

a. Johdanto

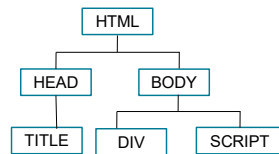
- JavaScript on tulkattava kieli (Interpreter), joka tarvitsee ohjelmien koodauksessa vain teksti editorin ja Web selaimen. Editorin pitää olla sellainen, että se tallentaa niin sanottua puhdasta ascii koodia, jota esimerkiksi lähdekoodin on oltava. Tällaisia editoreja ovat esimerkiksi Notepad++, Sublime ja Visual Code.
- JavaScript on siis selaimessa toimiva ohjelmointikieli. Kaikki selaimet, kuten Chrome, Firefox, Opera jne, tukevat JavaScriptiä. Sitä käytetään paljon erilaisten toimintojen tekemiseen web-sivuille. Näitä voivat olla vaikkapa käyttäjän syöttämän datan tarkistaminen tai mahdollistaa erilaista interaktiivisuutta Web sivustolla. Sen avulla tehdään nykyään paljon erilaisia Web sovellutuksia.
- Kun käyttäjä siirtyy selaimella haluamalleen web-sivulle, alkaa tapahtua paljon ennen kuin käyttäjä näkee mitään selaimessaan. Käyttäjän klikkaaman osoitteen takana oleva Web palvelin lähettää selaimen pyydetyn sivun. Selain puolestaan jäsentää (parseori) ensin sivun html koodista niin sanotun DOM puun (Document Object Model). Tämän jälkeen selain näyttää web sivun käyttäjän selaimessa. DOM puu on tärkeä esimerkiksi JavaScriptille. JavaScript pystyy käyttämään tätä tietorakennetta hyödyksi. JavaScriptillä pystyy esimerkiksi löytämään halutun HTML elementin, jos kyseiselle elementille on annettu yksikäsitteinen id –nimi. Tätä tulemme tekemään tällä kurssilla usein. Tämä etsiminen onnistuu JavaScript:in document –olion getElementById() –metodilla. Sen avulla voi löytää esimerkiksi <div id="result"></div> elementin. JavaScript lause tämän tekemiseksi on document.getElementById("result"), jossa hakusana annetaan metodin lähtötiedoksi lainausmerkkien sisällä. Jos perään lisää vielä innerHTML ominaisuuden, pystyy div elementin sisälle laittamaan haluamaansa tekstiä tai numeroita. Esimerkiksi tekstin Tervetuloa, kuten alla olevassa lauseessa on tehty:

```
document.getElementById("result").innerHTML = "Tervetuloa";
```

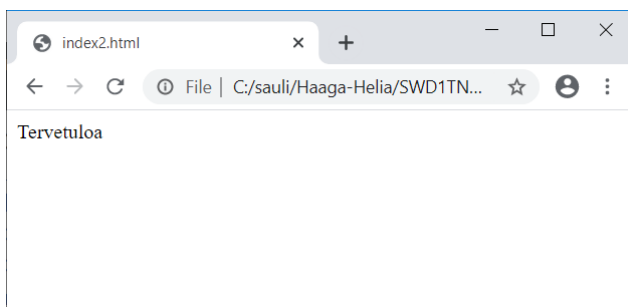
Tämä tarvitsee tietysti toimivan web sivun, joka on seuraavanlainen:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title></title>
5      <meta charset="utf-8">
6  </head>
7  <body>
8      <div id="result"></div>
9  <script>
10     function aja() {
11         document.getElementById("result").innerHTML="Tervetuloa";
12     }
13     aja();
14 </script>
15 </body>
16 </html>
```

Selain tekee tästä DOM puun, joka käyttää alla olevalta tälle html sivulle:



Ja koska yllä olevassa JavaScript ohjelmassa ajetaan aja() funktio, josta puolestaan löytyy rivi `document.getElementById("result").innerHTML = "Tervetuloa";` tulostuu Web selaimen:



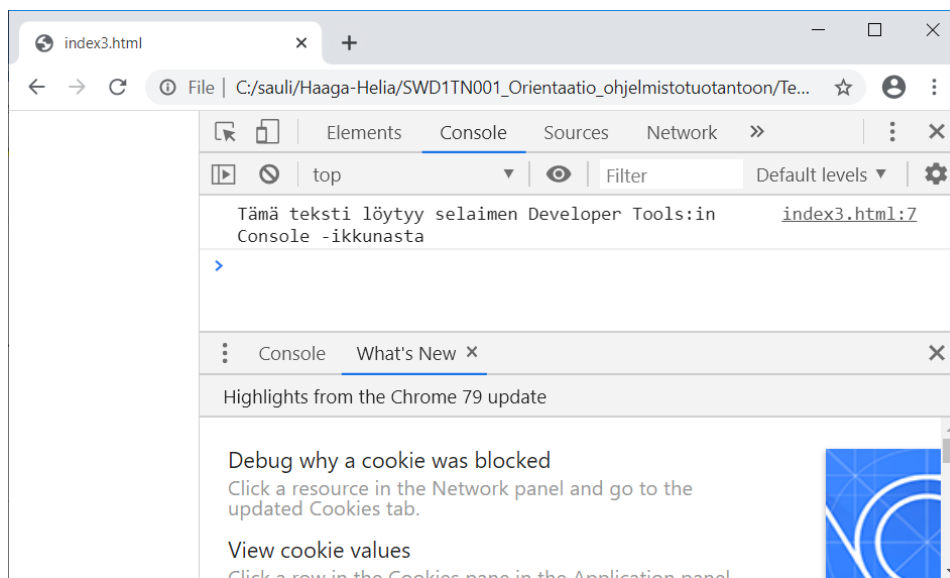
JavaScriptillä pystyy siis tällaiseen ja moneen muuhunkin. DOM puu on tässä tärkeä, sillä se mahdollistaa tavallaan tarttua tai löytää halutun elementin missä tahansa web sivua. Huomaa myös miten script elementtiä on käytetty JavaScript ohjelmoinnissa. Script elementtiin voisi kirjoittaa muunlaistakin scripti kieltä kuin vain JavaScriptiä, mutta tällä kurssilla käytämme vain sitä. JavaScript ohjelmassamme on myös käytetty funktiota aja(), joka on heti määrittelyn jälkeen ajettu käyntiin ajoon selaimessa. Funktiot tulevat tarkemmin myöhemmin esille tällä kurssilla, mutta ilman niitä on JavaScriptillä ohjelmointi hankalaa. Syynä tähän on se, että JavaScript on funktionaalinen kieli, eli funktiot ovat tämän kielen keskiössä.

Muita vastaavanlaisia JavaScript olioita kuin document ovat esimerkiksi console, josta löytyy tällä kurssilla paljon käytettävä log() –metodi. Sillä pystyy tulostamaan halutun tekstin tai numeron web selaimen console-ikkunaan. Console –ikkunaa tarvitaan JavaScript ohjelmien debuggaukseen eli virheiden löytämiseen ja poistamiseen. Se on siis tärkeä web sivujen kehittäjälle. Tulemme käyttämään sitä paljon tällä kurssilla.

console.log() –metodia voisi käyttää esimerkiksi seuraavanlaisesti:

```
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <script>
      // JavaScript ohjelma:
      console.log("Tämä teksti löytyy selaimen Developer Tools:in Console -ikkunasta");
    </script>
  </body>
</html>
```

Chromessa tulos näkyisi Developer Tools:in Console ikkunassa, alla olevan kuvan mukaisesti:

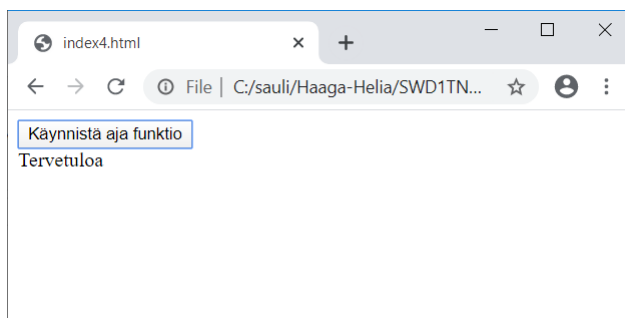


- Yllä olevassa esimerkissä käynnistimme siis JavaScript funktiossa olevan ohjelman lauseella:
aja();
Kuten tuli jo aikaisemmin todettua. Tämä on yksi tapa saada JavaScript funktiossa oleva ohjelma ajettua. Toinen tapa on käyttää selaimessa olevaa painonappulaa. Painonappulalla voi nimittäin käynnistää halutun funktion tapahtuman käsittelijän (event handler) avulla. Tapahtuman käsittelijän avulla voi ohjelmien käynnistämisen antaa käyttäjälle. Usein käynnistys tapahtuu jotain painonappulaa hiiren vasemmalla korvalla klikkaamalla. Tämä onnistuu html:n button elementillä ja sen onclick tapahtuman käsittelijällä:
<button onclick="aja()">Käynnistä aja funktio</button>

Koko web-sivun ohjelma koodi näyttäisi seuraavalta:

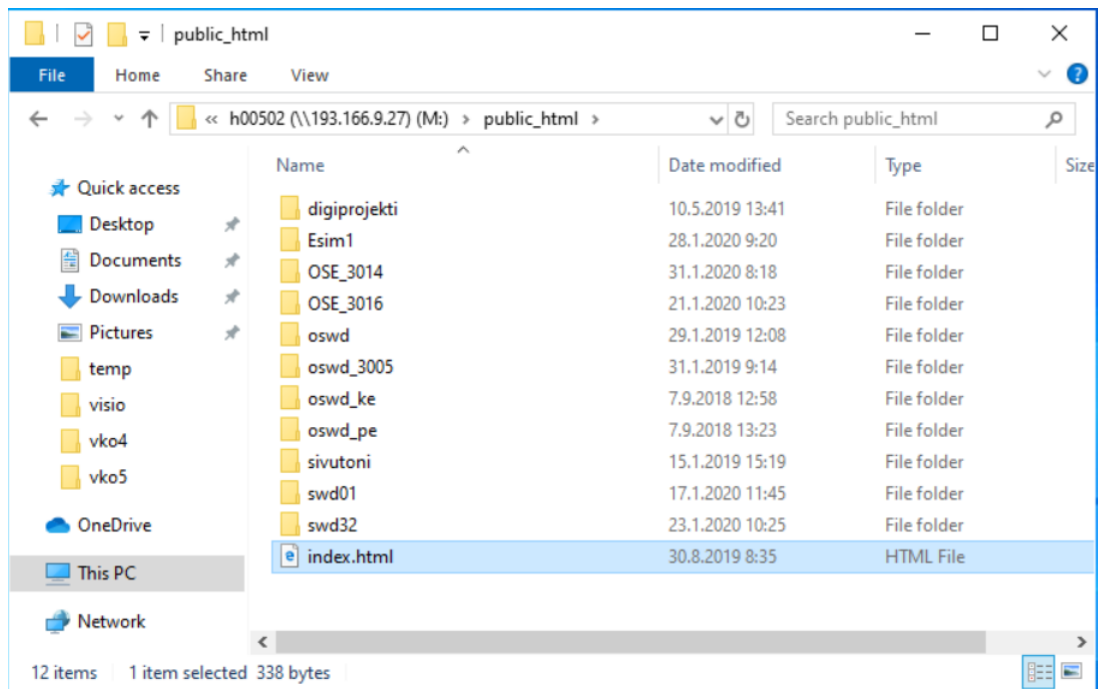
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title></title>
5      <meta charset="utf-8">
6  </head>
7  <body>
8      <button onclick="aja()">Käynnistä aja funktio</button>
9      <div id="result"></div>
10 <script>
11     function aja() {
12         document.getElementById("result").innerHTML="Tervetuloa";
13     }
14 </script>
15 </body>
16 </html>
```

- Jolloin loppu tulos näyttäisi alla olevalta, kun painonappulaa on klikattu:



b. Ensimmäinen JavaScript ohjelma

- Työkaluiksi web selaimeen tarkoitetun JavaScript ohjelman tekoon riittää siis teksti editori (esim. Visual Studio Code, Notepad++, Sublime, ...) ja Web selain (Chrome, Firefox, Opera, Safari, Edge).
- Lisäksi tarvitaan validaattori, jolla voi varmistaa, että tehty html koodi ja JavaScript täyttävät selaimien vaatimukset toimiakseen oikein. Esimerkiksi Html5 validator osoitteessa html5.validator.nu on tällainen.
- Kolmantena on hyvä olla olemassa Web –palvelin, jonne voi julkaista valmiita Web sivustoja Internetiin muidenkin nähtäville. Tällä kurssilla meillä on käytettävissä Linux palvelimelle (myy – palvelin) asennettu Apache Tomcat niminen Web –palvelin. Se löytyy koululla Windows koneissa M –aseman public_html –kansioista. Riittää kun html tiedostot ja JavaScript tiedostot kopioi sinne johonkin sopivaan kansioon:



- Seuraavassa esimerkissä esitetään yksinkertainen html sivu elementteineen, jotka minimissään tarvitaan. JavaScript ohjelma itsessään nykyään usein miten kirjoitetaan niin sanottuun ulkoiseen JavaScript tiedostoon, jonka pääte on .js (esim. javascriptOhjelmani1.js). Tällä kurssilla alkuun kirjoitetaan JavaScript ohjelma kuitenkin html sivuun suoraan <script> elementin sisään. Sen voi sijoittaa esim. <head> elementin sisälle tai <body> elementin alkuun tai loppuun. Esimerkiksi jQuery:ssä JavaScript viittaus on usein ulkoiseen JavaScript tiedostoon juuri <body> elementin lopussa.

Esimerkiksi:

```
<html>
  <head></head>
  <body>
    <script>
      // 1. JavaScript ohjelmani
    </script>
  </body>
</html>
```

Tämä ohjelma ei sisällä muuta kuin yhden kommentti rivin, mutta periaate on tämä. JavaScript ohjelma tulee <script> ... </script> elementin sisään.

Seuraavaksi ryhdymme katsomaan hieman vaativampaa JavaScript ohjelmaa. Ohjelmaa, joka on JavaScript funktiossa ja sen pystyy halutessaan käynnistämään Web sivun painonappulaa klikkaamalla.

Alla olevassa esimerkissä käyttäjä voi klikata Click Me painonappulaa käynnistääkseen substitute() –funktion. Kyseinen funktio on puolestaan <script> elementin sisällä, jonne JavaScript ohjelma kirjoitetaan aina kun ne ovat html sivulla suoraan.

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Example</title>
<script type="text/javascript">
    function substitute() {
        var myValue = document.getElementById('myTextBox').value;

        if (myValue.length == 0) {
            alert('Please enter a real value in the text box!');
        }
        document.getElementById('title').innerHTML = myValue;
    }
</script>
</head>
<body>
    <h1 id="title">JavaScript Example</h1>

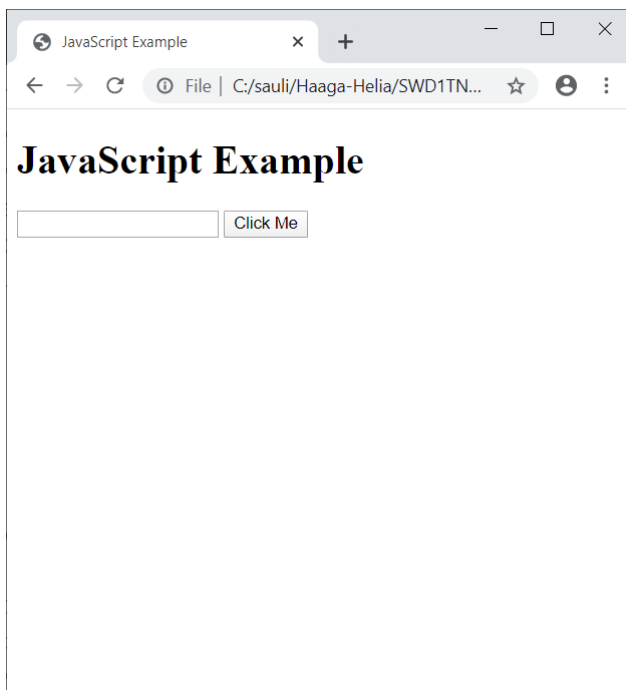
    <input type="text" id="myTextBox" />
    <input type="submit" value="Click Me" onclick="substitute()" />
</body>
</html>
```

- <script> elementin sisälle voisi kirjoittaa mitä tahansa skriptaus kieltä. Selaimet pystyvät niitä tunnistamaan. Type –attribuutilla voi kielen kertoa selaimelle, jolloin sivun toiminta nopeutuu. Näin tehdäänkin tällä kurssilla lähes kaikissa JavaScript tehtävissä jatkossa.
- function avainsanalla ("JavaScript komennolla") puolestaan voi määritellä eli ohjelmoida haluamansa funktion. Funktioita on helppo käynnistää vaikkapa painonappulan klikkauksella web selaimessa. Sen vuoksi haluamme ohjelmoida tässä uuden funktion ja siihen tarvittavan toiminnallisuuden. Funktiolle annetaan usein nimi, niin kuin alla olevassa esimerkissä on annettu tässä tapauksessa substitute(). Jostain muualta sivua, vaikkapa painonappulasta pääse tämän jälkeen käynnistämään tämän funktion. Riittää kun antaa funktion nimen substitute() niin kyseinen funktio ajetaan.
- Itse funktio luo heti ensimmäiseksi muuttujan myValue. Avainsana var (nykyään myös let) luo eli määrittää JavaScript funktioomme uuden muuttujan, johon voi tallentaa ajon ajaksi haluamansa arvon. Tässä esimerkissä talletettava arvo kysytään käyttäjältä tekstilaatikon avulla. Oliosta document nimittäin löytyy getElementById() –metodi, jolla voidaan löytää html sivulta elementti sen id –attribuutin arvon perusteella. Alla olevassa esimerkissä onkin tekstilaatikko, jonka id:n arvo on myTextBox. Käyttäjän syöttämä arvo voidaan sieltä lukea .value ominaisuuden avulla. Operaattorilla = sijoitetaan eli talletetaan muuttujan arvoksi haluttu arvo (tässä siis tekstilaatikossa oleva käyttäjän syöttämä arvo).

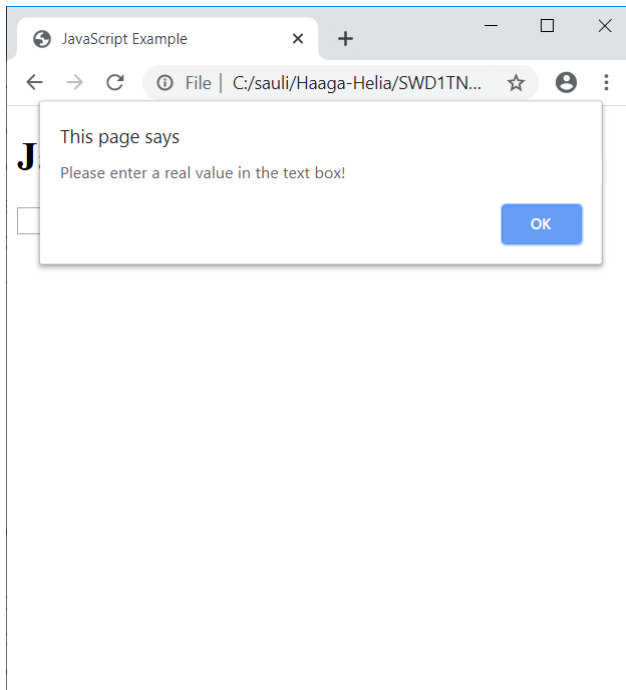
- if –lause tulee tällä kurssilla myöhemmin. If –lauseen avulla voidaan esimerkiksi tarkistaa täyttääkö käyttäjän antama arvo halutut kriteerit. Voimme esimerkiksi tutkia, antoiko käyttäjä tekstilaatikkoon yhtään mitään. Tämä onnistuu lauseella `myValue.length == 0`. JavaScriptissä on jokaisessa muuttujassa `length` ominaisuus, joka tietää muuttujassa olevan arvon pituuden merkkeinä.

Operaattoria `==` puolestaan käytetään tässä yhtäsuuruuden vertailuun. Sen avulla voi esimerkiksi selvittää antoiko käyttäjä mitään arvoa tekstilaatikkoon. Jos `myValue` muuttujan sisältämän arvon pituus merkkeinä on nolla, niin käytännössä se tarkoittaa, että mitään ei ole syötetty tekstilaatikkoon.

Web selaimessa esimerkkinä näyttäisi seuraavanlaiselta:



Jos Click Me –painonappulaa klikkaisi nyt, tulisi esille alla oleva ponnahdus ikkuna:



Tekstilaatikkoonhan ei syötetty mitään, joten if lauseessa päädytään tilanteeseen, josta löytyy alert() funktio. alert() funktio on myös JavaScriptistä löytyvä funktio, jota voi käyttää esimerkiksi viestien näyttämiseen käyttäjälle, kuten yllä olevassa kuvassa on tehty. if –lauseesta tulee myöhemmin lisää tällä kurssilla.

substitute() funktio käynnistetään esimerkissämme Web sivun click me painonappulaa klikkaamalla. Lause html sivulla näyttää tältä:
`<input type="submit" value="Click Me" onclick="substitute()" />.`
Siitä löytyy tapahtumankäsittelijä onclick, jolle on annettu arvoksi itse tekemämme substitute() funktion nimi. Nyt kun käyttäjä klikkaa tätä painonappulaa, käynnistyy substitute() funktio, jolloin ajetaan kaikki ne rivit mitä sen sisältä löytyy.

Tapahtumankäsittelijällä (Event Handler) tarkoitetaan kaikkia niitä tapahtumia, mitä voit tehdä. Painonappulan klikkaaminen hiiren vasemmalla korvalla on yksi tällainen tapahtuma. Muita tapahtumia ovat tietysti hiiren oikean korvan klikkaaminen, jolloin valitaan siihen sopiva tapahtuman käsittelijä. Näppäimistöltä löytyy lisää tapahtumia, joita tapahtuman käsittelijä kuuntelee. Tällä kurssilla tarvitsemme lähinnä vain onclick tapatumaa.

Suosittelutavinta olisi jos JavaScript ohjelma kirjoitettaisiin aina niin sanottuun ulkoiseen JavaScript tiedostoon. Tämän ulkoisen JavaScript tiedoston nimi annetaan script elementtiin src attribuutilla. Alla esimerkki tästä:

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Example</title>
  <script type="text/javascript"
        src="ohjelmani1.js">
  </script>
</head>
<body>
  <h1 id="title">JavaScript Example</h1>

  <input type="text" id="myTextBox" />
  <input type="submit" value="Click Me" onclick="substitute()" />
</body>
</html>
```

- Tämän jälkeen voi luoda tämän tiedoston vaikkapa Visual Studio Code:lla tai millä tahansa teksti editorilla. Tallenna JavaScript tiedosto nimellä ohjelmani1.js, samaan kansioon missä html tiedostokin on. ohjelmani1.js tiedoston sisältö on seuraavallinen:

```
function substitute() {
    var myValue = document.getElementById('myTextBox').value;

    if (myValue.length == 0) {
        alert('Please enter a real value in the text box!');
    }
    document.getElementById('title').innerHTML=myValue;
}
```

Nyt olemme saaneet erotettua html sivun elementit omaan index.html tiedostoon ja JavaScript ohjelman ohjelmani1.js tiedostoon. Nyt luettavuus on helpompaa. Näin olisi hyvä tehdä aina. Jatkossa tuntitehtävissä teemmekin oman html tiedoston ja oman JavaScript tiedoston.

- Example 2:

Chrome Developer Tools

Chromen Developer Tools:ia voi käyttää Web sivuston ja siihen liittyvän JavaScript ohjelman monenlaiseen debuggaukseen eli virheiden etsimiseen ja myös niiden poistamiseen. Moodlesta löytyy Googlen opetusvideoita näiden työkalujen käyttämiseen. Niihin kannattaa tutustua sillä ne ovat oleellinen asia osata.

