web browser.)
2. טאב פעם אחת או פעמיים מראה לך את כל המשתנים הגלובליים והמתודות שיש לך (node modules)
3. node [path and file name] לדוגמא 🡨 node index.js הרצת הקוד
4. תרגול ראשוני

// fs.readFileSync() 🡪 In general, it is recommended to use the asynchronous fs.readFile() method instead of fs.readFileSync() unless you specifically need to block the execution of the rest of the code.

|  |  |
| --- | --- |
| Pros | Cons |
| Simple to use | Blocks execution of the rest of the code |
| Returns the file contents as a string | Can be a performance bottleneck for large files |

'utf-8', which specifies the character encoding of the file. Character encoding is a way of representing characters as a sequence of bytes. There are many different character encodings, each with its own strengths and weaknesses.

UTF-8 is a widely used character encoding that is capable of representing all of the characters in the Unicode standard. UTF-8 is a variable-length encoding, which means that it can use one to four bytes to encode a character. This makes UTF-8 a very efficient encoding for representing ASCII characters, as they can all be encoded using a single byte. However, UTF-8 can also represent non-ASCII characters, such as Chinese characters, which may require multiple bytes to encode.

const fs = require('fs') // **f**ile **s**ystem

const textIn = fs.readFileSync('./txt/input.txt', 'utf-8') //

console.log(textIn);

const textOut = `This is what we know about the avocado: ${textIn}.\nCreated on ${new Date().toLocaleString()}`;

fs.writeFileSync('./txt/output.txt',textOut);

console.log('File written!');

1. 10. Reading and Writing Files Asynchronously

// Non-blocking, asynchronous way

fs.readFile(`./txt/start.txt`, 'utf-8', (err, data1) => {

fs.readFile(`./txt/${data1}.txt`, 'utf-8', (err, data2) => {

fs.readFile(`./txt/append.txt`, 'utf-8', (err, data3) => {

console.log(data3);

fs.writeFile('./txt/final.txt',`${data2}\n${data3}`,'utf-8', err =>{

console.log('Your file has been written 😀');

} )

})

})

})

console.log('Will read file!');