JS CHEAT SHEET

2. semester

Lavet af: Victor Schweitz

ARRAYS

Arrays

Et array opbevarer flere værdier i én variabel, hvilket gør koden mere overskuelig og lettere at arbejde med. For eksempel kan man samle alle familiemedlemmer i et array i stedet for at oprette en variabel til hver enkelt person.

```
const arr = []
```

```
const arr =
['element', 'element']
```

Syntaks

Først skrives const efterfulgt af variabelnavnet. Dernæst følger et lighedstegn, hvorefter der kommer et sæt firkantede parenteser, der indikerer arrayet. Inde i disse er elementerne opdelt med et komma.

OBJEKTER

Objekter

Objekter grupperer information, der hører til den samme ting, som for eksempel køn, alder og navn på en person. I stedet for at oprette en variabel til hver oplysning, er det mere overskueligt at samle dem i et objekt.

```
const obj = {}
```

```
const obj = {
   gender: 'male',
   age: 24,
   name: 'Victor',
}
```

Syntaks

Objekter sættes altid til at være lig med en variabel. Syntaksen består af const, variabelnavn, et lighedstegn og tuborgparenteser, hvori key/value-parrene findes. Keyen efterfølges af et kolon, og derefter kommer værdien. Key/value-parrene adskilles af komma.

FETCH

Fetch

Fetch er en indbygget JavaScript-funktion, der henter data fra andre hjemmesider eller databaser via en URL. Den gør det muligt at opdatere indhold dynamisk uden at reloade hele siden.

```
fetch()
```

```
fetch('url')
```

Syntaks

Først skriver man fetch efterfulgt af et sæt parenteser. Inde i disse kommer et sæt citationstegn, hvori URL'en, som man ønsker at hente sin data fra, står. Hertil kommer formatering, som kan findes i eksemplet linket på sidste side.

QUERYSELECTORALL

.querySelectorAll()

.querySelectorAll() vælger alle elementer, der matcher den angivne selector, mens .querySelector() kun vælger det første. .querySelectorAll() returnerer en NodeList, der kan tilgås ligesom et array.

```
.querySelectorAll()
```

```
const elements =
document.querySelectorAll
(‘.element’)
```

Syntaks

Først skriver man const, som efterfølges af variablens navn. Derefter kommer et lighedstegn, som efterfølges af document.querySelectorAll(). Inde i parenteserne er et sæt citationstegn med selectoren.

FOREACH

.forEach()

.forEach() er en looptype, der kører for hvert element i et array eller NodeList, der returneres af .querySelectorAll(). .forEach() kan bruges både på et array og en NodeList.

```
.forEach()
```

```
const elements =
document.querySelectorAll
(‘.element’)

elements.forEach(sayHello)

function sayHello(parameter) {
    // Kode her
}
```

Syntaks

Først oprettes en variabel, der er lig med et array eller document.querySelectorAll(). Derefter skrives variabelnavnet efterfulgt af .forEach(), hvis parameter er navnet på funktionen, der skal kaldes. Til sidst deklareres denne funktion.

TEMPLATE

Template

Template er et HTML-element, der fungerer som en pladsholder for indhold, der genereres af JavaScript. Indholdet bliver først synligt, når det tilføjes til browseren via metoder som .cloneNode(), .append(), .appendChild() og .content-attributten.

```
.cloneNode(),
.append(),
.appendChild(),
.content
```

```
<template>
  // Indhold her
</template>
```

Syntaks

Syntaksen er rimeligt lige til, da det betstår af et åben- og lukke-element med indhold i - nøjagtigt ligesom de fleste andre elementer.
Genereringen af indholdet kan ses i eksemplet på sidste side.

URL-PARAMETRE

URL-parametre

URL-parametre bruges til at sende data mellem sider via URL'en, hvilket skaber et bedre flow på hjemmesiden. Webshops bruger denne teknik til at sende varer videre fra oversigtssiden til kurven og videre til betalingssiden.

```
URLSearchParams()
```

```
base-url.html?parameter=value
```

Syntaks

Først kommer basis-URL'en, som er den, der ender med ".html". Dernæst kommer et spørgsmålstegn, som indikerer, at det første parameter kommer, hvorefter navnet på det første parameter kommer efterfulgt af et lighedstegn og parameterets værdi.

ATTRIBUTTER

Attributter

Attributter bruges til at gøre et element mere specifikt. Dette kan være med attributter som class, id, href. Disse kan også sættes dynamisk via JavaScript med .setAttribute() og tages fat på med .getAttribute().

```
<div class="square" id="square">
```

```
// Tager fat på elementet
const div =
document.querySelector('div')

// Giver det klassen "square"
div.setAttribute('class', 'square')

// Tager fat på klasseattributten
div.getAttribute('class')
```

Syntaks

Både .setAttribute() og .getAttribute() anvendes på variablen. .setAttribute() tager to parametre: attributten og dennes værdi - begge i citationstegn og opdelt af komma. .getAttribute() tager kun ét parameter i citationstegn, som er den attribut, der blev givet med .setAttribute().

EKSEMPLER

Arrays

Objekter

Fetch

Selectors

Loop

Template

URL-parametre

Attributter