

## ARRAYER

# Sortere arrayer

Arrayobjektet har en innebygd metode som lar oss sortere verdiene i en array. Den bruker det vi kaller *leksikografisk sortering*, som vil si at verdiene blir sortert alfabetisk. Dette fungerer fint for tekster, men vi får problemer med tall.

La oss se på et eksempel:

```
let drikke = ["te", "vann", "kaffe", "appelsinsaft"];  
drikke.sort();  
console.log(drikke);  
// Skriver ut: ["appelsinsaft", "kaffe", "te", "vann"]
```



Her blir ordene sortert som vi ønsker, men hva skjer med tall?

```
let tall = [1, 4, 41, 82, 101, 14];  
tall.sort();  
console.log(tall);  
// Skriver ut: [1, 101, 14, 4, 41, 82]
```



Her blir tallene sortert alfabetisk (leksikografisk), men det er ikke slik vi ønsker å sortere dem. For å kontrollere *hvordan* metoden `sort()` skal sortere verdiene, kan vi lage en *sammenligningsfunksjon* som vi lar `sort()` bruke.

## Snakk!

### Hvorfor blir det feil?

Hvorfor blir tallarrayen ovenfor sortert feil når vi sorterer den alfabetisk?

Sammenligningsfunksjonen skal sammenligne to tall om gangen, la oss kalle dem *a* og *b*. Da kan vi kalle funksjonen for `sammenlignFunksjon(a, b)`. Denne funksjonen fungerer slik:

- Hvis et positivt tall returneres, skal *a* sorteres etter *b*
- Hvis et negativt tall returneres, ellers hvis 0 returneres, skal verdiene *stå der de er*

I noen tilfeller må vi angi kode i sammenligningsfunksjonen slik at den enten returnerer en positiv verdi, en negativ verdi eller null. Men her har vi med tall å gjøre, så da ser du kanskje at vi ganske enkelt kan returnere `a - b`? Funksjonen vil da se slik ut:

```
function sammenlignFunksjon(a, b) {  
  return a - b;  
}
```



For å bruke sammenligningsfunksjonen må vi gi den som en parameter til `sort()`-metoden. Vi gjør altså nesten det samme som ovenfor, men vi må sørge for at vår sammenligningsfunksjon brukes:

```
let tall = [1, 4, 41, 82, 101, 14];  
tall.sort(sammenlignFunksjon);  
console.log(tall);  
// Skriver ut: [1, 4, 14, 41, 82, 101]
```



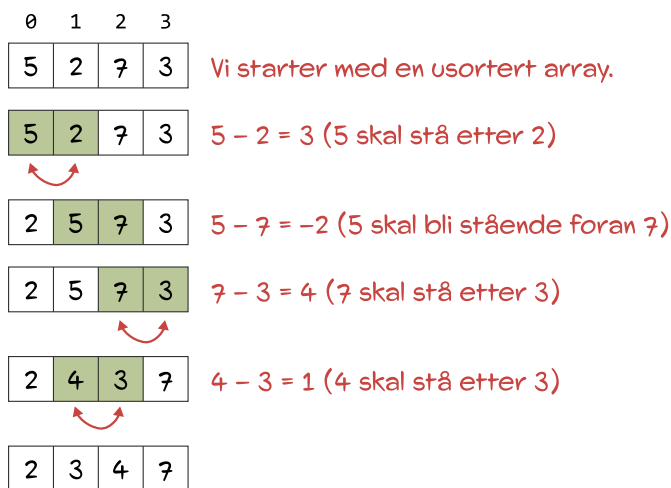
Nå fikk vi sortert tallene som vi ønsker. Husk at vi bare bruker denne funksjonen hvis vi vil sortere tall. Vi kan bruke `sort()` uten parameter om det er tekster vi vil sortere. Og hvis vi ønsker å sortere andre ting, for eksempel objekter som vi skal se på i neste læringsløp, kan vi justere denne funksjonen slik at innholdet vårt sorteres slik vi ønsker.

Hvis du ønsker å sortere verdiene i omvendt rekkefølge, kan du gjøre slik som vi har gjort nå, for så å bruke metoden `reverse()`, som også hører til array-objektet:

```
let tall = [1, 4, 41, 82, 101, 14];
tall.sort(sammenlignFunksjon);
tall.reverse();
console.log(tall);
// Skriver ut: [101, 82, 41, 14, 4, 1]
```



Du kan selvfølgelig også justere sammenligningsfunksjonen om du ønsker det.



Ulike nettlesere bruker ulike sorteringsteknikker, men de tolker resultatet av sammenligningsfunksjonen på samme måte. Et eksempel på en slik sorteringsteknikk, som kalles «bubble sort», illustreres i figuren. Denne skal vi se nærmere på i «3G Algoritmer og effektivitet».

Det kan være lurt å lage en egen JavaScript-fil, for eksempel **mineFunksjoner.js**, der du plasserer viktige funksjoner, som sammenligningsfunksjonen vi har skrevet nå. Da kan du inkludere den i kodene dine, slik at du alltid har tilgang til funksjonene du selv har skrevet. Du kan legge inn ekstra JavaScript-filer slik:

```
<script src="mineFunksjoner.js"></script>
<script>
/* Resten av JavaScript-koden din */
</script>
```



Her bruker vi to `<script>`-elementer. Det ene for å hente inn et eksternt JavaScript-dokument, og det andre for å skrive resten av JavaScript-koden som vanlig.

**10** Lag sorterte versjoner av følgende arrayer:

**a** `let array1 = [2, 1, 7, 5];`

**b** `let array2 = ["melon", "eple", "appelsin", "ananas", "pære"];`

**c** `let array3 = [2, 10, 104, 17, 82, 109];`

**11** Legg til linjen `console.log(a + " - " + b + " = " + (a - b));` i

sammenligningsfunksjonen din. Prøv å sortere en usortert array som består av tall, og se hva som kommer i konsollen. Kan du se hvordan nettleseren din sorterer tallene?

**12** Lag en «omvendt» versjon av sammenligningsfunksjonene du laget i oppgave 10, slik at arrayene blir sortert i omvendt rekkefølge. Klarer du å skrive om funksjonen slik at den kan sortere tekstene i 10b i omvendt rekkefølge?

**Utfordring:** Sorter personene i denne arrayen etter alder:

`let personer = [{"Hans", 12}, {"Nils", 3}, {"Sofie", 5}];` I denne oppgaven må du lage en ny sammenligningsfunksjon.