Bilde: zoff-photo / iStock

ARRAYER

Lage og redigere arrayer

La oss se på hvordan vi kan lage en array, og hvordan vi kan jobbe med en array etter at vi har laget den.

Vi lager en array omtrent som vi lager en variabel. Vi bruker nøkkelordet `let`, hakeparenteser (`[` og `]`) og komma for å skille verdiene fra hverandre:

```
let tall = [4, 6, 14, 75, 3, 17, 42];
let byer = ["Bergen", "Bodø", "Oslo", "Kristiansand", "Trondheim"];
```

Alle verdiene får sitt unike indeks-nummer som starter på 0 og teller oppover, akkurat som vi så med datatypen string i «3B Variabler, datatyper og operatorer».

Vi bruker hakeparenteser for å finne de ulike verdiene i en array:

```
console.log(tall[0]); // Skriver ut: 4
console.log(tall[3]); // Skriver ut: 75
console.log(byer[2]); // Skriver ut: "Oslo"
console.log(byer[4]); // Skriver ut: "Trondheim"
```

Snakk!

Hvorfor starte på 0?

Kan du tenke deg hvorfor vi teller fra 0 i en array, og ikke fra 1? Tror du det er slik i alle programmeringsspråk? Søk gjerne på internett for å finne ut mer om det.

I JavaScript er en array egentlig et objekt. Arrayen har egenskaper og metoder akkurat som andre objekter. En viktig array-egenskap er `length`, som gir oss arrayens lengde, altså antall verdier i en array. Vi kan bruke `length` til å finne antall verdier i de to arrayene våre:

```
console.log(tall.length); // Skriver ut: 7
console.log(byer.length); // Skriver ut: 5
```

En array kan inneholde tall eller tekst, som ovenfor, men den kan også inneholde arrayer og objekter. En array som inneholder arrayer, kan se slik ut:

```
let matOgDrikke = [
  ["pizza", "pølse", "hamburger"],
  ["kaffe", "te", "brus", "vann"]
];
```

For å få tilgang til verdiene i de «indre» arrayene må vi først finne den arrayen vi er ute etter i hovedarrayen:

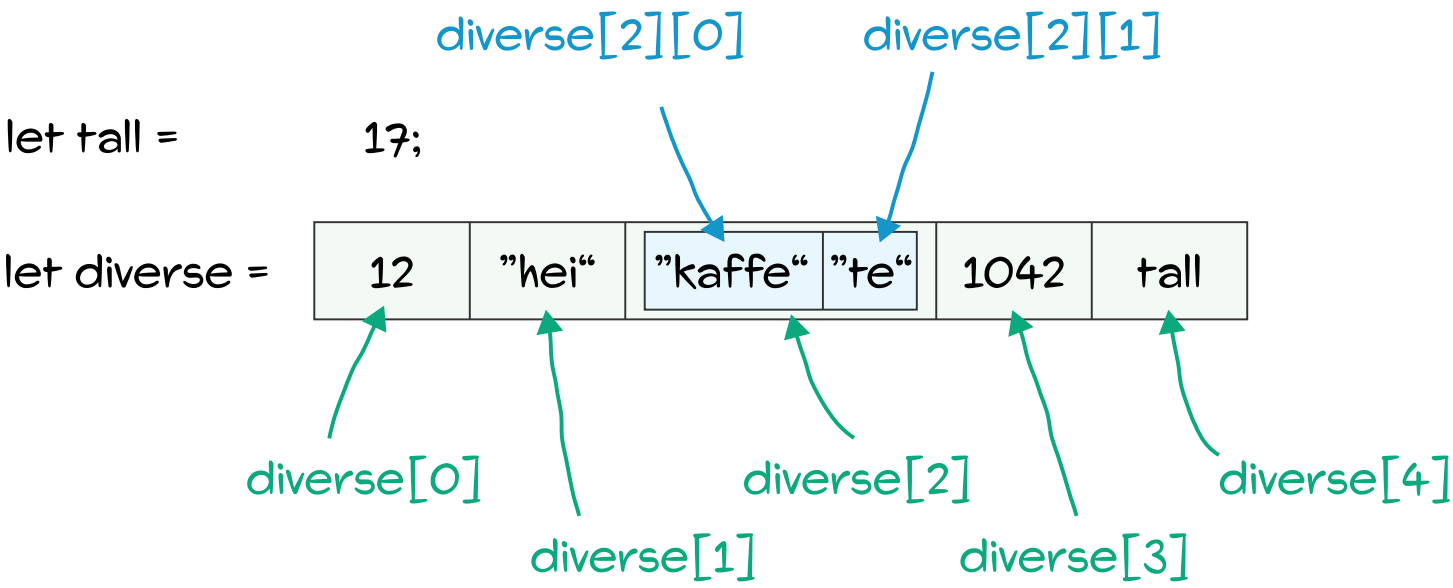
```
console.log(matOgDrikke[0]); // Skriver ut: ["pizza", "pølse", "hamburger"]
console.log(matOgDrikke[1]); // Skriver ut: ["kaffe", "te", "brus", "vann"]
```

Vi ser altså at «maten» ligger i `matOgDrikke[0]`, mens «drikken» ligger i `matOgDrikke[1]`. Vi kan videre hente ut verdiene i de to arrayene slik:

```
console.log(matOgDrikke[0][1]); // Skriver ut: "pølse"
console.log(matOgDrikke[1][0]); // Skriver ut: "kaffe"
```

Hvis hver av verdiene i «hovedarrayen» er en array, kaller vi arrayen vår en `todimensjonal array`. Vi skal se nærmere på dem senere.

Du kan se en oversikt over nummereringen av verdier i arrayer i figuren nedenfor.



Oppgaver

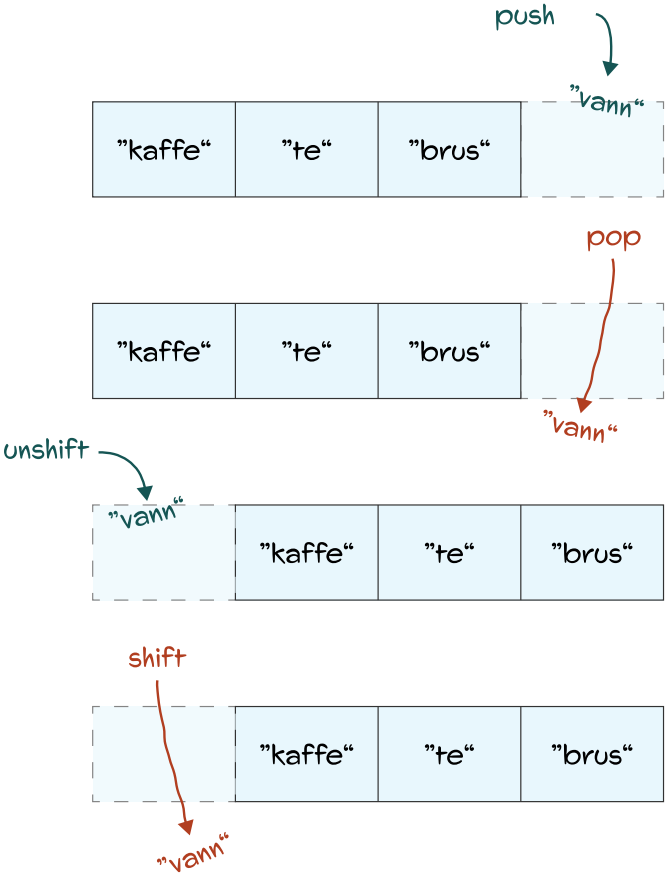
1 Lag en array med tallene 0, 1, 2, 3, 4, 6, 7, 8, 9 og 10 og gi den et passende navn.

2 Skriv ut alle partallene fra arrayen i konsollen.

Redigere arrayer

Det er i hovedsak tre forskjellige endringer vi ønsker å gjøre med én array:

1. redigere verdier som allerede er i en array
2. legge til nye verdier i en array
3. fjerne verdier fra en array



Noen av metodene vi skal se på her, er illustrert i figuren.

For å redigere verdier som allerede er i en array, kan vi bruke samme framgangsmåte som for en variabel:

```
let tall = [4, 2, 5, 8, 12];
console.log(tall[2]); // Skriver ut: 5
tall[2] = 42;
console.log(tall); // Skriver ut: [4, 2, 42, 8, 12]
```

For å legge til nye verdier finnes det to metoder vi kan bruke: `push()`, som legger til en verdi bakerst i en array, og `unshift()`, som legger til en verdi i starten av en array. Vi skal se på en annen metode senere, som lar oss legge til verdier hvor som helst i en array. Vi sender verdien vi vil legge til, som en parameter:

```
let tall = [2, 3, 4];
tall.push(5);
console.log(tall); // Skriver ut: [2, 3, 4, 5]
tall.unshift(1);
console.log(tall); // Skriver ut: [1, 2, 3, 4, 5]
```

For å fjerne verdier finnes det to tilsvarende metoder: `pop()`, som fjerner den siste verdien i en array, og `shift()`, som fjerner den første verdien i en array. Disse metodene krever ikke en parameter:

```
let tall = [1, 2, 3, 4];
tall.pop();
console.log(tall); // Skriver ut: [1, 2, 3]
tall.shift();
console.log(tall); // Skriver ut: [2, 3]
```

Finne og fjerne enkeltverdier

Vi skal senere se hvordan vi kan bruke løkker til å gå gjennom arrayer. Vi kan blant annet bruke løkker til å lete etter verdier i en array. Det finnes også en innebygd metode, `indexOf()`, som lar oss lete etter verdier:

```
let tall = [2, 4, 7, 14, 7, 42];
console.log(indexOf(7)); // Skriver ut: 2
```

Metoden `indexOf()` gir oss den *første* forekomsten av verdien i arrayen. Selv om det er to forekomster av tallet 7 i arrayen ovenfor, får vi bare posisjonen til den første av dem. Vi må derfor bruke en løkke (se neste side) for å finne *alle* forekomstene.

Det finnes også en lignende metode, `lastIndexOf()`, som gir oss den *siste* forekomsten av en verdi.

Vi har sett at det er greit å fjerne første og siste verdi fra en array, men vi kan også få bruk for å fjerne en verdi som er i midten av en array. Til det bruker vi metoden `splice()`. Denne metoden tar to parametre, den første angir indeksen vi ønsker å starte på, og den andre angir hvor mange verdier vi ønsker å ta bort. Hvis vi for eksempel ønsker å ta bort verdien med indeks 2, kan vi skrive `splice(2, 1)`:

```
let tall = [1, 2, 3, 4, 5];
tall.splice(2, 1);
console.log(tall); // Skriver ut: [1, 2, 4, 5]
```

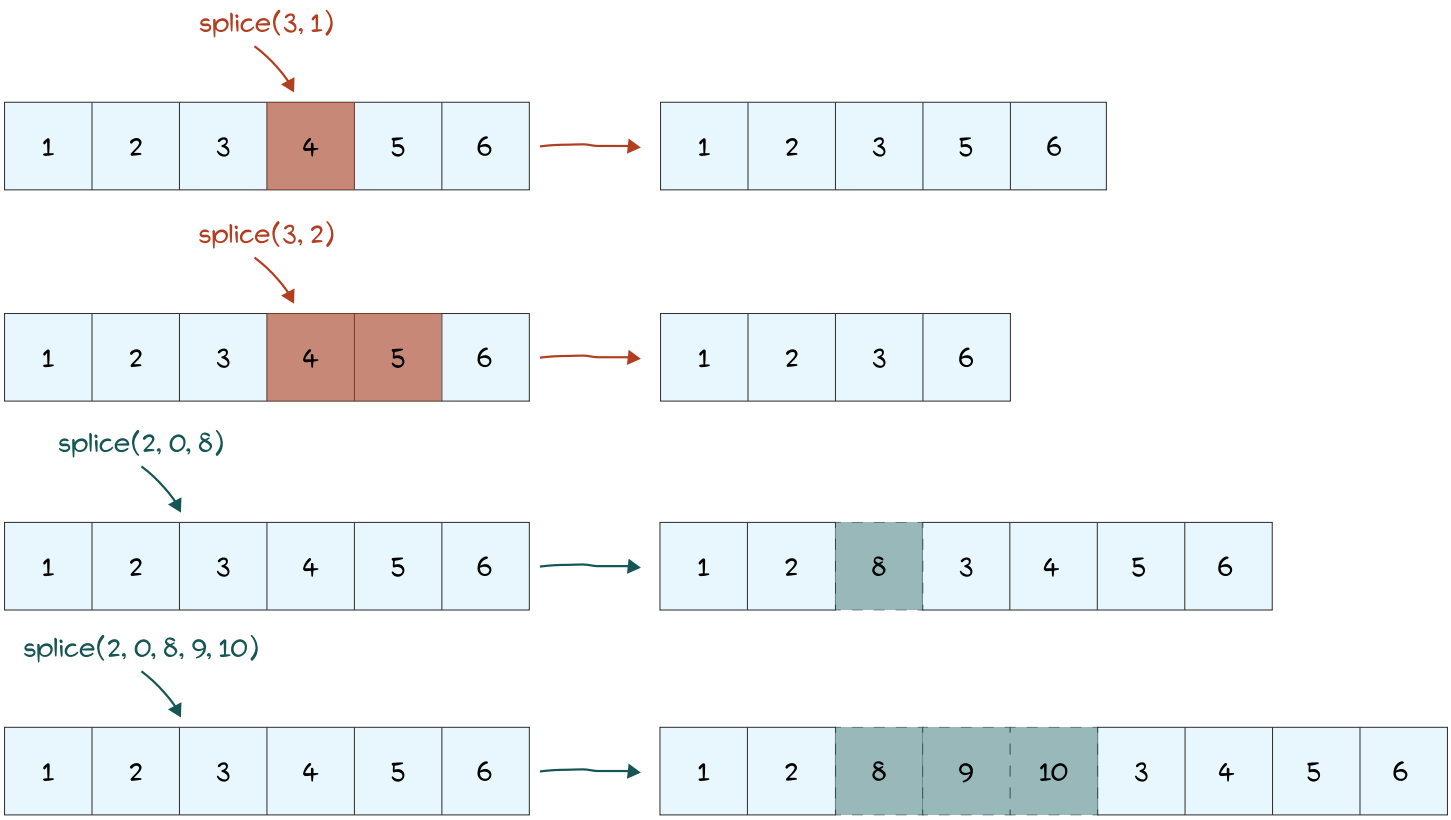
Vi kan kombinere `indexOf()` og `splice()` for å fjerne den første forekomsten av en gitt verdi:

```
let tall = [2, 4, 7, 14, 7, 42];
let fjernIndeks = tall.indexOf(7);
tall.splice(fjernIndeks, 1);
console.log(tall); // Skriver ut: [2, 4, 14, 7, 42]
```

Her bruker vi `indexOf()` til å finne den første forekomsten av tallet 7, og så bruker vi `splice()` for å fjerne tallet fra arrayen. Hvis vi gjentar prosessen, vil også det andre 7-tallet forsvinne fra arrayen.

Metoden `splice()` kan også brukes til å legge til elementer inni en array. Vi bruker fremdeles de samme to parametrene, men vi kan legge til flere parametre. Ekstra parametre blir lagt til i arrayen på posisjonen vi angir som den første parameteren. Fordi vi her ønsker å legge til verdier, er det lurt å sette den andre parameteren til 0, altså at null verdier skal fjernes. Nye verdier blir satt inn før indeksen vi angir som den første parameteren:

```
let tall = [1, 2, 3, 6];
tall.splice(3, 0, 4, 5);
console.log(tall); // Skriver ut: [1, 2, 3, 4, 5, 6]
```



Oppgaver

3. Lag arrayen `let tall = [2, 4, 6, 8];`.

   a. Bruk metodene vi har sett på, og gjør om arrayen tall til `[4, 6]`.

   b. Legg til tallet 5 mellom tallene 4 og 6, slik at arrayen nå inneholder `[4, 5, 6]`.

   c. Gjør om arrayen slik at den inneholder `[3, 4, 5, 6, 7]`.

   d. Gjør om arrayen slik at den inneholder `["tre", 4, "fem", 6, "syv"]`.

4. Igjen utgangspunkt i arrayen du laget i forrige oppgave.

   a. Hva tror du skjer hvis du skriver `tall.indexOf(10)` eller `tall.lastIndexOf(10)` ?

   b. Prøv ut begge variantene og se hva du får. Hvordan tolker du resultatet?

   c. Ta en titt på dokumentasjonen for `indexOf`. Ser du hvorfor metoden returnerer den verdien du fikk?