

# Arrays og loops

# Ny datatype: **Array**

Datatyper: tekster, tal, boolean og nu også **array** - kendes på [ ]

Et **array** er en liste/tabel med flere elementer, som regel af samme type eller samme kategori som f.eks. "titler", "navne", "kæledyr", "postnumre" og lign.

Variabler med forskellige datatyper:

```
const minTekst = "en kort eller lang sætning"; // string " "  
const pris = 200; // number  
const lærerTeam = ["Alan", "Martin", "Klaus", "Kamilla"]; // array  
console.log(lærerTeam[0]); // returnerer "Alan"  
console.log(lærerTeam[3]); // returnerer "Kamilla"  
lærerTeam.length // giver antallet af elementer i array'et lærerTeam = 4
```

Tomt array:

```
let lærerTeam=[];
```

OBS!

Første  
index er 0.

Læs mere om Array:

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)

# Loop'e igennem et array: **forEach**

En "loop" kalder man den mekanisme hvor man gentager de samme kodelinjer for flere forskellige variabler - f.eks et arrays elementer.

**forEach** er én af de loop-syntaxer javascript byder på.

```
const lærerTeam = ["Alan", "Martin", "Klaus", "Kamilla"];  
lærerTeam.forEach(lærer => console.log(lærer));
```

**lærer** er et variabelnavn, som vi selv finder på,  
der repræsenterer hver enkelt værdi i arrayet.

HTML DOM querySelectorAll() Method, W3schools: [https://www.w3schools.com/jsref/met\\_document\\_queryselectorall.asp](https://www.w3schools.com/jsref/met_document_queryselectorall.asp)

JavaScript Array forEach() Method, W3schools: [https://www.w3schools.com/jsref/jsref\\_foreach.asp](https://www.w3schools.com/jsref/jsref_foreach.asp)

# forEach syntax

```
array.forEach(function(currentValue, index, arr))
```

```
const frugter = ["pærer", "æble", "banan", "appelsin"];  
frugter.forEach((frugt, index, arr) => console.log(frugt, index, arr));
```

```
pærer 0 ▶ (4) ["pærer", "æble", "banan", "appelsin"]
```

```
æble 1 ▶ (4) ["pærer", "æble", "banan", "appelsin"]
```

```
banan 2 ▶ (4) ["pærer", "æble", "banan", "appelsin"]
```

```
appelsin 3 ▶ (4) ["pærer", "æble", "banan", "appelsin"]
```

**Index** og **arr** er optional parametre. Index returnerer elementets indexnummer og arr returnerer hele arrayet

# Øvelse 9 - array

Lav en ny html-fil (eller gem en gammel) og kald den **array.html**

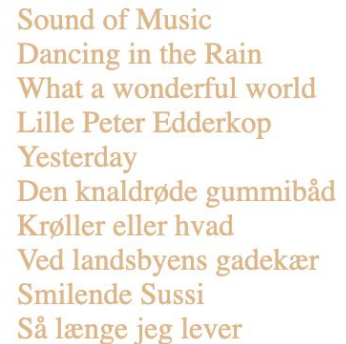
Lav et html-skellet med et liste element i <body> med id *display*  
`<ul id="display"></ul>`

Lav herunder et javascript:

Opret et **array** "playliste" med navnene på dine 10 yndlingstracks.

**Loop** arrayet igennem og få hvert tracknavn til at stå på hver sin linie på listen.

Commit og push til gitHub, når du er tilfreds med opgaven.



Sound of Music  
Dancing in the Rain  
What a wonderful world  
Lille Peter Edderkop  
Yesterday  
Den knaldrøde gummibåd  
Krøller eller hvad  
Ved landsbyens gadekær  
Smilende Sussi  
Så længe jeg lever

Vælg og loop igennem  
flere DOM-elementer

# querySelectorAll


Eksempel: Alle elementer på siden med class="red", skal forsvinde:

```
document.querySelector(".red").style.display="none";
```

Problem: kun det første element forsvinder!

For at få fat i dem alle:

```
const alle = document.querySelectorAll(".red");
```



Variablen **alle** bliver en  
liste (**array**) som vi kan  
loop'e igennem med  
**forEach**

For at "gøre noget" ved alle elementer, i eksemplet: få dem til at forsvinde:

```
alle.forEach (element => element.style.display = "none");
```

# querySelectorAll + event + this = interaksjon

Det bliver endnu mere interessant, når vi kombinerer tingene.

F.eks. kan vi tilføje en eventListener til alle elementer i et array, så de bliver interaktive:

```
<body>
  <div class="boks"></div>
  <div class="boks"></div>
  <div class="boks"></div>

  <script>
    const alleBokse = document.querySelectorAll(".boks");
    alleBokse.forEach(boks=>boks.addEventListener("click", aktiver));

    function aktiver(){
      this.classList.toggle("aktiv");
    }
  </script>
</body>
```

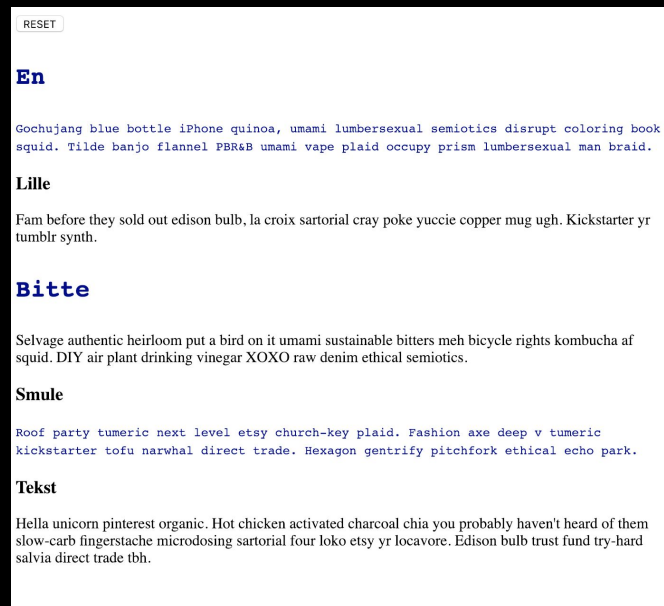
```
<style>
  .boks{
    width: 10vw;
    height: 10vw;
    margin: 1em;
    background-color: orange;
  }

  .aktiv{
    background-color: blue;
  }
</style>
```



# Øvelse 8: Eventlisteners på alle p-tags

1. Tag udgangspunkt i denne [klargjorte html-fil](#), og gem den som **08-alle.html**
2. I scriptet - lav en eventlistener der tjekker om DOM'en er indlæst. Sæt herefter en eventlistener på hvert p-tag: når der klikkes på en sætning, skal den styles med en klasse ex: "typo".
3. Lav en reset-knap øverst på siden.
4. Når der klikkes på knappen, skal class "typo" fjernes fra alle p-tags igen
5. Udvid programmet med en klasse, **typo2**, som styles alle overskrifter på samme måde - men med dobbelt størrelse.



## Øvelse 8b: Skift billedkategori ved klik

1. Gem **06-skiftBillede.html** fra øvelse 6 i en ny kopi med navnet **08-skiftKategori.html**.
2. Byg videre på dit script, så det andet billede skifter kategori i stedet, når der klikkes på det.
3. Lav først et array med de fem kategorier som der tilbydes fra placeimg.com:  
["animals", "arch", "people", "nature", "tech"]
4. Tilføj en eventlistener til det andet billede som kalder funktionen **skift()** ved click.
5. Lav funktionen **skift()** så den vælger en tilfældig kategori fra dit nye array og skifter både src og alt på billedet til denne kategori.

Hint:

```
function skift() {  
  const nummer = Math.round(Math.random() * 4);  
  billede.src = "https://placeimg.com/600/400/" + kategorier[nummer];  
  billede.alt = kategorier[nummer];  
};
```

# Opsamling

- Array
- Array index
- `querySelectorAll()`
- `forEach()`