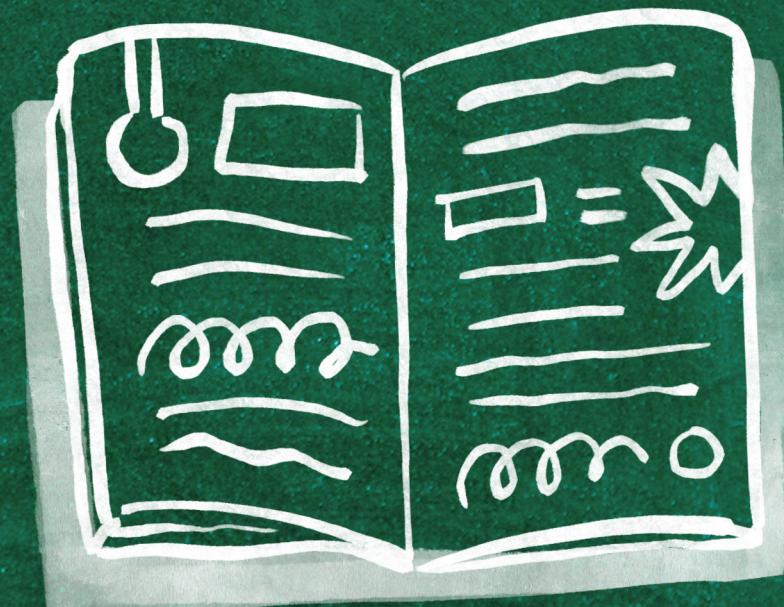


OHJELMOINNIN PERUSTEET

JAVASCRIPT MISSÄ JA MITEN



MIHIN SKRIPTI LAITETAAN?

Samalla tavalla kuin css-määrittelyt voi laittaa osaksi html-koodia, myös JavaScript koodin voi laittaa osaksi html-koodia. Näin on tehty W3Schoolsin esimerkeissä.

Mutta samalla tavalla, kuin css-koodi on järkevintä laittaa omaan tiedostoon, on järkevintä laittaa JavaScript myös omaan tiedostoonsa. Tehdään jatkossa niin.

JavaScript tiedostot ovat muotoa `.js`
esim. **script.js**

Vaikka kahden tiedoston välillä suhaaminen voi tuntua hankalalta, mitä monimutkaisemmaksi koodi käy, sitä pidemmäksi tiedosto muuttuu. Eikä eepoksen selaaminen enää olekaan helppoa.

MITEN SKRIPTIIN LINKATAAN?

CSS-tiedoston linkkaamineen tapahtuu osana html-tiedoston <head> osiota. Aiemmin JavaScript on myös suositeltu laitettavaksi sinne, mutta nykyään suositus on muuttunut.

Jotta skripti voi toimia, pitää sivun ja sen elementtien olla jo latautuneet. Siksi paras paikka on html:n lopussa, juuri ennen body-elementin sulkemista.

```
<footer>Opettajan esimerkki  
sc  
</bo  script  
</html>  script:module  
  script:src
```

```
 <script src="script.js"></script>  
</body>  
</html>
```

Muista käyttää Emmetiä apuna! Pelkillä s ja c kirjaimilla saat oikean ehdotuksen

MITÄ SKRIPTIIN TARVITAAN?

Skriptiin ei varsinaisesti tarvita tiettyä aloitusta, kuten html-tiedostossa.

Koodi kuitenkin kannattaa aloittaa ottamalla käyttöön **Strict mode** eli “tiukka tila”. Se aktivoi enemmän virhetarkistuksia ja estää tekemästä joitakin huonoja valintoja.

```
1 "use strict";
```

Jos esimerkiksi typotan muuttujan x isolla kirjaimella, tiukka tila estää ohjelmaa toimimasta.

```
"use strict";  
  
let x = 3.14;  
X = 2;  
console.log("Luku on " + x);
```

MITÄ SKRIPTIIN MYÖS VOI LAITTAÄÄ?

Toinen, mitä koodiin kannattaa laittaa, on kommentit. Kaikkea itsestäänselvää ei tarvitse kommentoida. Mutta jos on jokin kinkkisempi juttu, joka menee helposti sekaisin muiden kanssa, laita kommentti.

```
// Yhden riviin kommentti näin  
  
/* Useamman  
riviin kommentin saat  
näin */
```

Komentitkin jäävät näkyviin sivun lähdekoodiin, joten mitään asiatonta ei sinnekään kannata laittaa.



MITÄ JOS KOODISSA ON KIRJOITUSVIRHE?

1

Koodissa voi olla monenlaisia virheitä ja usein kirjoitusvirheet estäävät koko skriptiä toimimasta.

Silloin tulee virheilmoitus konsoliin.

Virheilmoituksessa lukee mistä on kyse ja millä rivillä virhe tapahtuu.

```
"use strict";
let x = 3.14;
X = 2;
console.log("Luku on " + x);
```

```
X = 2;
^

ReferenceError: X is not defined
at Object.<anonymous> (c:\Users\jarla\OneDrive - Business College Helsinki\Desktop\testisivu\script.js:4:3)
    at Module._compile (internal/modules/cjs/loader.js:1063:30)
```

Jos virheitä on useita, virheilmoitus näyttää niistä ensimmäisen. Virheilmoituksen muuttuminen on yleensä siis merkki siitä, että ongelman ratkaisu etenee.

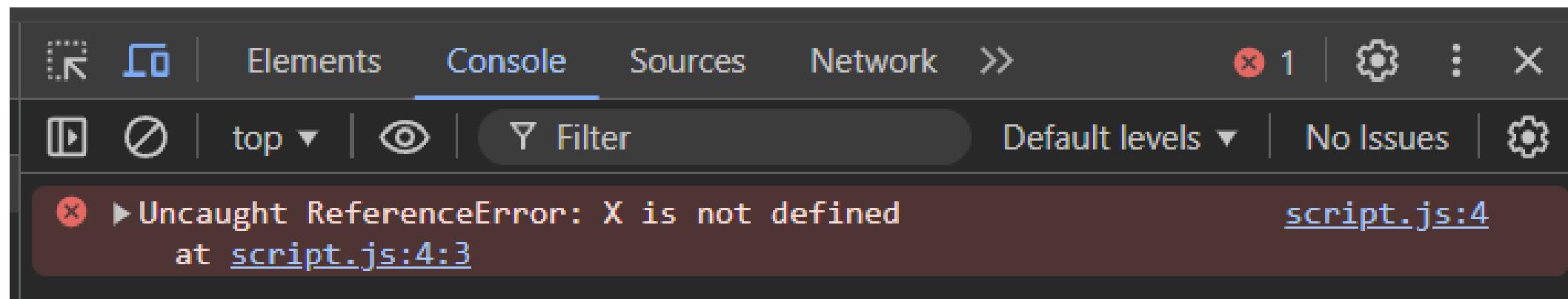
MITÄ JOS KOODISSA ON KIRJOITUSVIRHE?

2

Code runneria käyttäessä Visual Studion Coden konsoli on automaattisesti edessämme. Mutta verkkosivulla konsolia ei oletuksena näy. Selaimen konsoli on osa **kehittäjätökaluja** (eng. *Dev tools*). Avaa kehittäjnäkymä:

- Paina **F12** TAI
- **ctrl+shift+i** TAI
- **hiiren oikea napsautus > Tarkista / Inspect**

ja valitse **Console-välilehti**.



Ainakin Chrome-selaimessa virheet myös näkyvät kehittäjnäkymän ylälaidassa pienenä punaisena merkinä jonka vieressä on numero (jos virheitä ei ole, merkkiä ei näy).

MITÄ JOS KOODIN LOGIIKASSA ON VIRHE ELI BUGI?

Usein ongelmana on, että ohjelma toimii, mutta se toimii väärin. Valitettavasti näistä vioista ei tule varoituksia vaan ne täytyy itse löytää ja osata korjata.

```
"use strict";

let pisteet = prompt("Anna pistemääräsi");
let arvosana = "";

if (pisteet < 10) {
|   arvosana = "hylätty";
}
if (pisteet < 20) {
|   arvosana = "tyydyttävä";
}

console.log("Sait arvosanan", arvosana);
```

Tämä koodi palauttaa arvosanan “tyydyttävä” myös silloin, kun pitäisi palauttaa “hylätty”.

Ehtolauseissa pitäisi ottaa huomioon, että alle 10 on myös alle 20.

MITÄ JOS KOODIN LOGIIKASSA ON VIRHE ELI BUGI?

Väärin toimivan ohjelman löytämistä varten omaa ohjelmaa tulee testata useammalla oikealla arvolla ja muutamalla huonolla arvolla.

Testaus on oma tärkeää osa ohjelmisto-kehitystä ja osa erikoistuu nimenomaan etsimään virheitä, jotka koodaajat sitten korjaavat.

Nyt aiemmin löydetty ongelma on korjattu.

```
"use strict";

let pisteet = prompt("Anna pistemääräsi");
let arvosana = "";

if (pisteet < 10) {
    arvosana = "hylätty";
} else if (pisteet < 20) {
    arvosana = "tyydyttävä";
}

console.log("Sait arvosanan", arvosana);
```