

Harjoitustehtävä 3: Johdatus JavaScriptiin

Tämän henkilökohtaisen harjoitustehtävän tarkoitus on johdatella opiskelija JavaScriptin maailmaan. Tehtävässä käydään läpi kielen peruskielioppia, mutta toisaalta myös perehdytään JavaScriptin kehitykseen, merkitykseen ja rajoitteisiin.

JavaScript on voimakkaassa muutoksessa, mikä tulee esille myös tässä harjoitustehtävässä. Muutos liittyy etenkin EcmaScript 6 – standardissa vuonna 2015 esitelyihin merkittäviin uusiin ominaisuuksiin. Uusia ominaisuuksia – tosin hieman hillitympää tahtia – on esitelty myös uudemmissa EcmaScript 7, 8 ja 9 standardeissa.

Tehtävä voi hyödyntää seuraavia kirjoja. Marijn Haverbeken kirjasta Eloquent JavaScript 3rd edition (<http://eloquentjavascript.net/>) voi lukea lukuja 1 ja 2. Ethan Brownin Learning JavaScript – kirjasta (3. painos) voi lukea lukua 3, 4 ja 5. Ne ovat erityisen hyödyllisiä varsinkin ES6-kysymysten osalta. Näillä edellä mainituilla kahdella kirjalla pärjää mainiosti. Douglas Crockfordin kirjasta JavaScript – The Good Parts (etsi kirja DawsonErasta) voi puolestaan halutessaan silmäillä liitteitä. A ja B. W3Schoolsin tutoriaali on myös hyödyllinen, mutta kannattaa muistaa kirjoissa mainitut opit ja varoitukset.

Upota teoria- ja koodivastauksesi tähän dokumenttiin. Palauta dokumentti Optiman palautuslaatikkoon kurssiympäristössä ilmoitettuun määräaikaan mennessä.

Keltaisella taustalla olevia kohtia on ainakin osittain tarkoitus tehdä tai käydä tunnilla yhdessä opettajan kanssa.

1. Mitä on JavaScript?

a. Mitä tarkoittaa, että JavaScript on tulkattava kieli?

V: JavaScriptiä voidaan lukea ja suorittaa ilman, että se käännetään konekielelle.

b. Missä JavaScriptiä tulkitaan?

V: mm. Selaimessa

Harjoitustehtävä 3: Johdatus JavaScriptiin

c. Mihin JavaScriptä käytetään?

V: JavaScriptillä voidaan muuttaa staattinen sivu dynaamiseksi ja tekee siitä mahdollisesti myös interaktiivisen.

d. Mikä on EcmaScript?

V: EcmaScript on scriptauskieli joka standardoi JavaScriptiä.

2. Laadi itsellesi html5-pohjatiedosto, jonka kopioita on helppo käyttää JavaScript-tehtävien perustana. Tallenna se nimellä jspohja.html. Tee siitä myös varmuuskopio.

V:

```
<!DOCTYPE html>
<html>
<body>
<script>
// js koodi tulee tähän
</script>
</body>
</html>
```

3. Millä eri tavoin www-sivuun voidaan liittää JavaScriptiä selaimen suoritettavaksi? Minkälaisia hyviä ja huonoja puolia eri tavoilla on? Anna lyhyt koodiesimerkki ainakin 3 eri vasta.

V:

```
<script type="text/javascript" src="..polku js tiedostoon "></script>
```

tai script tageilla

```
<script>
//.....JavaScript tulee tähän
</script>
```

Harjoitustehtävä 3: Johdatus JavaScriptiin

tai ulkoisella js tiedostolla

```
<script src="jokuURL"></script>
```

4. Miten ES5 – standardin mukaisessa JavaScriptissä luodaan globaaleja ja lokaaleja muuttujia? Anna esimerkkejä. Miksi globaalien muuttujien käyttöä tulisi välttää?

V: Lokaali muuttuja:

```
function minunFunktioni() {  
    var autoMerkki = "Audi";  
    // tässä oleva koodi voi käyttää autoMerkki:ä  
}
```

V: Globaali muuttuja

```
var autoMerkki = "Audi";  
// tässä oleva koodi voi käyttää autoMerkki:ä  
function minunFunktioni() {  
    // tässä oleva koodi voi myöskin käyttää autoMerkki:ä  
}
```

// Globaalien muuttujien käyttöä tulisi välttää koska kaikki funktiot pääsevät käsiksi globaaleihin muuttujiin, on vaikeaa löytää mitkä funktiot oikeasti lukevat ja kirjoittavat näitä muuttujia.

5. Luo ES5 – standardin mukaisesti kaksi muuttujaa, joissa toisen arvoksi sijoitetaan yhteenlaskun tulos, ja toiseen merkkijonojen yhdistämisen tulos. Mitä sanot plus-merkin toiminnasta?

V:

```
<!DOCTYPE html>  
<html>  
<script>
```

Harjoitustehtävä 3: Johdatus JavaScriptiin

```
var summa = 4 + 4;
var merkkijonoSumma = ("Hello" + "world!")
console.log(summa);
console.log(merkkijonoSumma);
</script>
</html>
```

Yhteenlaskussa tulos on 8 ja merkkijonojen yhdistämisessä tulos on Helloworld!, plusmerkki siis laskee numeroilla ja merkkijonoilla yhdistää ne peräkkäin.

6. Tutustu ES6 – standardin mukaisen JavaScriptin tapoihin luoda muuttujia ja vakioita. Minkälaisia muutoksia standardin muutos on tuonut mukanaan JavaScriptiin?

V: Vakiot (constants) on uutta ES6:ssa (Esim. Eulerin numero)

7. Kirjoita yksinkertaiseen funktioon ES5 – standardin mukainen for-silmukka, joka palauttaa järjestyksessä n:en fibonaccin luvun. Olkoon ensimmäinen ja toinen fibonaccin luku arvoltaan 1 niitä seuraavat luvut ovat aina kahden edellisen luvun summa. Esim. $f_3 = 1 + 1 = 2$. Kutsu funktiota www-sivun lataamisen yhteydessä ja tulosta n sekä fibonaccin luku www-sivulle.

V:

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write("Fibonaccin 14. luku on " + getNthFibonacciNumberES5(14));
function getNthFibonacciNumberES5(nth) {

    var a = 0, b = 1, temp; //temp == undefined

    for (var i = 0; i <= nth; i++) {
```

Harjoitustehtävä 3: Johdatus JavaScriptiin

```
temp = a;  
a = a + b;  
b = temp;  
}
```

```
return b;
```

```
}  
</script>  
</body>  
</html>
```

8. Toteuta edellinen tehtävä ES5 – standardin mukaista while-silmukkaa käyttäen.

V:

```
<!DOCTYPE html>  
<html>  
<body>  
<script>
```

```
document.write("Fibonaccin 17. luku on " + getNthFibonacciNumberES5(17) + ".");
```

```
function getNthFibonacciNumberES5(nth){
```

```
    var a = 0, b = 1, temp;
```

```
    while (nth >= 0){
```

```
        temp = a;
```

```
        a = a + b;
```

```
        b = temp;
```

```
        nth--;
```

```
    }
```

```
    return b;
```

Harjoitustehtävä 3: Johdatus JavaScriptiin

```
}  
</script>  
</body>  
</html>
```

9. Mitkä ovat JavaScriptin primitiiviset tietotyypit? Kerro myös oliotietotyyppjä. Onko ES6 – standardin mukana tullut mitään uutta?

V: Boolean, Null, Undefined, Number, String, Symbol (uutta ES6). Esimerkki oliotietotyyppistä: `var x = {etuNimi:"Matti", sukuNimi:"Meikäläinen"};`

10. Vastaa seuraaviin kysymyksiin:

- a. Mitä eroa on vertailuoperaattoreilla `==` ja `===`?

V: käytettäessä `===` verrataan tiukasti. Eli sekä tyyppin että arvon pitää olla samat.
esim.

```
5 === 5  
// true  
7 === '7'  
// false (numero V string)  
false === 0  
// false
```

lyyhemmin tarkasteltuna

```
7 == '7'  
// true  
false == 0  
// true
```

- b. Kumpaa edellisistä tulisi käyttää ja miksi?

V: Riippuu tilanteesta. Jos halutaan verrata ilman että otetaan muuttujien tyyppiä huomioon tulisi käyttää `==` operaattoria. Jos taas halutaan verrata kahta asiaa siten, että ne ovat tismalleen samanlaiset tulisi käyttää `===` operaattoria.

Harjoitustehtävä 3: Johdatus JavaScriptiin

c. Luettele ne eri tietotyyppisiä edustavat arvot, joiden totuusarvo on JavaScriptissä FALSE.

V: false, 0, null, "", NaN, ja undefined

d. Mitä \ -merkillä tehdään?

V: \ -merkkiä käytetään ennen erikoismerkkejä, kun niitä halutaan tulostaa.

11. Kommentoi edellisistä ohjelmointitehtävistä kaksi. Huolehdi koodin sisennyksistä. Käytä monisanaisissa muuttujananimissa kamelinotaatiota.

V: Teht.7

```
H03_Teht5_JonneKaajalahti.js esim.html new 1 H03_Teht7_JonneKaajalahti.html H03_Teht8_JonneKaajalahti.html
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <script>
5 document.write("Fibonacci 14. luku on " + getNthFibonacciNumberES5(14)); // tulostus HTML-dokumenttiin
6 function getNthFibonacciNumberES5(nth) { // luodaan funktio
7
8     var a = 0, b = 1, temp; //temp == undefined
9
10    for (var i = 0; i <= nth; i++) { // fibonacci for-loopissa
11        temp = a;
12        a = a + b;
13        b = temp;
14    }
15    return b; // funktio palauttaa arvon b
16 }
17 </script>
18 </body>
19 </html>
20
```

Teht.8

```
H03_Teht5_JonneKaajalahti.js esim.html new 1 H03_Teht7_JonneKaajalahti.html H03_Teht8_JonneKaajalahti.html
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <script>
5
6 document.write("Fibonacci 17. luku on " + getNthFibonacciNumberES5(17) + "."); // tulostus HTML-dokumenttiin
7 function getNthFibonacciNumberES5(nth){ // luodaan funktio
8     var a = 0, b = 1, temp;
9
10    while (nth >= 0){ // fibonacci while-loopissa
11        temp = a;
12        a = a + b;
13        b = temp;
14        nth--;
15    }
16    return b; // funktio palauttaa arvon b
17 }
18 </script>
19 </body>
20 </html>
21
```

Harjoitustehtävä 3: Johdatus JavaScriptiin

12. Miten hyvin selaimet tukevat ES6 – standardin mukaisia ominaisuuksia? Mitä se tarkoittaa näiden ominaisuuksien käytettävyyden kohdalla? Miksi? (Kannattaa vilkaista myös sivua <https://kangax.github.io/compat-table/es6/>)

V: Eniten käytetyt nykyajan selaimet tukevat hyvin lähestulkoon kaikkia ominaisuuksia (poislukien optimisaation proper tail callit). Ominaisuuksien kannalta se on hyvä sillä kehittäjät voivat käyttää niitä huoletta.