

# JavaScript ohjelmointi

## *Ehtolauseet jatkuvat*

MARGIT TENNOSAARIN MATERIAALISTA  
LAURA JÄRVISEN MUOKKAAMA

BTW: KURSSIN REPON  
ETUSIVULLE ON LISÄTTY  
LINKIT TEORIA-  
OSUUKSIIN KANSIOIDEN  
SISÄLLÄ

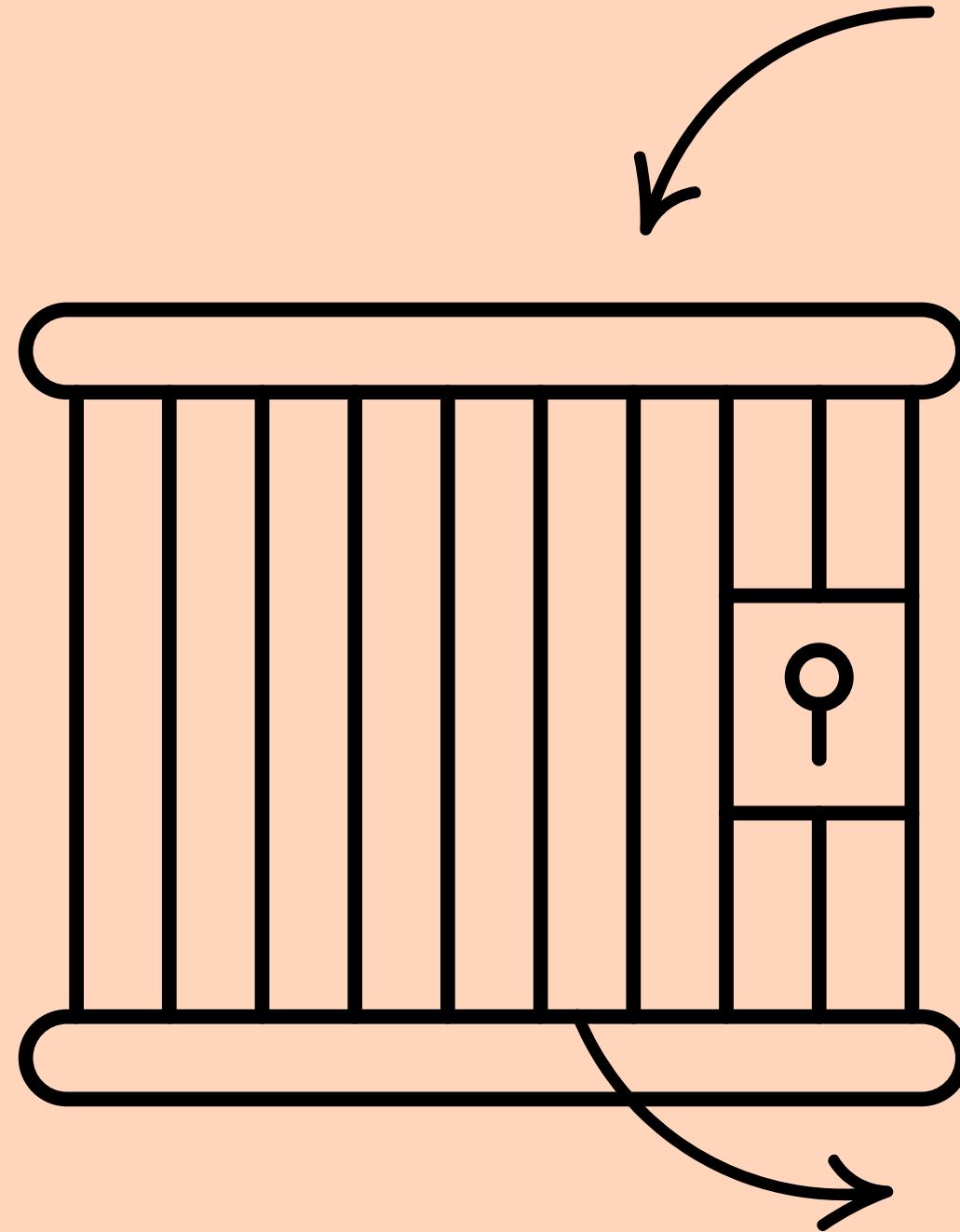
Linkit teoriaosuuksiin kansioiden sisällä:

- [Muuttujat](#)
- [Tietotyypit](#)
- [Numerot](#)
- [Operaattorit](#)
- [Merkkijonot](#)
- [Funktioiden perusteet](#)
- [Ehtolauseet](#)

# Kertaus: Funktiot

Jos tämä auttaa: Funktiota voi ajatella häkkinä, jonne ei kuka tahansa pääse sisään tai ulos.

1 Funktiota pitää kutsua, jotta se tekee mitään. Muuten se vain odottelee.



2 Jos funktion sisälle halutaan antaa tietoja, ne annetaan funktiokutsussa.



3 Jos funktion halutaan antavan jotain ulos lopputuloksena, se tehdään **return**-sanalla\*.

\*Paitsi jos kyseessä on lyhyt nuolifunktio

# Kertaus: Funktiot

Funktiot vaikuttavat nyt irrallisilta, koska emme ole vielä liittäneet niitä mihinkään järkevään. Funktiota on tehty vain harjoituksen vuoksi.

Pian niitä kutsutaan verkkosivun kautta ja niiden hyödyllisyys pääsee esiin. Sitten saatte tehdä funktiota koko rahan edestä.

# Kertaus: Ehtolauseet

Käytetään silloin kun ohjelman toimintaa kuvataan sanoi

*“jos.... niin....”*

Ehtolauseessa on nimensä mukaisesti aina vähintään yksi **ehto**,  
jonka toteutumista tarkastellaan

Usein mukana on myös muita vaihtoehtoja.

# Kertaus: Ehtolauseet

- Voidaan suorittaa koodi, jos ehto toteutuu

if ... -lause

- Voidaan antaa vaihtoehto, mitä tehdään, jos ehto ei toteudu

if ... else -lause

- Voidaan antaa useampi ehto, joiden toteutumista tarkastellaan

if ... else if ... -lause

- Ehtojen määrää ei ole rajoitettu

if ... else if ... if ... else if ... if ... else if ...  
if ... else if ... if ... else if ... else-lause

# Kertaus: Ehtolauseet

```
if (lightColor === "red") {  
  console.log("Pysähdy.");  
}
```

If-lauseet ilman *elseä* voivat  
kaikki toteutua

```
if (isPedestrianCrossing) {  
  console.log("Odota jalankulkijaa.");  
}
```

```
if (isPoliceNearby) {  
  console.log("Aja extravarovasti, poliisi on lähellä.");  
}
```

```
if (isEmergency) {  
  console.log("Poikkeustilanne - voit ajaa varoen vaikka valo olisi punainen.");  
}
```

# Loogiset operaattorit ehtolauseessa

- Yhdistä useita ehtoja samaan lauseeseen

## **&& (AND) Operaattori**

- Kaikkien ehtojen tulee toteutua
- Esim. if (age >= 18 && hasID)

## **! (NOT) Operaattori**

- Kääntää ehdon
- Esim. if (!isRaining)

## **|| (OR) Operaattori**

- Ainakin yhden ehdoista tulee toteutua
- Esim. if (age >= 18 || hasPermission)



# Ternary operator

## eli ternäärinen / ehdollinen operaattori

- Lyhyempi tapa kirjoittaa if...else - lause
- Hyödyllinen yksinkertaisille ehdoille. Menee helposti sekavaksi, jos näitä ketjuttaa

```
ehto ? josTosi : josEpätosi
```

```
let viesti = ika >= 18 ? "Aikuinen" : "Lapsi";  
console.log(viesti); // Tulostaa: "Aikuinen"
```

# Ternary operator jatkuu

## eli ternäärinen / ehdollinen operaattori

- Käytetään usein totuusarvon eli booleanin kanssa

```
let message =  
  isLoggedIn == true ? "Tervetuloa takaisin!" : "Kirjaudu sisään jatkaaksesi.";
```

- Totuusarvon kanssa ei tarvitse kirjoittaa “on tosi”, eli kirjoita näin:

```
let message = isLoggedIn  
  ? "Tervetuloa takaisin!"  
  : "Kirjaudu sisään jatkaaksesi.";
```

# Parittomien ja parillisten lukujen tunnistaminen

- Käytä jakojäännöstä % luvun parillisuuden/parittomuuden tarkistamiseen

```
// Parillinen  
8 % 2; // 0, eli parillinen  
// Pariton  
7 % 2; // 1, eli pariton
```

# Switch -lause

- Vaihtoehto if...else - lauseelle tapauksessa, jossa yhtä muuttujaa verrataan useisiin arvoihin

```
switch (arvo) {  
    case vaihtoehto1:  
        // koodi jos arvo === vaihtoehto1  
        break;  
  
    case vaihtoehto2:  
        // koodi jos arvo === vaihtoehto2  
        break;  
  
    default:  
        // koodi jos mikään yllä olevista ei täsmää  
}
```

# Switch -lause jatkuu

- **break** lopettaa switch-lohkon
- ilman break-sanaa suoritus valuu seuraavaan caseen ks. “lauantai” ja “sunnuntai”
- **default** on vapaaehtoinen, mutta hyvä lisätä

```
let day = "perjantai"

switch (day) {
  case "maanantai":
    console.log("Viikko alkaa!");
    break;
  case "tiistai":
    console.log("Toinen työpäivä.");
    break;
  case "lauantai":
  case "sunnuntai":
    console.log("Viikonloppu!");
    break;
  default:
    console.log("Tavallinen arkipäivä.");
}
```

# Switch vs. if-lause

- **Käytä switch:** Kun vertailet samaa muuttujaa useisiin vakioarvoihin (esim. jos arvo on huhtikuu)
- **Käytä if:** Kun arvioit erilaisia ehtoja tai arvoalueita (esim. jos arvo on välillä 0 ja 7)

# Ehtolauseiden tiivistäminen

```
function isAdult(age) {  
  if (age >= 18) {  
    return true;  
  }  
  return false;  
}
```

=

```
function isAdult(age) {  
  return age >= 18;  
}
```

- Ehtolauseita voidaan usein kirjoittaa lyhyemmin ilman, että koodin toiminnallisuus kärsii.
- Alussa kannattaa kirjoittaa koodi niin, kuin se on sinulle luontevinta.

# Parhaat käytännöt

- Pidä koodi luettavana; vältä liian monta sisäkkäistä **if...else**-rakennetta.
- Yhdistä ehtoja käyttämällä **&&** (AND) ja **||** (OR).
- Käytä **else**-haaraa kattamaan kaikki muut tapaukset, joita *if* tai *else if* eivät kata.
- Ota huomioon odottamattomat syötteet, kuten *null*, *undefined* tai epätavalliset arvot.
- Käytä ternäristä operaattoria yksinkertaisiin *if...else*-ehtoihin.
- **Switch** sopii hyvin tilanteisiin, joissa käsitellään useita mahdollisia arvoja yhdelle muuttujalle.



# Parhaat käytännöt

- Palauta ehto suoraan, kun se on mahdollista.
- Käytä johdonmukaista koodityyliä virheiden välttämiseksi.
- Lisää kommentteja selittämään monimutkaista logiikkaa ehdoissasi. Ja koodissasi ylipäättään.

# Tee tehtäviä

Ehtolauseetehtäviä on nyt lisätty samaan kansioon.

<https://github.com/bc-web-ohjelmistokehitys/WP25K-JS>