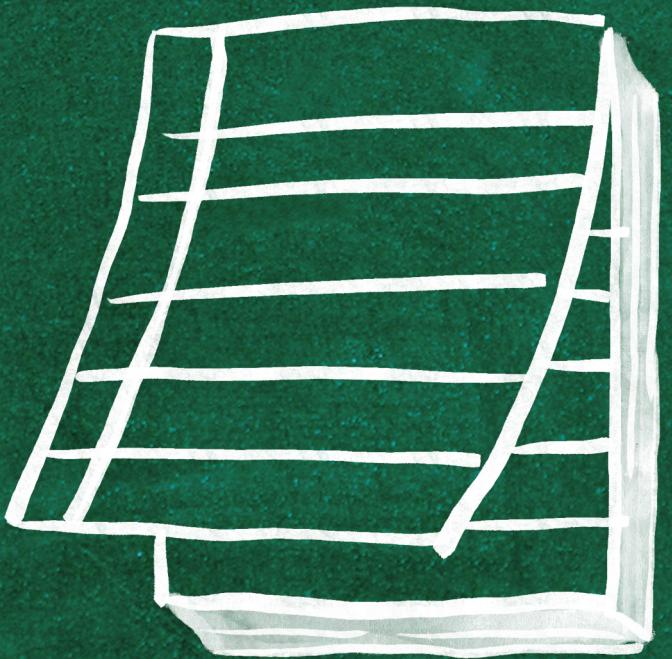


OHJELMOINNIN PERUSTEET

JAVASCRIPT TAULUKOT ALKAA



JS

NÄISSLÄ DIOSSA:

- Mitä taulukot ovat (sis. alkiot ja indeksit)
- Taulukon luonti
- Alkioon viittaaminen
- Alkion arvon muuttaminen
- Taulukon tulostus sellaisenaan
- Taulukon pituus **length**
- Taulukon metodit
- Alkion lisääminen ja poistaminen **push(), pop(), shift(), unshift()**
- Alkion lisääminen, poistaminen ja/tai korvaaminen **splice()**
- Alkion etsiminen **indexOf(), lastIndexOf()**
- Taulukon tulostus halutulla välimerkillä **join()**
- Kuva metodeista

JS

TAULUKOT

Muuttuihin voidaan tallentaa aina kerrallaan yksi tieto. Joskus on tarpeellista tallentaa joukko tietoa, jota voidaan käsitellä kokonaisuutena.

JavaScriptissä on tästä varten taulukot (eng. array) ja ne muistuttavat Pythonin listoja.

TAULUKOT

Taulukot ovat “lokerikko” muuttujia, jossa jokaisella “lokerolla” on oma numeronsa. Numerointi alkaa nollasta, joten taulukon ensimmäinen lokero on taulukko[0] ja viimeinen taulukko[taulukon pituus – 1]. “Lokeroissa” olevia tietoja kutsutaan **alkioiksi**.

```
let varit = ["keltainen", "vihreä", "sininen", "violetti", "punainen", "oranssi", "musta"]
```

Indeksit	0	1	2	3	4	5	6	...
Alkiot	keltainen	vihreä	sininen	violetti	punainen	oranssi	musta	

Erona on se, että Javascriptin taulukossa voi olla “reikiä”. Esimerkiksi jos taulukossa olisi kolme lukua ja lisäisit uuden luvun lokeroon (indeksiin) 10, tämä ei olisi mikään ongelma. Se on sitten eri asia, onko tämä hyvä idea.

JS

TAULUKON LUONTI

Taulukon voi luoda joko let tai const -sanalla.

- Kun käytät taulukon luomisessa const, JavaScript 'lukitsee' luomasi taulukon kyseiseen nimeen - **mutta taulukon sisältöä voi silti muuttaa.**
- Let-sanan avulla luodut taulukot voi myöhemmin muuttaa vaikka merkkijonoiksi.

Taulukko käyttää **hakasulkeita []** sekä **pilkkuja** erottamaan arvoja:

```
const tyhjäTaulukko = [];
const numerotTaulukko = [1, 2, 3, 4];
const merkkijonoTaulukko = ["a", "b", "c"];
const sekalainenTaulukko = [1, "a", true, null];
```

JS

ALKIOON VIITTAAMINEN

Taulukon tiedot ovat alkioita. Niihin viitattaessa käytetään hakasulkeita [] ja indeksinumeroida.

```
const autot = ["Saab", "Volvo", "BMW"];
let luvut = [1, 2, 3];

console.log(autot[1]); // tulostaa "Volvo"

let autoNimi = autot[0]; // autoNimi saa arvon Saab
let luku = luvut[2]; // luku saa arvon 3
```

Esimerkissä muuttujat `autoNimi` ja `luku` saavat arvon sen perusteella, mikä alkion arvo on tässä vaiheessa koodia. Jos alkion arvo myöhemmin muuttuu, se ei päivity automaattisesti näihin muuttajiin.

JS

ALKION ARVON MUUTTAMINEN

Alkion arvoa muutettaessa käytetään myös hakasulkeita[], indeksinumeroa sekä yhtäsuuruusmerkkiä = .

Muutos tehdään aika samalla tavalla, kuin muuttujan arvon muuttaminen.

Arvo
muuttuu

```
const autot = ["Saab", "Volvo", "BMW"];
console.log(autot[0]); // Tulostaa "Saab"

autot[0] = "Opel"

console.log(autot[0]); // Tulostaa "Opel"
```

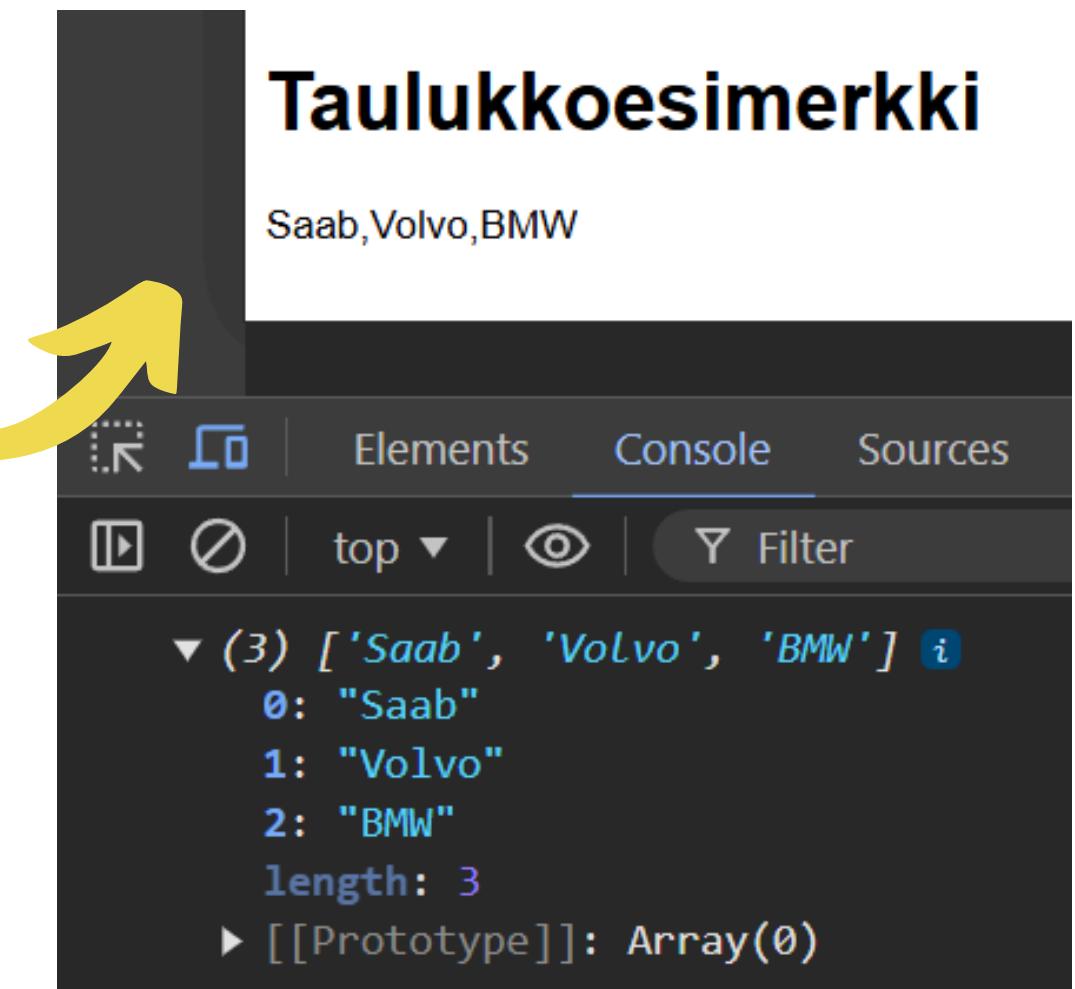
JS

KOKO TAULUKON TULOSTAMINEN SELLAISENAAN*

Taulukon voi tulostaa kokonaisuudessaan esimerkiksi näin:

```
const autot = ["Saab", "Volvo", "BMW"];
console.log(autot);
document.getElementById("testi").textContent = autot;
```

*Tämä tulostuu selaimessa sivulle
kohtaan, jonka id on "testi"*

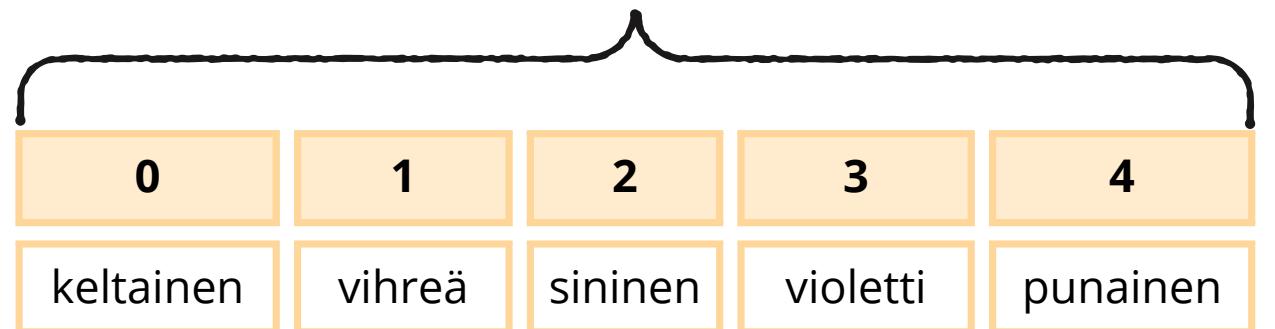


(* toinen vaihtoehto myöhemmässä diassa)

TAULUKON PITUUS

taulukko.length

Taulukon pituus on 5



Taulukon pituuden (*length*) selvittäminen on monesti tärkeää. Taulukon pituus kertoo alkioiden määrän, joka alkaa yhdestä ylöspäin. Muistathan, että indeksi alkaa nollasta!

```
let hedelmat = ["Banaani", "Appelsiini", "Omena", "Mango"];
let pituus = hedelmat.length; // pituus on 4
console.log(hedelmat);
console.log("viimenen alkio on:", hedelmat[pituus - 1]);
```

Taulukon viimeisen alkion indeksi on taulukon pituus - 1.

JS

TAULUKOIDEN METODEJA

Taulukoita voi muokata muutenkin, kuin yksittäisiä alkioita vaihtamalla. Näitä kutsutaan taulukko metodeiksi (eng. *Array method*).

Metodeja käytetään laittamalla taulukon nimen perään piste . ja metodi.

```
let pituus = hedelmat.length; // pituus on 4
```

Seuraavaksi esitellään joitakin yleisimpiä taulukkometodeja.

JS

ALKION LISÄÄMINEN TAI POISTAMINEN

Lisää tai poista elementtejä taulukon **lopusta**
.push() / .pop()

```
let hedelmat = ["omena", "banaani"];
hedelmat.push("appelsiini"); // ['omena', 'banaani', 'appelsiini']
hedelmat.pop(); // ['omena', 'banaani']
```

Lisää tai poista elementtejä taulukon **alusta**
.unshift() / .shift()

```
hedelmat.unshift("mansikka"); // ['mansikka', 'omena', 'banaani']
hedelmat.shift(); // ['omena', 'banaani']
```

JS

ALKIOIDEN LISÄÄMINEN, POISTAMINEN JA / TAI KORVAAMINEN

Lisää, poistaa tai korvaa elementtejä missä tahansa indeksissä.

Sulkujen sisään määritellään

- ensin indeksipaikka
- sitten kuinka monta poistetaan (valinnainen)
- ja lopuksi mitä lisätään (valinnainen).

.splice()

```
let hedelmat = ["Banaani", "Appelsiini", "Mango"];

// Muokataan indeksistä 2 alkaen, ei poisteta mitään, lisätään "Sitruuna" ja "Kiivi"
hedelmat.splice(2, 0, "Sitruuna", "Kiivi");
console.log(hedelmat); // Tulostaa 'Banaani', 'Appelsiini', 'Sitruuna', 'Kiivi', 'Mango'

// Muokataan indeksistä 1 alkaen, poistetaan yksi, ei lisätä mitään
hedelmat.splice(1, 1);
console.log(hedelmat); // Tulostaa 'Banaani', 'Sitruuna', 'Kiivi', 'Mango'
```

JS

ALKION ETSIMINEN

Etsi elementin **ensimmäisen** esiintymän indeksin
.indexOf()

```
const numerot = [1, 2, 3, 2, 1];
numerot.indexOf(2); // 1
```

Etsi elementin **viimeisen** esiintymän indeksin
.lastIndexOf()

```
numerot.lastIndexOf(2); // 3
```

JS

TAULUKON TULOSTAMINEN HALUTULLA VÄLIMERKILLÄ

Yleensä taulukon sisältö halutaan tulostaa niin, että alkioiden välissä on pilkun lisäksi myös välilyönti. Joskus halutaan jokin muu välimerkki kuin pilkku.

.join()-metodi palauttaa taulukon alkiot merkkijonona. Alkioiden väliin laitetaan parametrina eli sulkeiden sisällä annettu merkki - muista myös lainausmerkit.

```
let hedelmat = ["Banaani", "Appelsiini", "Omena", "Mango"];
document.getElementById("demo").innerHTML = hedelmat.join(" ");

document.getElementById("demo2").innerHTML = hedelmat.join(" ja ");
```



Banaani Appelsiini Omena Mango
Banaani ja Appelsiini ja Omena ja Mango

alkutilanne

0 1 2 3



.push()



.pop()



.unshift()



.shift()



.splice(2, 1,)



.length

lopputulos

0 1 2 3 4



4

3

2

X + O + ▲ + ★