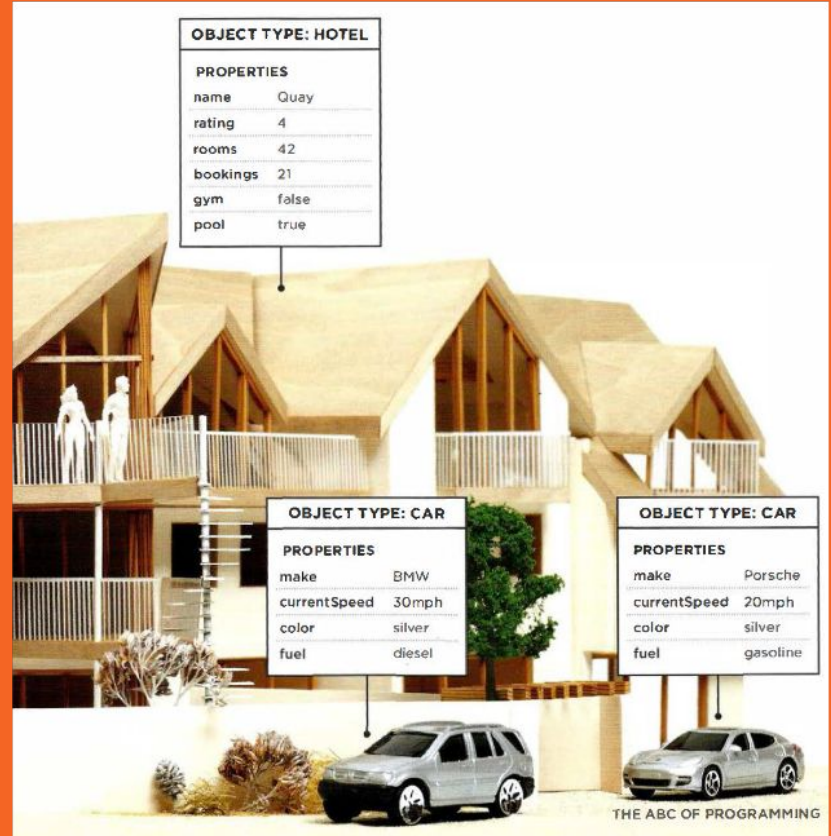

Javascript objekter + HTML <template>

samt kombination af HTML template og array af JS objekter

Hvad er et objekt?

- En repræsentation af noget
 - ofte et objekt i den "rigtige" verden såsom en person, et produkt, et køretøj el.lign. med en række fælles egenskaber
- En abstraktion
- En logisk gruppering
- En datastruktur



Du kender dem allerede bl.a. fra T4

```
// gem DOM-objekt som værdi i en variabel
const figur1 = document.querySelector("container1");

// vi kan tilgå DOM-objektets egenskaber via variabelen
figur1.classList.add("flyv");
figur1.textContent = point;
figur1.innerHTML = "<h1>En helt ny overskrift</h1>";
figur1.classList.remove("flyv");
```

DOM objekter
har mange
egenskaber

Math er et
"indbygget"
JS objekt

Du kan kende dem
på punktummet
(dot notation)

```
blueContainer.addEventListener("click", spillYd);

function spillYd() {
  document.querySelector("#lyd").currentTime = 0;
  document.querySelector("#lyd").play();
}
```

```
myRand = Math.floor(Math.random() * 4);
document.querySelector("#container1").classList.add("delay" + myRand);
```

Datatyper

- Streng
- Tal
- Dato
- Objekter
- Arrays
- Booleans
- Null
- Undefined

Objekter i javascript

- Objekter oprettes ved hjælp af to tængeklammer { }
- De består af *egenskab* : *værdi* par
 - adskilt af et *kolon* :
 - afsluttet med et *komma* ,
- Kaldes også *object initializer* og *object literals*

```
const bil = {  
  brand: "Peugeot",  
  model: "5008",  
  motor: "bensin",  
  passengers: ["Jonas", "Martin", "Louise", "Peter", "Alan"],  
};
```

Arbejde med JS objekter

```
const bil = {  
  brand: "Peugeot",  
  model: "5008",  
  motor: "bensin",  
  passengers: ["Jonas", "Martin", "Louise", "Peter", "Alan"],  
};  
  
console.log(bil.brand); // Peugeot  
console.log(bil.passagerer[0]); // Jonas  
const besked = `Din bil er en ${bil.model}`;  
const passager = `Den tredje pasager er ${bil.passagerer[2]}`;
```

Værdier tilgås med punktum . (dot notation)

Øvelse 1

Et superhelt objekt

1. Lav en ny js-fil i VS Code
 2. Lav et objekt med fx din yndlings superhelt eller band, album, drink el.lign i js-filen
 - a. Overvej hvilke oplysninger og egenskaber du ville gemme fx om en superhelt - har den fx flere særlige evner eller fjender som kunne passe i arrays (data-struktur)?
 3. Gem filen til senere
-

Fra JS objekt til HTML

For at få vist indholdet fra et JS objekt i HTML, skal vi:

1. Vælge et relevant HTML element med `querySelector`
2. Finde det relevante data i objektet med dot notation
fx `bil.brand` eller `bil.passagerer[2]`
3. Udskrifte indholdet af elementet via en relevant attribut (fx `.textContent` eller `.src`) med det tilsvarende indhold fra objektet

```
document.querySelector("h2").textContent = bil.brand;  
document.querySelector("img").src = bil.billede;
```

Fra JS objekt til HTML

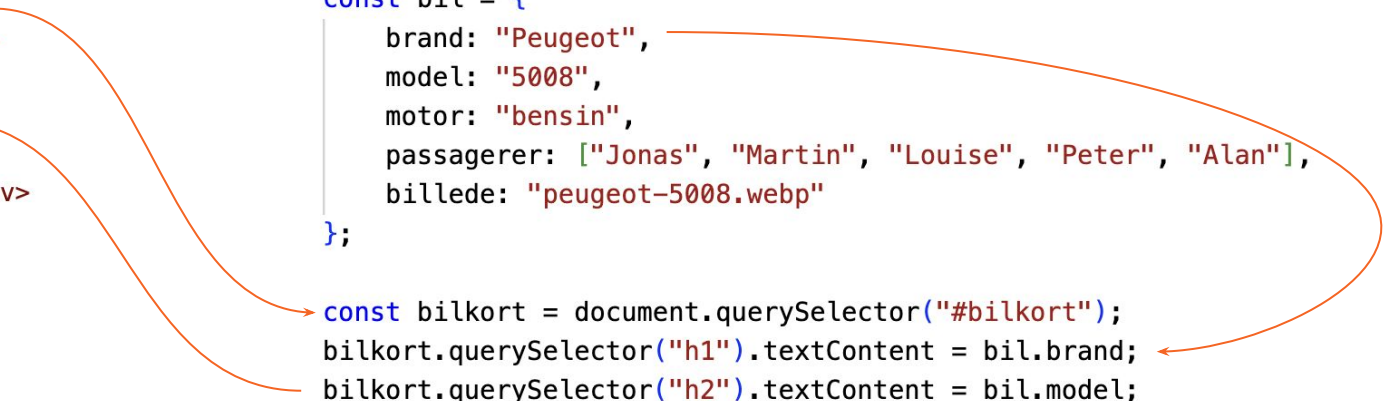
HTML

```
<article id="bilkort">
  <img src="" alt="">
  <h1>brand</h1>
  <h2>model</h2>
  <p>molor</p>
  <div>passagerer</div>
</article>
```

javascript

```
const bil = {
  brand: "Peugeot",
  model: "5008",
  motor: "bensin",
  passagerer: ["Jonas", "Martin", "Louise", "Peter", "Alan"],
  billede: "peugeot-5008.webp"
};
```

```
const bilkort = document.querySelector("#bilkort");
bilkort.querySelector("h1").textContent = bil.brand;
bilkort.querySelector("h2").textContent = bil.model;
bilkort.querySelector("p").textContent = bil.motor;
bilkort.querySelector("div").textContent = bil.passagerer;
bilkort.querySelector("img").src = bil.billede;
```



Arrays og objekter

- Vi ser ofte arrays af objekter, arrays i objekter, objekter i objekter osv.
 - Det er vigtigt at lære at navigere i disse strukturer!
 - Objekter tilgås ved hjælp af `.` (punktum)
 - Arrays tilgås ved hjælp af `[]` (kantede paranteser)
-

Eksempel på array af objekter

```
const biler = [  
  {  
    brand: "Peugeot",  
    model: "5008",  
    motor: "bensin",  
    passagerer: ["Jonas", "Martin", "Louise", "Peter", "Alan"],  
    billede: "peugeot-5008.webp"  
  },  
  {  
    brand: "Tesla",  
    model: "S",  
    motor: "el",  
    passagerer: ["Elon", "Jeff", "Marc"],  
    billede: "tesla-s.webp"  
  },  
  {  
    brand: "BMW",  
    model: "M340i",  
    motor: "diesel",  
    passagerer: ["Jytte", "Hans", "Asta", "Storm"],  
    billede: "bmw-M340i.webp"  
  },  
];
```

Hvordan finder vi...

Husk:

- Objekter tilgås ved hjælp af `.`
- Arrays tilgås ved hjælp af `[x]`

Gæt hvad disse linjer vil udskrive:

```
console.log(books[0].reviews[1].text);  
console.log(books[0].name);  
console.log(books[1].author);  
console.log(books[1].released.year);
```

```
const books = [  
  {  
    name: "JavaScript & jQuery",  
    author: "John Duckett",  
    released: {  
      year: 2014,  
      month: 8,  
    },  
    reviews: [  
      {  
        author: "Hansi",  
        rating: 9,  
        text: "Extremely relevant for any frontend developer"  
      },  
      {  
        author: "Poe",  
        rating: 1,  
        text: "Very boring story, totally lack of drama"  
      },  
    ],  
  },  
  {  
    name: "Peppa Pig - My little pocket book",  
    author: "Alvilda Anderson",  
    released: {  
      year: 2019,  
      month: 8,  
    },  
    reviews: [  
      {
```

Tip: Se slide
med biler for
inspiration

Øvelse 2

Lav et array med objekter

1. Åbn din js-fil fra sidste øvelse
 2. Lav nu et array med flere superhelte (eller bands, albums, drinks eller hvad du fandt på)
 3. Alle objekter i arrayet skal have samme data-struktur
 4. Gem filen til senere
-

Arrays af objekter, HTML <template> og forEach loops

vis gentagne data

Objekter og HTML template

Når vi skal vise indholdet af JS objekter i et **array** på en HTML-side er det smart at bruge et **<template>** element

I en **forEach** loop lave vi en kopi (clone) af en HTML skabelon (template) og ændrer dens indhold til indholdet fra hver enkelt objekt i **array**'et og **append**'er (tilføjer) den ændrede kopi til DOM'en

Lad se et eksempel med biler

MDN's forklaring på <template>

HTML elementet <template> er en mekanisme til at opbevare indhold på klientsiden, som ikke skal gengives, straks en side indlæses, men som efterfølgende kan instantieres under kørsel ved hjælp af JavaScript.

Tænk på en skabelon som et indholdsfragment, der gemmes til efterfølgende brug i dokumentet.

[MDN](#)

Eksempel template

```
<p>almindelig HTML</p>
...
<template id="minTemplate">
  <article>
    <header>
      <h1>HEADER</h1>
    </header>
    <section class="indhold">
      <h2>MELLEMRUBRIK</h2>
      <img src="" alt="BILLEDE">
      <p>TEKST</p>
    </section>
  </article>
</template>
...
```

Brug af template

Procedure:

1. Vælg indholdet af din <template>
 2. Lav en "klon" (en kopi) af skabelonen
 3. Skift indholdet i klonen
 4. Vælg "forælder"-elementet
 5. Tilføj (append) klonen til forælderen (og dermed DOM)
-

Kloning af template

```
//Step 1: Vælg <template>'s content
const templateElement = document.querySelector("#minTemplate").content;

//Step 2: lav en "klon"
const minKlon = templateElement.cloneNode(true);

//Step 3: Vælg klonens HTML elementer og ændr indholdet i disse
minKlon.querySelector("h1").textContent = "Hej Mor";
minKlon.querySelector("img").src = "mitBillede.wepp";
//...

//Step 4: Vælg et "parent" element
const parentElement = document.querySelector("body");

//Step 5: Tilføj (Append) klonen til DOM
parentElement.appendChild(minKlon);
```

Eksempel

```
const biler = [  
  {  
    brand: "Peugeot",  
    model: "5008",  
    motor: "bensin",  
    passagerer: ["Jonas", "Martin", "Louise", "Peter", "Alan"],  
    billede: "peugeot-5008.webp"  
  },  
  {  
    brand: "Tesla",  
    model: "S",  
    motor: "el",  
    passagerer: ["Elon", "Jeff", "Marc"],  
    billede: "tesla-s.webp"  
  },  
  {  
    brand: "BMW",  
    model: "M340i",  
    motor: "diesel",  
    passagerer: ["Jytte", "Hans", "Asta", "Storm"],  
    billede: "bmw-M340i.webp"  
  },  
];
```

Vi kloner skabelonen,
hver gang vi viser en
bil på listen (men vi
behøver kun en
"original" (const))

```
<template id="bilTemplate">  
  <article>  
    <img src="" alt="">  
    <h1>brand</h1>  
    <h2>model</h2>  
    <p>motor</p>  
    <div>passagerer</div>  
  </article>  
</template>
```

```
const temp = document.querySelector("#bilTemplate").content;  
  
biler.forEach(visBil);  
  
function visBil(bil) {  
  const klon = temp.cloneNode(true);  
  klon.querySelector("h1").textContent = bil.brand;  
  klon.querySelector("h2").textContent = bil.model;  
  klon.querySelector("p").textContent = bil.motor;  
  klon.querySelector("div").textContent = bil.passagerer;  
  klon.querySelector("img").src = bil.billede;  
  const beholder = document.querySelector("body");  
  beholder.appendChild(klon);  
}
```

Eksempel

```
const biler = [  
  {  
    brand: "Peugeot",  
    model: "5008",  
    motor: "bensin",  
    passagerer: ["Jonas", "Martin", "Louise", "Peter", "Alan"],  
    billede: "peugeot-5008.webp"  
  },  
  {  
    brand: "Tesla",  
    model: "S",  
    motor: "el",  
    passagerer: ["Elon", "Jeff", "Marc"],  
    billede: "tesla-s.webp"  
  },  
  {  
    brand: "BMW",  
    model: "M340i",  
    motor: "diesel",  
    passagerer: ["Jytte", "Hans", "Asta", "Storm"],  
    billede: "bmw-M340i.webp"  
  },  
];
```

Vi kloner skabelonen,
hver gang vi viser en
bil på listen (men vi
behøver kun en
"original" (const))

```
<template id="bilTemplate">  
  <article>  
    <img src="" alt="">  
    <h1>brand</h1>  
    <h2>model</h2>  
    <p>motor</p>  
    <div>passagerer</div>  
  </article>  
</template>
```

```
const temp = document.querySelector("#bilTemplate").content;  
  
biler.forEach(visBil);  
  
function visBil(bil) {  
  const klon = temp.cloneNode(true);  
  klon.querySelector("h1").textContent = bil.brand;  
  klon.querySelector("h2").textContent = bil.model;  
  klon.querySelector("p").textContent = bil.motor;  
  klon.querySelector("div").textContent = bil.passagerer;  
  klon.querySelector("img").src = bil.billede;  
  const beholder = document.querySelector("body");  
  beholder.appendChild(klon);  
}
```

Tip: Se forrige slide for inspiration ;-)

Øvelse 3

Vis dit array med objekter

1. Lav en ny HTML fil
2. Lav i denne et link til din js-fil fra sidste øvelse med array'et med flere superhelte
3. Lav en `<template>` i HTML-filen der passer til dit array med objekter, fx med en `<H2>` til heltenes navn, en `<div>` til fjender osv. Lav en `<article>` yderst
4. Lav også en "container" fx en `<section>` til listen med helte i din html-fil
5. Åbn js-filen og lav en funktion der kloner din template og skifter indholdet til data fra dine objekter
6. Kald funktionen i en `forEach` loop på dit array - noget i stil med:
`mineHelte.forEach(visHelt)`
7. Test at dine helte bliver vist på siden