

Løsningsforslag kapittel 12

12.1 For eksempel:

```
var meg = {  
  navn: "Ola",  
  favorittfilm: "Lego City",  
  favorittall: 3,  
  favorittmat: ["Taco", "Pølser", "Sushi"]  
};
```

12.2 Vi kan hente ut enkeltegenskaper slik:

```
console.log(meg.navn); // eller  
console.log(meg["navn"]);
```

Arrayen kan vi hente ut slik:

```
console.log(meg.favorittmat);
```

Og enkeltverdier i arrayen slik:

```
console.log(meg.favorittmat[0]); // eller  
console.log(meg["favorittmat"][1]);
```

Vi kan også skrive ut hver av enkeltverdiene slik:

```
for (var i = 0; i < meg.favorittmat.length; i++) {  
  console.log(meg.favorittmat[i]);  
}
```

12.3 Bruker begge metodene fra side 191 i én løkke:

```
for (var egenskap in meg) {  
  // Egenskapenes navn  
  console.log(egenskap);  
  
  // Egenskapenes verdier  
  console.log(meg[egenskap])  
}
```

12.4 og 12.5

```
meg.favorittfarge = "lilla"; // eller
meg.favorittfarge = ["lilla", "grønn", "gul"];

console.log(meg.favorittfarge);
```

12.6 I oppgave **12.1** brukte vi bare egenskapen «navn». Vi må bytte ut den med egenskapene «fornavn» og «etternavn» for at denne metoden/funksjonen skal fungere. Vi kan enten lage objektet på nytt, med metoden `fulltNavn()`, eller vi kan legge den til slik:

```
meg.fulltNavn = function() {
  return this.fornavn + " " + this.etternavn;
};
```

12.7 Finner først et tilfeldig tall som representerer en indeks i arrayen `favorittmat`. Skriver deretter ut den tilhørende verdien til konsollen:

```
meg.hvaSkalJegHaTilMiddag = function() {
  var tilfeldig = Math.floor(Math.random() * this.favorittmat.length);
  console.log(this.favorittmat[tilfeldig]);
}
```

12.8

```
var bilmerker = [
  { forkortelse: "VW", bilmerke: "Volkswagen", land: "Tyskland"},
  { forkortelse: "Toyota", bilmerke: "Toyota", land: "Japan"},
  { forkortelse: "Audi", bilmerke: "Audi", land: "Tyskland"},
  { forkortelse: "BMW", bilmerke: "Bayerische Motoren Werke", land:
    "Tyskland"},
  { forkortelse: "Ford", bilmerke: "Ford Motor Company", land: "USA"}
];
```

12.9 Her skriver vi ut egenskapene `bilmerke` og `land`. Forkortelsen kan vi skrive ut på samme måte:

```
for (var i = 0; i < bilmerker.length; i++) {
  console.log(bilmerker[i].bilmerke + " er fra " + bilmerker[i].land +
    ".");
}
```

12.10

```
function sammenlignForkortelser(a, b) {
  if (a.bilmerke > b.bilmerke) {
    return 1;
  } else if (a.bilmerke < b.bilmerke) {
    return -1;
  } else {
    return 0;
  }
}
```

12.11 Denne oppgaven er utfordrende, men den gir oss et verktøy som vi kan få bruk for i mange situasjoner. Funksjonen vi skal lage her lar oss lage en tabell av en hvilken som helst array med objekter (så lenge objektene har like mange egenskaper). Vi har valgt å legge til et argument i tabell-definisjonen, slik at vi også kan angi et element som tabellen skal plasseres i. I tillegg til bilmerker-arrayen fra oppgave 12.8, må vi da også ha et element å plassere tabellen i, og en overskriftsarray:

```
var divEl = document.querySelector("div");
var overskrifter = ["Forkortelse", "Bilmerke", "Land"];
```

Da kan vi lage tabellfunksjonen:

```
function lagTabell(tilEl, overskrifter, objArr) {

  // Først lager vi et table-element og et tbody-element.
  var tabellEl = document.createElement("table");
  var tbodyEl = document.createElement("tbody");

  // Lager en løkke som teller fra -1 til lengden til arrayen
  // med objektene. Vi starter på -1 fordi vi ønsker å lage en
  // rad med overskrifter først. Deretter ønsker vi å legge til
  // alle objektene (som starter på indeks 0, verdien vi får etter -1).
  for (var i = -1; i < objArr.length; i++) {
    // Lager et tr-element
    var trEl = document.createElement("tr");

    // Hvis i er -1 skal vi legge til overskriftsraden
    if (i == -1) {
      // Går gjennom overskriftsarrayen
      for (var j = 0; j < overskrifter.length; j++) {
        // Lager et th-element
        var thEl = document.createElement("th");
```

```
// Legger til en overskrift som tekst i th-elementet
thEl.innerHTML = overskrifter[j];
// Legger til th-elementet i tr-elementet
trEl.appendChild(thEl);
}
// Hvis i ikke er -1, altså (0, 1, 2, 3, ... osv.)
// skal vi legge til verdiene i objektene
} else {
    // Går gjennom egenskapen i objekt nummer i
    for (var egenskap in objArr[i]) {
        // Lager et td-element
        var tdEl = document.createElement("td");
        // Legger til verdien i gjeldende egenskap i td-elementet
        tdEl.innerHTML = objArr[i][egenskap];
        // Legger til td-elementet i tr-elementet
        trEl.appendChild(tdEl);
    }
}

// Legger til tr-elementet i tbody-elementet
tbodyEl.appendChild(trEl);
}

// Legger til tbody-elementet i tabell-elementet
tabellEl.appendChild(tbodyEl);
// Legger til tabell-elementet i elementet
// vi angir når vi bruker funksjonen
tilEl.appendChild(tabellEl);
}
```

For å bruke funksjonen kan vi nå skrive (og denne framgangsmåten kan vi bruke på alle arrayer som inneholder like objekter):

```
lagTabell(divEl, overskrifter, bilmerker);
```

12.12 Denne løser vi ved å lage et div-element og skrive koden:

```
lagTabell(divEl, overskrifter, bilmerker);
```

12.13 Vi får et resultat, men det blir ujevnt (og HTML-koden blir egentlig feil). Det er ikke naturlig å ville lage en tabell fra objekter som ikke inneholder lignende informasjon, så dette vil ikke bli et vanlig problem. Hvis vi ønsker å skrive ut en feilmelding, kan vi sjekke om alle objektene har like mange egenskaper som det er overskrifter i overskriftsarrayen:

```
if (Object.keys(objArr[i]).length !== overskrifter.length) {  
  console.error("Det er feil antall egenskaper i objekt nummer " + i);  
}
```

12.14 Her er løsningen presentert i kapittelteksten.