**Node-harjoitus vastaukset 1-7**

**Harjoitus 1**

*// Tee ohjelma, joka tulostaa annetun tiedoston sisällön.*

*//tuodaan fs- ja chalk-kirjasto käyttöön*

const fs = require('fs')  
const chalk = require ('chalk')

*// Komentirivi parametriin päästää käsiksi process-argv*

const tiedosto = process.argv[2]

*try* {

    const dataBuffer = fs.readFileSync(tiedosto)

    const data = dataBuffer.toString()

    console.log(data)

} *catch* (e) {

    console.log(chalk.red('Tiedostoa ei löytynyt!'))

}

**1.B Lisäharjoitus (Kirjoittaminen)**

Tiedoston luominen ja sen sisälle kirjoittaminen

Vinkit!

1. Tuo fs-kirjasto
2. Luo tiedosto **uusitiedosto.txt**, jonka sisälle teksti **Moikka Maailma, function(err)**
3. Jos tallennus onnistui, niin tulee teksti Talletettu! 🡪

if (err) throw err;

const fs = require('fs');

*//luo tiedosto uusitiedosto.txt:*

fs.writeFile('uusitiedosto.txt', 'Moikka Muailima!', function (err) {

*if* (err) *throw* err;

  console.log('Talletettu!');

});

**Harjoitus 2**

*// Tee ohjelma, joka tulostaa annetun tiedoston sisällön.*

*//tuodaan fs- ja chalk-kirjasto käyttöön*

const fs = require('fs')

const chalk = require ('chalk')

*// Komentirivi parametriin päästää käsiksi process-argv*

const nimi = process.argv[2]

*try* {

    const dataBuffer = fs.readFileSync('nimet.txt')

    const data = dataBuffer.toString()

*if* (data.includes(nimi)){

        console.log(chalk.green('Nimi löytyi!'))

    } *else* {

        console.log(chalk.red('Nimeä ei löytynyt!'))

    }

} *catch* (e) {

    console.log(chalk.red('Tiedostoa ei löytynyt!'))

}

**Harjoitus 3**

*// ASENNA npm i validator*

*//tuodaan validator- ja chalk-kirjasto käyttöön*

const validator = require('validator')

const chalk =require('chalk')

*// Komentirivi parametriin päästää käsiksi process-argv*

const osoite = process.argv[2]

*if* (validator.isEmail(osoite)) {

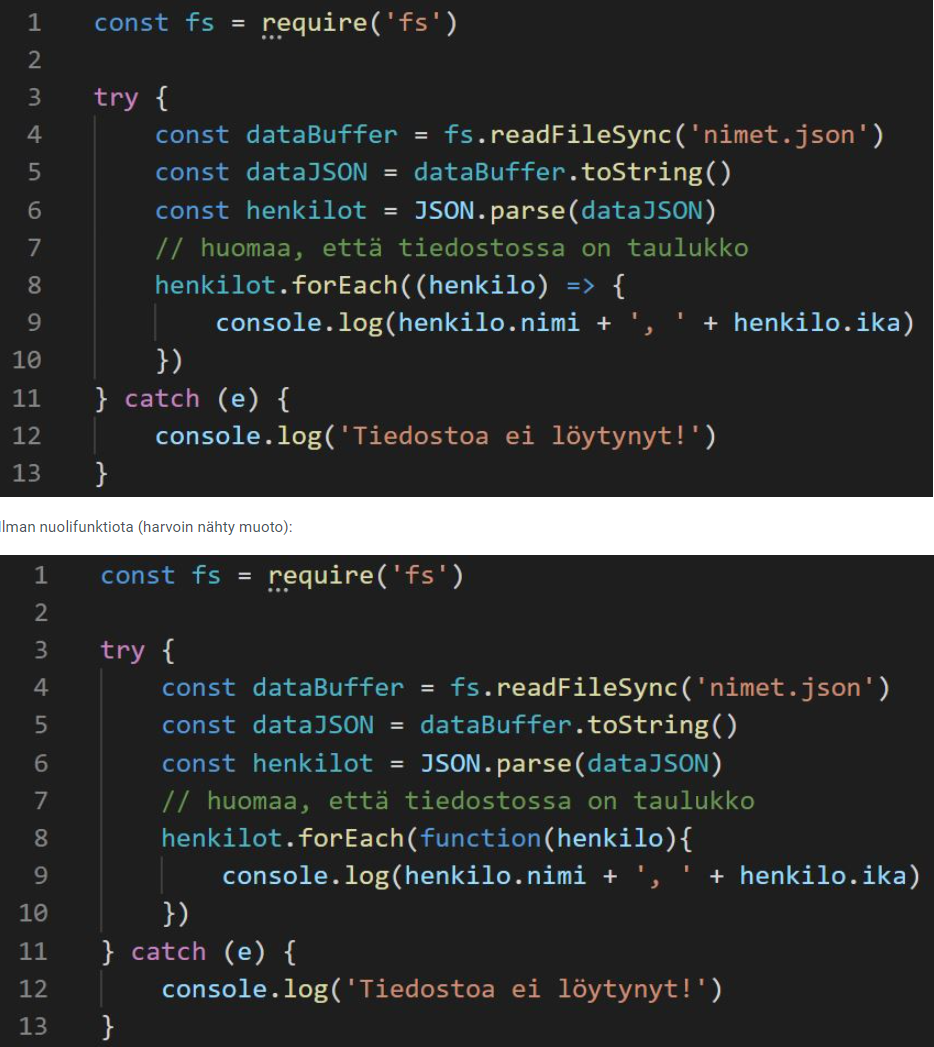
    console.log(chalk.green('Kelvollinen sähköpostiosoite'))

} *else* {

    console.log(chalk.red('Epäkelpo sähköpostiosoite'))

}

**Harjoitus 4**



const fs =require('fs')

*try* {

*// luetaan json-tiedosto*

    const dataBuffer = fs.readFileSync('nimet.json')

*// muutetaan buffer stringiksi*

    const dataJSON = dataBuffer.toString()

*// muutetaan json-muodosta JavaScriptiin objektiksi*

    const henkilot = JSON.parse(dataJSON)

*// huomaa, että tiedostossa on taulukko!!!*

    henkilot.forEach(function(henkilot) {

        console.log(henkilot.nimi + ', ' + henkilot.ika)

    })

} *catch* (e) {

    console.log('Tiedostoa ei löytynyt')

}

**Harjoitus 5**

const fs =require('fs')

*try* {

*// luetaan json-tiedosto*

    const dataBuffer = fs.readFileSync('henkilot.json')

*// muutetaan buffer stringiksi*

    const dataJSON = dataBuffer.toString()

*// muutetaan json-muodosta JavaScriptiin objektiksi*

    const henkilo = JSON.parse(dataJSON)

*// huomaa, että tiedostossa on taulukko!!!*

  henkilo.nimi = 'Kalle'

  henkilo.ika = 20

  const henkiloJSON = JSON.stringify(henkilo)

  fs.writeFileSync('henkilot.json', henkiloJSON)

} *catch* (e) {

    console.log('Tiedostoa ei löytynyt')

}

**Harjoitus 6**

const fs =require('fs')

const henkilo1 = {

    nimi: "Kalle",

    ika: 30,

    asuinpaikka: "Joensuu"

}

const henkilo2 = {

    nimi: "Matias",

    ika: 105,

    asuinpaikka: "Honolulu"

}

const henkilo3 = {

    nimi: "Maija",

    ika: 42,

    asuinpaikka: "Kuusamo"

}

const henkilot = [henkilo1, henkilo2, henkilo3]

*// muutetaan json-muodosta JavaScriptiin objektiksi*

const henkiloJSON = JSON.stringify(henkilot)

fs.writeFileSync('henkilot.json', henkiloJSON)

**Harjoitus 7**

const fs =require('fs')

*try* {

*// luetaan json-tiedosto*

    const dataBuffer = fs.readFileSync('opiskelijat.json')

*// muutetaan buffer stringiksi*

    const data = dataBuffer.toString()

*// muutetaan json-muodosta JavaScriptiin objektiksi*

    const opiskelijat = JSON.parse(data)

    const aikuiset = palautaAikuiset(opiskelijat)

*// tulostetaan täysi-ikäiset nimet*

    aikuiset.forEach((a) => console.log(a.nimi))

} *catch* (e) {

    console.log('Tiedostoa ei löytynyt')

}

*// funktio, joka palauttaa täysi-ikäiset*

 function palautaAikuiset (array) {

     console.log('TÄYSI-IKÄISET')

*return* array.filter((opiskelija) => opiskelija.ika >= 18)

 }