# 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

Reminder: To run the code, open the CMD in the folder where the file is located and write the following command: node file-name.js

## about-time.js

This function reads the content of a text file containing time-stamps, and prints out the relative time:

A group of text on a white background

Description automatically generated

##### Tips

* Use the npm package: [ms](https://www.npmjs.com/package/ms)
  + > npm init
  + > npm install ms
* File format: separate line for each time-stamp
* Tip: str.split('\r\n')

## sum-from-file.js

This function reads the content of a text file containing numbers, where each number is in a separate line. The function 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

## draw-to-file.js

Code the function writeToFile that writes a string to a file

|  |
| --- |
| drawSquareToFile()  function drawSquareToFile() {      const str = getSquare(getRandomIntInclusive(3, 20))      writeToFile(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 *pic.txt* 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 the JSON file with utilService.readJsonFile      // sort by population (descending)      // return the top 5      return countries  }  function downloadFlags(countries) {    const prms = countries.map(country => {      return utilService.download(        country.flags.svg,        `flags/${country.name.common}.svg`      )      })    return Promise.all(prms)  } |

##### Directions

* Create a *flags* folder in your project and run this code to download to that folder
* Modify the code to download the PNG version of the flag, which one is lighter?
* See how we map an array of countries to an array of promises?