Help to complete the tasks of this exercise can be found on the chapter 1 ” Values, types and operators”, chapter 2 “Program structure” and chapter 4 “Data Structures: Objects and Arrays” of our course book “Eloquent JavaScript” (3rd edition) by Marijin Haverbeke. The aims of the exercise are to become familiar with JavaScript basics, especially values, types, arrays, and operators. The student will also learn what kinds of structures a JavaScript program has, and how the flow of the program is controlled.

Embed your theory answers, drawings, codes, and screenshots directly into this document. Always immediately after the relevant question. Return the document into your return box in itsLearning by the deadline.

It’s also recommendable to use Internet sources to supplement the information provided by the course book. Especially the * marked task can require that.

The maximum number of points you can earn from this exercise is 10 + 1 = 11.

Tasks:

1. Answer the questions? (4 * 0,25 = 1 point):

a. What is the result and why?

```
10 == '10'
```

On totta, koska löytyy samankaltainen arvo. JavaScript vertailee niitä vain arvon perusteelle, ei tietotyypijen mukaan.

b. What is the result and why?

```
12 === '12'
```

On epätosi, koska === vertaa myös tietotyyppejä, ja numero ei ole sama kuin merkkijono.

c. What is the result and why?

```
typeof "Kissa";
```

Tämä palauttaa string, joten tulos on merkijono

d. What is the value of the variable `currentPort` and why?

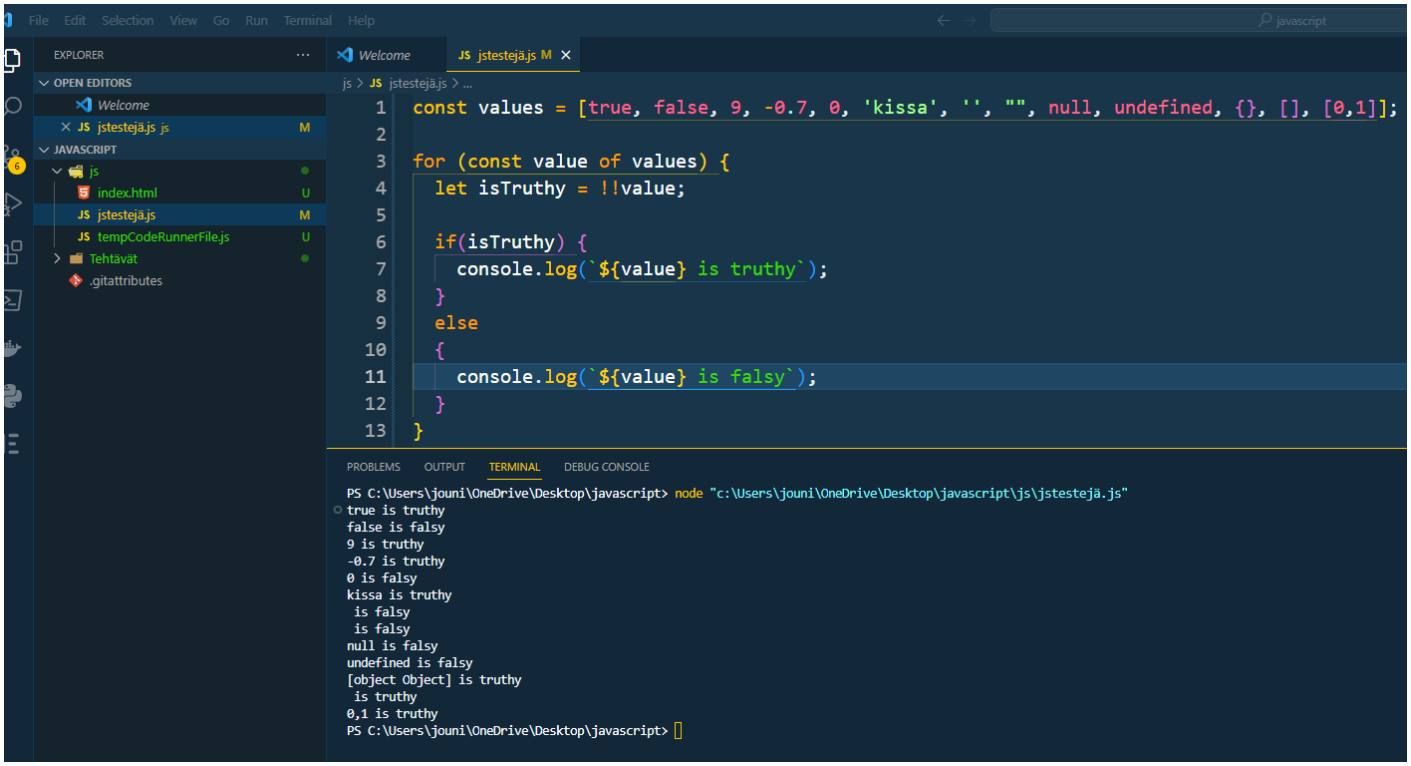
```
let port = 3001; let currentPort = port || 3000;
```

currentPort saa arvon 3001, koska portti on määritelty ja se on totta. Jos portti olisi epätosi tai määrittelemätön, currentPort olisi 3000.

2. JavaScript boolean values. (2 * 0,5 = 1 point)

a. Are the following values true or false in JavaScript? Use a browser console and program an if-else statement to find the answers.

`true, false, 9, -0.7, 0, 'kissa', '', "", null, undefined, {}, [], [0,1]`



b. Why?

Arvojen totuusarvoisuus tai epätotuusarvoisuus perustuu JavaScriptin tyyppimuuntelusääntöihin. If-else-lauseessa tai missä tahansa tilanteessa, jossa odotetaan boolean-arvoa, JavaScript muuntaa nämä arvot automaattisesti joko trueksi tai falseksi niiden luontaisista ominaisuuksista riippuen. Epätotuusarvoiset arvot ovat niitä, jotka katsotaan "tyhjiksi" tai "olematon", kuten false, 0, '', null, undefined sekä tyhjät objektit ja taulukot. Kaikki muut arvot katsotaan totuusarvoisiksi. Tämä käyttäytyminen on hyödyllinen ehtojen ja ohjausrakenteiden käytössä JavaScriptissä. Voit esimerkiksi käyttää sitä tarkistaaksesi, onko muuttujalla arvo ennen tiettyjen toimintojen suorittamista. Tässä on

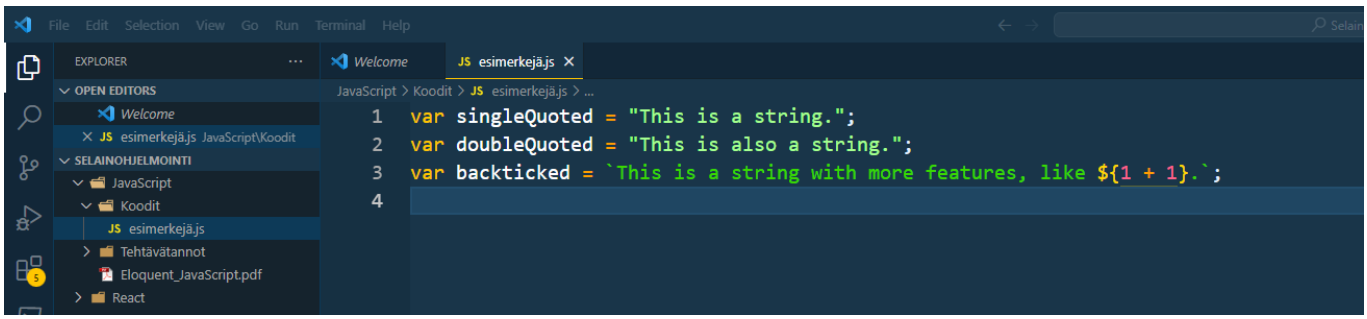
3. Strings. (4 * 0,5 = 2 points)

In JavaScript, you can use single quotes, double quotes, or backticks to mark strings.

a. Are there any differences between these differently marked strings?

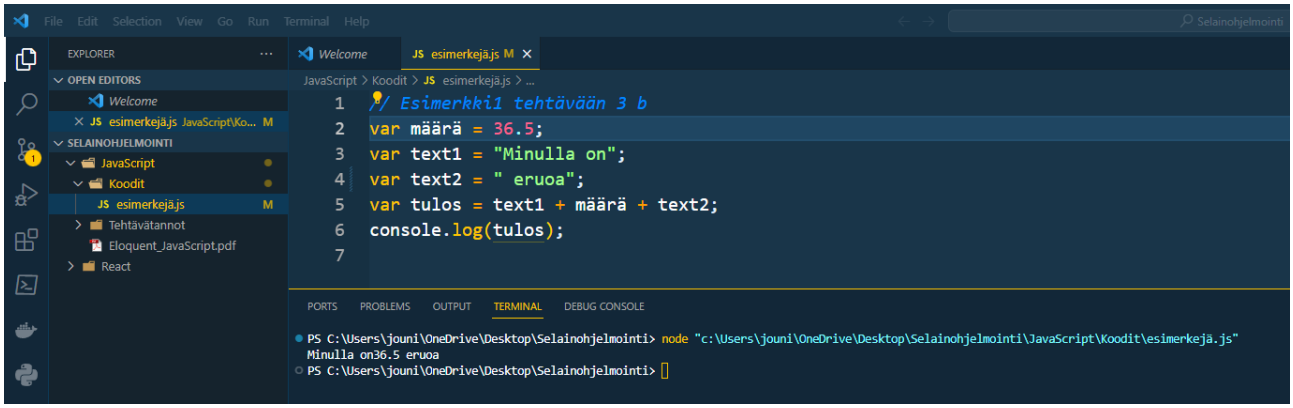
Yksinkertainen lainausmerkki (') ja kaksinkertainen lainausmerkki (") ovat perinteiset merkit merkkijonojen ympärillä JavaScriptissä. Voit käyttää kumpaakin vaihtoehtoa merkkijonojen merkitsemiseen. Takaperin lainausmerkit (``) ovat uudempi lisäys JavaScriptiin, ja ne mahdollistavat moniriviset merkkijonot ja voivat sisältää ilmaisuja, jotka arvioidaan (interpoloidaan) merkkijonon sisällä. Tämä tekee niistä monipuolisempia kuin yksinkertaiset tai kaksinkertaiset lainausmerkit.

Esim:



b. Catenate two literal strings with a variable. Give two different ways to do it. The end result should be following: I have 36.5 euros. Please note that the amount is from the variable, the text parts are literals.

Esimerkki + käyttäen

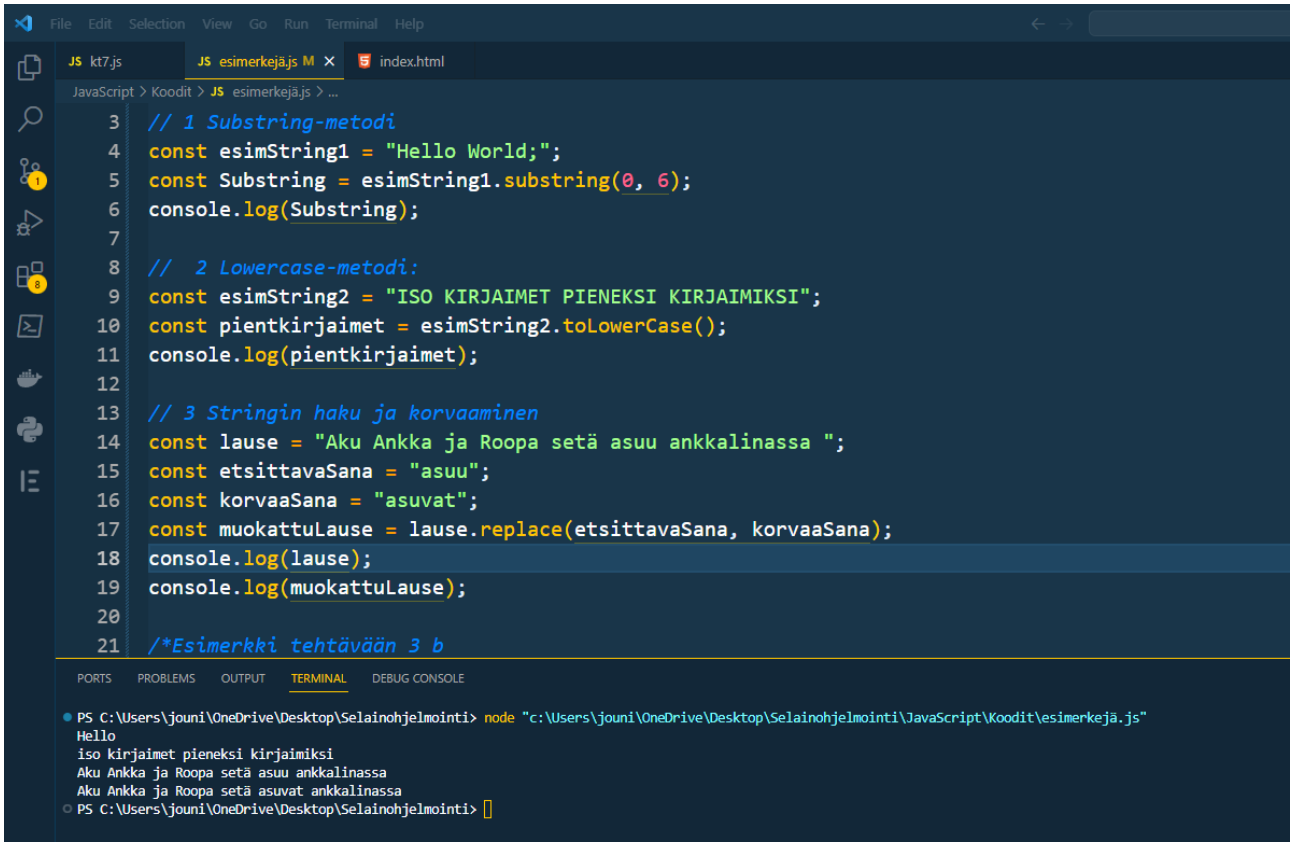


Esimerkki literals käyttäen

c. Give reasons to use the character \ inside strings.

Kauttaviivaa (\) käytetään merkkijonoissa erityisten merkkien, kuten lainausmerkkien tai rivinvaihtojen, käsittelyyn. Se auttaa merkkijonoja olemaan selkeämpiä ja tehokkaampia ilmaisemaan monenlaisia tekstiin liittyviä tarpeita, kuten erityisten merkkien lisääminen, rivinvaihtojen hallinta ja tekstien muotoilu.

d. Give three practical examples of different String methods. Please take care that you include taking a substring and converting the entire string to lowercase into your examples.



Where did you find that information?

Löysin tiedot tutkimalla https://developer.mozilla.org/en-US/docs/Web/JavaScript sivustoa

4. Variables and constants. (1 point)

a. What does it mean that JavaScript variables have no external datatype?

JavaScriptin muuttujat eivät ole sidottuja tiettyyn ulkoiseen tietotyyppiin. Tämä tarkoittaa, että voit muuttaa muuttujan tietotyyppiä sen mukaan, minkä tyyppisen arvon sille annat. Esimerkiksi voit ensin antaa muuttujalle numeron ja sitten merkkijonon, eikä muuttujan tietotyyppi ole ennalta määritelty, vaan se mukautuu arvojen mukaan. Tätä kutsutaan dynaamiseksi tyypitykseksi.

b. What happens if you do not remember use either of the keyword let (or var) when defining a variable? (0,25 points)

JavaScriptissä on pakollista käyttää joko let, var tai const -avainsanaa, kun määrittelet muuttujan. Jos et käytä näitä avainsanoja, saat virheen, kuten "Uncaught ReferenceError: myVariable is not defined." Jokainen avainsana määrittelee muuttujan eri tavalla, esimerkiksi let luo muuttujan, jonka koko on lohko, const tekee muuttujasta muuttumattoman, ja var käyttää funktion tasoista määrittelyä. Valitse avainsana tarpeidesi mukaan.

c. Is there any differences between the keywords `var` and `let` when defining a variable? (0,5 points)

var ja let eroavat JavaScriptissä muuttujien näkyvyyden ja käyttäytymisen suhteen: var on funktiotason lokeroitu, voi olla uudelleenmääriteltävä, ja se nousee (hoist) funktion alkuun. let on blokkilokeroitu, ei voi olla uudelleenmääriteltävä samassa lohkossa, ja se ei nouse (hoist) samalla tavalla kuin var. Suositellaan yleisesti letin käyttöä modernissa koodissa sen paremman ennakoitavuuden vuoksi.

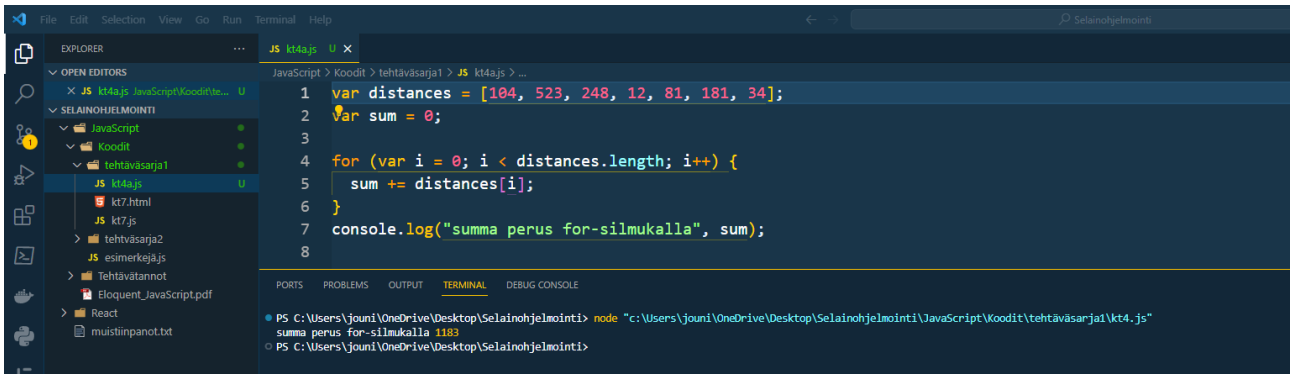
d. How do you define a constant in JavaScript? (0,25 points)

JavaScriptissa voit määritellä vakion käyttämällä const-avainsanaa. Vakio on muuttuja, jonka arvoa ei voi muuttaa sen alustamisen jälkeen. Esimerkiksi const myConstant = 42; määrittää vakion nimeltä myConstant, jonka arvo on 42, eikä sitä voi myöhemmin muuttaa.

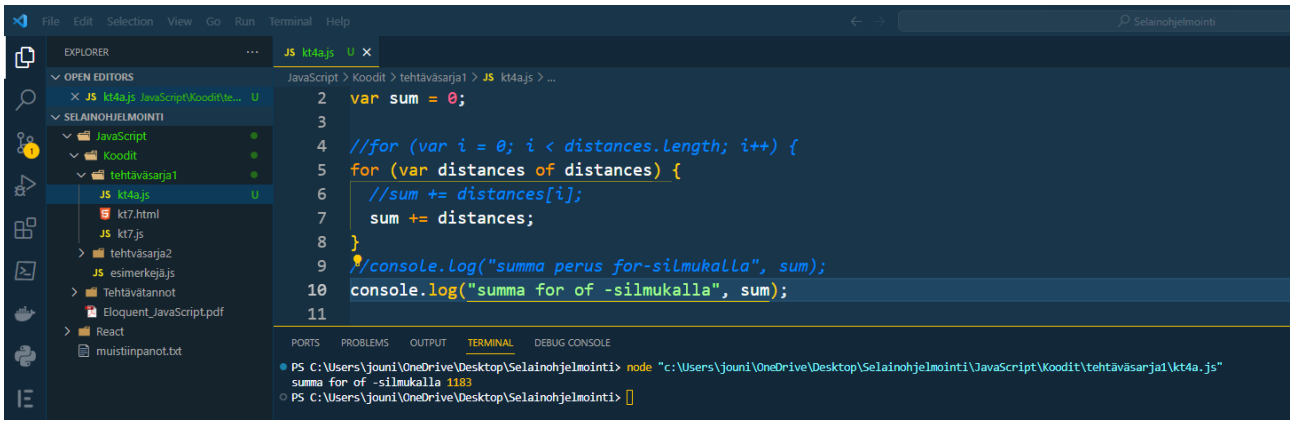
4. Looping in JavaScript. (2 * 0,5 = 1 point):

Let's use the array `distances = [ 164, 526, 248, 12, 81, 181, 34 ].`

a. Use a basic for loop to calculate the sum of the distances. (0,5 points)



b. Use another kind of a for loop to calculate the same sum . (0,5 points)



5. Considering JavaScript arrays. (4 * 0,25 = 1 point)

a. Can an array contain both numbers and objects at the same time in JavaScript?

JavaScriptissä taulukko voi sisältää erilaisia arvoja, kuten numeroita ja objekteja, samanaikaisesti. Taulukot ovat monipuolisia ja voivat pitää sisällään eri tietotyyppejä, kuten merkkijonoja, numeroita, objekteja, funktioita ja jopa muita taulukoita. Esimerkiksi taulukko voi sisältää sekä numeroita (kuten 1 ja 42) että objekteja (kuten { nimi: "John", ikä: 30 }). Näitä elementtejä voi käyttää taulukon indeksien avulla tarpeen mukaan.

b. Explain what it means to

1) modify an array in place or to

Taulukon muokkaaminen paikan päällä:

Taulukon muokkaaminen paikan päällä tarkoittaa, että alkuperäistä taulukkoa muutetaan suoraan, eikä uutta taulukkoa luoda. Tämä tarkoittaa, että kaikki muutokset vaikuttavat alkuperäiseen taulukkoon. Esimerkki tällaisesta metodista on push-metodi, joka lisää yhden tai useampia alkioita taulukon loppuun:

2) return a modified copy of an array. Give one example of a JavaScript method that modifies an array in place, and one example of a method that returns a modified copy.

Muokatun kopion palauttaminen tarkoittaa, että alkuperäinen taulukko säilyy muuttumattomana, ja uusi taulukko luodaan muutoksilla. Tämä mahdollistaa alkuperäisen taulukon säilyttämisen muuttumattomana samalla kun työskennellään muokatun version kanssa. Metodeja, jotka palauttavat muokatun kopion taulukosta, kutsutaan muuttumattomiksi metodeiksi. Esimerkki tällaisesta metodista on concat, joka luo uuden taulukon yhdistämällä kaksi tai useampia taulukoita:

c. What does it mean that a JavaScript array is mutable? *

JavaScriptissa "muuttuva" tarkoittaa, että voit muokata tai muuttaa taulukon sisältöä sen luomisen jälkeen. Kun sanomme, että JavaScriptin taulukko on muuttuva, se tarkoittaa, että voit lisätä, poistaa tai muuttaa taulukon elementtejä ilman uuden taulukon luomista. Tämä ominaisuus tekee taulukosta joustavan ja käyttökelpoisen, mutta vaatii huolellisuutta, jotta varmistetaan, että koodi toimii odotetusti, erityisesti jos useat osat koodista voivat muokata samaa taulukkoa samanaikaisesti.

d. You have a following code clip. How many arrays do you have in the memory at the end?

```
let array1 = [1,3,5];

let array2 = array1;
```

Tässä koodissa muistissa on yksi taulukko. Sekä array1 että array2 viittaavat samaan taulukkoon, joka sisältää [1, 3, 5]. Ne eivät luo erillisiä taulukoita muistiin, vaan jakavat yhden ja saman taulukon, joten lopputuloksena on vain yksi taulukko muistissa.

Esimerkki koodi 1

6. Working with JavaScript arrays (2 points)

Let's use the array `distances = [ 164, 526, 248, 12, 81, 181, 34 ]`.

a. Write a code clip that returns the length of the array. (0,5 points)

b. Write a code clip that adds the distances `8`, `533` and `76` at the end of the array and in this order. Use one of the array methods. (0,5 points)

c. Write a code clip that removes the number `248` from the array in place. Use an array method. A tip: One of the methods to consider could be `splice`. (0,5 points)

d. Clone the array `distances` to the variable `distances_duplicate`. Use ES6 way: The spread operator. * (0,5 points)

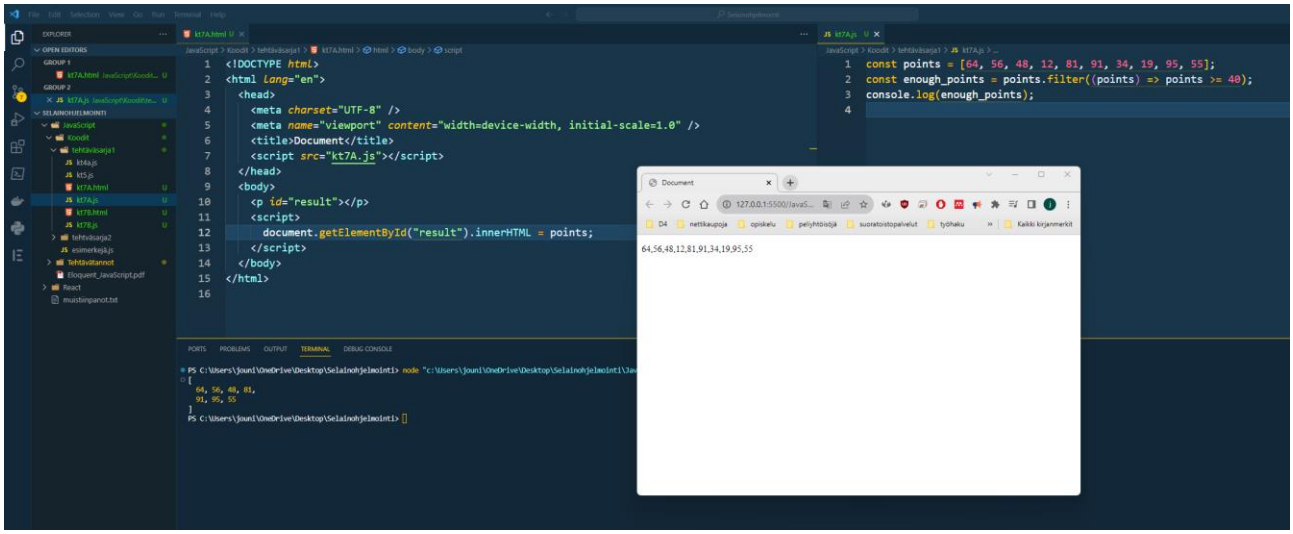
Please, search also the Net to get help.

7. Working with JavaScript Array methods filter, map and reduce (2 points) *

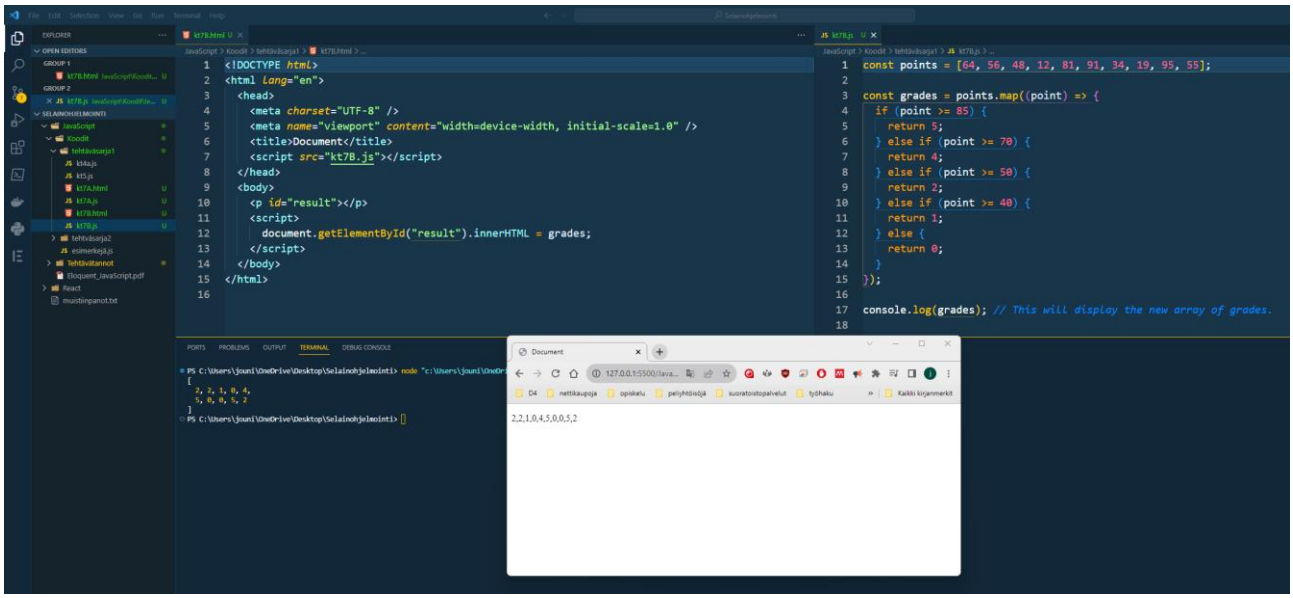
These JavaScript methods are heavily used in modern JavaScript applications.

Let's use the array `points = [ 64, 56, 48, 12, 81, 91, 34, 19, 95, 55 ]`.

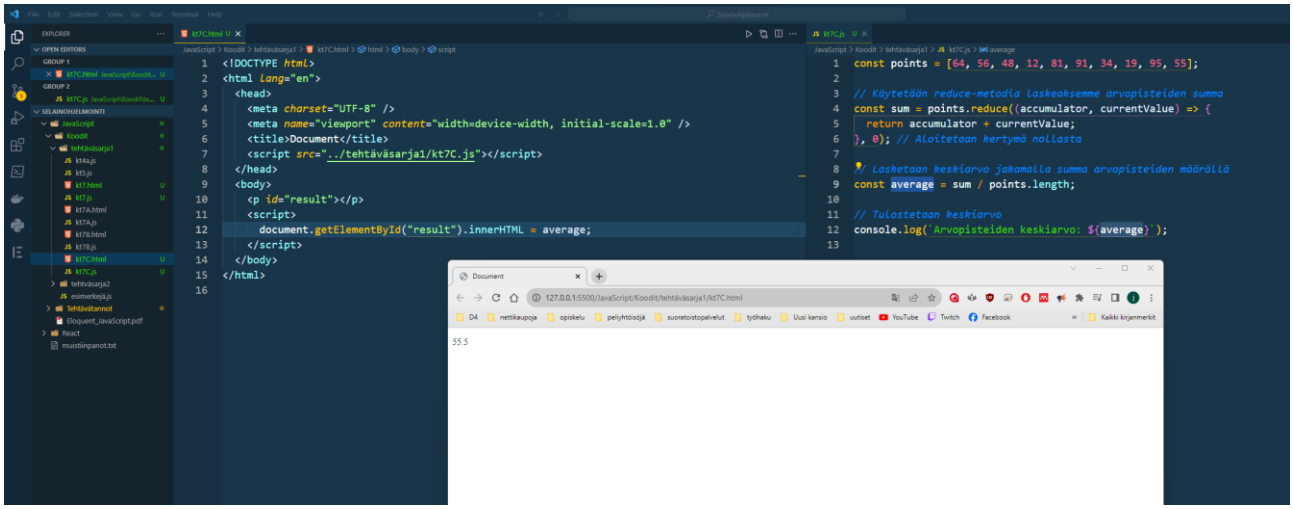
a. Return a new array into a variable called `enough_points`. The new array contains all numbers of the original array `points` that are at least 40. Use the method `filter`. (0,5 points)



b. Return a new array into a variable called `grades`. The new array contains the grades that are calculated from the numbers of the original array `points`. Use the method `map`. The evaluation scale is the following: at least 40 points -> 1; 50 -> 2; 60 -> 3; 70 -> 4; 85 -> 5. Otherwise, the grade is 0. (0,5 points)



c. Calculate the average grade by using a method reduce. Use the original array `points`. (0,5 points)



d. Explain shortly in purposes of the above methods. (0,5 points)

filter: filter-metodia käytetään uuden taulukon luomiseen, jossa ovat vain ne alkuperäisen taulukon elementit, jotka täyttävät tietyt ehdot. Se valitsee ja säilyttää vain ne elementit, jotka täyttävät annetut kriteerit.

map: map-metodia käytetään uuden taulukon luomiseen soveltamalla annettua funktiota jokaiseen alkuperäisen taulukon elementtiin. Se muuttaa alkuperäisen taulukon uudeksi taulukoksi, jossa on muokattuja tai laskettuja arvoja.

reduce: reduce-metodia käytetään taulukon tiivistämiseen yhdeksi arvoksi iteroivasti. Sitä voidaan käyttää erilaisiin tarkoituksiin, kuten summan, keskiarvon tai muiden kertyvien tulosten laskemiseen. Se yhdistää taulukon elementit yhdeksi tulokseksi toistuvien operaatioiden avulla.

Yhteenvetona, filter valitsee tiettyjä elementtejä, map muokkaa elementtejä ja reduce yhdistää taulukon elementtejä eri tietojenkäsittelytehtävien saavuttamiseksi.

Please, search the Net to help.