

Sammendrag

Arrayer

Vi bruker nøkkelordet `let` og hakeparenteser for å lage en array:

```
let minArray = [];
```



En array kan være tom som ovenfor, eller den kan inneholde verdier:

```
let mineTall = [2, 4, 5];
```



Alle verdier i en array har sitt unike indeksnummer. Den første verdien har indeks 0, den andre verdien har indeks 1, den tredje verdien har indeks 2, osv.

Verdiene kan være samme type verdier som vi kan ha i vanlige variabler, men de kan også være andre arrayer:

```
let mineTall = [2, 4, 5, ["en", "to", "tre"], 14];
```



Arraymetoder

Metoden `push()` lar oss legge til en verdi bakerst i en array. Metoden `unshift()` lar oss legge til en verdi i starten av en array.

Metoden `pop()` lar oss fjerne den bakerste verdien i en array. Metoden `shift()` lar oss fjerne den første verdien i en array.

Metoden `splice()` lar oss fjerne verdier inni en array:

```
let tall = [1, 2, 3];  
tall.splice(1, 1);
```



Her fjernes verdien med posisjon 1, altså tallet 2.

Den lar oss også legge til nye verdier inni en array:

```
let tall = [1, 2, 3];  
tall.splice(3, 0, 4);
```



Her legger vi til tallet 4 på posisjon 3. Legg merke til at andre parameter settes til 0. Vi ønsker ikke å slette noen verdier i dette tilfellet.

Vi kan lete etter verdier i en array med metodene `indexOf()`, som finner første forekomst av en verdi, og `lastIndexOf()`, som finner siste forekomst av en verdi.

Løkker og arrayer

Vi kan løpe gjennom alle verdiene i en array med en `for`-løkke:

```
for (let i = 0; i < mineTall.length; i++) {  
  console.log(mineTall[i]);  
}
```



Sortering av arrayer

For å sortere arrayer med tekstverdier kan vi bruke metoden `sort()`. Hvis vi vil sortere tall, må vi lage en egen funksjon for sammenligning av verdier:

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

