JS CHEAT SHEET

1. semester

Lavet af: Victor Schweitz

FUNKTIONER

Funktioner

En funktion er en måde at skrive sin kode på, så man kan genanvende det i stedet for at skrive de samme linjer kode alle steder, hvor den samme ting skal udføres. En funktion gør det også meget lettere at overskue sin kode.

```
myFunction()
```

```
function myFunction()
{
    // Kode her
}
```

Syntaks

En funktion starter med function efterfulgt af funktionsnavnet, som man helt selv bestemmer. Derefter kommer et sæt parenteser efterfulgt af et sæt tuborgparenteser, og inde i disse det kode, som funktionen skal udføre. En funktion gør ikke noget, før den bliver kaldt.

VARIABLE & KONSTANTE

Variable og konstante

En variabel er en reference til en værdi. De tre primære variabeltyper er *var*, *let* og *const*. *const*-variablens værdi kan ikke overskrives, når først den er deklareret. Det kan *var* og *let*, men det anbefales at bruge *let* til variable, hvis værdi skal overskrives.

```
const, let, var
```

```
  var myVariable = værdi
  let myVariable = værdi
const myVariable = værdi
```

Syntaks

Variable defineres ved først at skrive deres type (*const*, *let* (eller *var*)). Herefter kommer navnet på variablen, som fungerer som variablens reference. Efter dette kommer et lighedstegn, om efterfølges af, hvad variablens værdi er.

SELECTORS

Selectors

En selector bruges til at vælge et bestemt element i ens HTML. De tre primære er klasse, id og elementet selv. Nedenunder vises, hvordan de hver især tilgås.

```
klasse, id, element
```

```
Klasse:   .klasse-navn
Id:       #id_navn
Element:  elementnavn
```

Syntaks

For at tilgå et element via dettes klasse i CSS anvendes punktum efterfulgt af klassenavnet. Hvis et element skal tilgås via dets id, er det et hashtag efterfulgt af id'ets navn. Hvis man tilgår det via selve elementet, kommer der ingenting foran.

EVENTLISTENERS

Eventlisteners

En eventlistener er en måde at lytte til events på. Alle interaktioner med browseren er events (tastetryk, klik, input med videre). De bruges til at køre noget kode, når det angivne event udløses.

```
.addEventListener()
```

```
variabelNavn.addEventListener
(‘eventtype’, funktionsNavn)
```

Syntaks

Eventlistener tilføjes ved at bruge .addEventListener(). Denne sættes på variablen. I citationstegnene inde i parenteserne angives det ønskede event. Efter dette kommer et komma efterfulgt at funktionen, der skal køres, når eventet udløses.

KLASSEMANIPULATION

Klassemanipulation

Klassemanipulation er at tilføje klasser til og/eller fjerne klasser fra et element. De tre primære metoder, som I skal kende, er .add(), .remove() og .toggle().

```
.add(), .remove(), .toggle()
```

```
variabel.classList.add('klasse-navn')

variabel.classList.remove('klasse-navn')

variabel.classList.toggle('klasse-navn')
```

Syntaks

Syntaksen er magen til eventlistenernes, hvilket vil sige variabelnavn efterfulgt af .classList, som igen efterfølges punktum og metodenavn. Inde i parenteserne er et sæt citationstegn med den klasse, der skal tilføjes eller fjernes.

TIMERS

Timers

Timerfunktioner bruges til at få noget til at ske enten *efter* et givent interval er gået, eller *hver gang* et givent interval er gået. De mest anvendte er setTimeout og setInterval. setInterval kører, *hver gang* det angivne interval er gået, mens setTimeout kun kører *en gang*, *efter det angivne interval er gået*.

```
setInterval()/setTimeout()
```

```
// setInterval
setInterval(function() {
    // Kode her
}, interval)

// setTimeout
setTimeout(function() {
    // Kode her
}, interval)
```

Syntaks

Syntaksen er ens for setInterval og setTimeout. De starter med deres navn efterfulgt af parenteser, inde i hvilke der står function efterfulgt af parenteser og tuborgparenteser. Herinde står koden, der skal køres. Intervallet står som vist til venstre.

TILFÆLDIGHED

Tilfældighed

Tilfældighed bruges, når noget skal genereres tilfældigt. I kodning bruges det, når man vil lade computeren overtage noget af styringen af en bestemt handling. I JavaScript anvendes Math.random() til at generere en tilfældig værdi (som standard mellem 0 og 1).

```
0-1
```

```
// Tilfældigt decimaltal mellem 0 og 1
Math.random()

// Tilfældigt decimaltal mellem 1 og 10
Math.random() * 10

// Tilfældigt heltal mellem 1 og 10
Math.floor(Math.random() * 10) + 1
```

Syntaks

Man starter med at skrive Math, som herefter følges af .random og et sæt parenteser, så det bliver til Math.random(), hvilket giver et tal mellem 0 (*inkluderet*) og 1 (*ekskluderet*). Se eksemplerne til venstre.

OPERATORER

Operatorer

Bruges til at bestemme en værdi ud fra to eller flere tal. I programmering bruges de til at tjekke, om noget er sandt (for eksempel, om et tal er lig med et andet).

```
&&, ||, >, <
<=, >=, =, ==
```

```
&&                          (and)
||                          (or)
>                  (større end)
<                  (mindre end)
<=       (mindre end eller lig med)
>=       (større end eller lig med)
=                      (lig med)
==           (lig med samme værdi)
```

Syntaks

Der er ikke som sådan en syntaks for operatorerne alene, da disse bare er symboler, der udfører en bestemt handling, og disse derfor altid er brugt i sammenhæng med noget andet.

FORHOLD

Forhold

Forhold bruges til at tjekke, om et udsagn er sandt eller falsk. Forestil dig, at du kun har plads til ti varer i din indkøbskurv. For hver vare, du lægger i kurven, tjekker du, om den er fyldt. Dette er falsk, indtil den tiende vare lægges i, og kurven dermed er fyldt.

```
if/else
```

```
if(forhold)
{
  // Kode her
}
```

Syntaks

En if-sætning starter med if efterfulgt af parenteser indeholdende forholdet. Herefter kommer tuborgparenteserne, hvor koden, der skal køres, står.

KONSOLLEN

Konsollen

Konsollen er det primære fejlsøgningsværktøj. Her gives fejlmeddelelserne, og det er også her, man kan tjekke sin kode for at finde fejl. Genvejstasterne til at åbne konsollen i Chrome og Firefox på Mac og Windows ses til højre.

MacBook:

Chrome:	⌘ + Option + J
Firefox:	⌘ + Option + K

Windows:

Chrome:	CTRL + Shift + K
Firefox:	CTRL + Shift + K

```
console.log()
```

Syntaks

Der er ingen syntaks for konsollen, da der skrives almindeligt JavaScript i den. For at logge til konsollen skriver man console efterfulgt af .log. Herefter parenteser indeholdende det, man ønsker at logge.

LYD

Lyd

For at afspille lyd i browseren med JavaScript skal man have et audio-element med en src-attribut i sin HTML og hente det ind som en variabel i sin JavaScript. Man kan derefter arbejde med lyden ved at bruge metoder som .play(), .pause(), .muted, .volume, .duration og .currentTime.

Afspil lyd	.play()
Paus lyd	.pause()
Mute lyd	.muted
Volumen	.volume
Lydens længde	.duration
Tid i lydfilen	.currentTime

```
// Element
<audio src=””></audio>

// Variabel
const audio =
document.querySelector(‘audio’)

// Afspil lyd
audio.play()
```

Syntaks

Man skal have et audioelement i sin HTML for at kunne arbejde med lyd, som hentes ind som variabel i JavaScripten. Til venstre vises, hvordan man afspiller lyd.

EKSEMPLER

Funktioner

Variable

Selectors

Events

Klassemanipulation

Timers

Tilfældighed

Operatorer

Forhold

Konsollen

Lyd