

# Node.js Basics

## Get out of the Browser

Create the folder: [nodejs-basics](#), code the following functions:

Setup git, make sure to add a *.gitignore* file to your project

Code each of the following in its own file

To run the code, open the CMD in the folder where the file is located and write the following command:

`node file-name.js` with `file-name` being the name of the file you want to run.

### [about-time.js](#)

This function reads the content of a text file containing time-stamps, where each timestamp is in a separate line, and prints out the relative time:

```
ms(6000, { long: true } ) // 6 seconds
ms(12000, { long: true } ) // 12 seconds
ms(-23000, { long: true } ) //-23 seconds
```

#### Tips

- Use the npm package: [ms](#)

### [sum-from-file.js](#)

This function reads the content of a text file containing numbers, where each number is in a separate line, and returns a Promise to the sum of the numbers.

```
sumFromFile('data/nums.txt')
  .then(sum => console.log('Sum:', sum))
  .catch(err => console.log('Cannot sum:', err))
```

#### Tips

- The function returns a new Promise
- Use `fs.readFile` and `resolve/reject` the promise accordingly
- Use `Array.split` to separate the lines

### [draw-to-file.js](#)

```
drawSquareToFile()
```

```
function drawSquareToFile() {
  const str = getSquare(getRandomIntInclusive(3, 20))
  drawToFile(str)
    .then(() => {
      setTimeout(drawSquareToFile, 200)
    })
}

function getSquare(size) {
  var str = '*'.repeat(size) + '\n'
  for (let i = 0; i < size; i++) {
    str += '*' + ' '.repeat(size - 2) + '*\n'
  }
  str += '*'.repeat(size) + '\n'
  return str
}
```

#### Tips

- The function returns a new Promise
- Use fs.writeFile and resolve/reject the promise accordingly
- The function writes to a *data/pic.txt* file
- Open the output file in your vscode and watch it changes
- Experience using the VSCode debugger

### download-country-flags.js

Read the content of a JSON file, find the top 5 populated countries and download their flags

```
downloadCountryFlags()
function downloadCountryFlags() {
  const countries = getCountries()
  console.log('Countries:', countries.map(c => c.name))
  downloadFlags(countries)
    .then(() => {
      console.log('Your flags are ready')
    })
}

function getCountries() {
  var countries = []
  // TODO: get the countries from file (require),
  // sort by population (descending)
  // and return the top 5
  return countries
}

function downloadFlags(countries) {
  // TODO: use the download() function to download a flag
}
```

```
//      the name of the file should be the country name
// TODO: use the Array.map function to generate a promise for each download
// TODO: use Promise.all()
}
```

#### Tips

- Create a flags folder in your project and download to there
- Use the provided download function
- Use Array.map to map an array of countries to an array of promises