

Node.js Basics

Get out of the Browser

Create the folder: **nodejs-basics**, code the following functions:

First create a node project with: `npm init`

and add the "type": "module" option to the - *package.json* file to allow ES6 imports.

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:

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

Tips

- Use the npm 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

countries-pdf.js

Use the [pdfkit-table](#) package to create a .pdf file with a table of countries, with three columns for the country name, capital and population.

Use the provided countries.json file as your data source.

Implementation steps

- install pdfkit-table: `npm i pdfkit-table`.
- Read the countries.json into an array using the `readJsonFile()` utility function.
- Use the following demo code as a reference for how to create a table in a .pdf file:

```
import fs from 'fs'
import PDFDocument from 'pdfkit-table'

// init document
let doc = new PDFDocument({ margin: 30, size: 'A4' })

// connect to a write stream
doc.pipe(fs.createWriteStream('./students.pdf'))

createPdf(doc)
  .then(() => doc.end())      // close document

function createPdf() {
  const table = {
    title: 'Students',
    subtitle: 'Sorted by age',
    headers: ['Name', 'Age', 'Grade'],
    rows: [
      ['Donni', '22', '10'],
      ['Ronni', '25', '9'],
      ['Boris', '27', '10'],
    ],
  }
  return doc.table(table, { columnsSize: [200, 100, 100] })
}
```

Directions

- Sort the array by country name before writing the table to the .pdf file.
- Check out the provided countries.pdf file for a reference to what your result should look like.

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?