Time based stuff

`setTimeout()`

Met `setTimeout()` kunnen we zorgen dat een bepaalde actie met enkele milliseconden uitgesteld wordt. Stel nu dat we een welkomstboodschap 5 seconden na het laden van het script willen tonen.

./js_essentials/time-based/setTimeout.js

```js
function sayHello() {
  alert("Hello, this script was loaded 5 seconds ago!")
}

const myTimeout = setTimeout(sayHello, 5000);
```

Het eerste argument is de te uitvoeren functie en het tweede argument is de te wachten tijd in milliseconden.

```js
myTimeout = setTimeout(function, milliseconds);
```

Je kan ook extra argumenten meegeven aan de te uitvoeren functie.

```js
setTimeout(functionRef, delay, param1, param2, /* … ,*/ paramN)
```

Zo kan je bijvoorbeeld een boodschap meegeven aan de te tonen melding.

./js_essentials/time-based/setTimeout-params.js

```js
function sayHello(name) {
  alert(`Hello ${name}, this script was loaded 5 seconds ago!`)
}

const myTimeout = setTimeout(sayHello, 5000, "Jannes");
```

Om te zorgen dat een functie in een bepaalde situatie toch niet wordt uitgevoerd, kan je `clearTimeout` gebruiken.

```js
clearTimeout(myTimeout)
```

> ⚠ **Opgelet**
>
> De effectieve delay kan langer zijn dan de opgegeven seconden, dit komt door oa. vertragingen in het programma of een kleine geforceerde delay om het programma te optimaliseren.

Asynchrone funtie

`setTimeout()` is een asynchrone functie, wat betekend dat deze functie de overige code niet blokkeerd. Je kan dus `setTimeout()` niet gebruiken om een pauze in de codete implementeren.

```js
setTimeout(() => {console.log("Deze code wordt uigevoerd achter 5 seconden")}, 5000);
setTimeout(() => {console.log("Deze code wordt uigevoerd achter 1 seconde")}, 1000);

// Output:

// Deze code wordt uigevoerd achter 1 seconde
// Deze code wordt uigevoerd achter 5 seconden
```

`setInterval()`

Om een code uit te voeren met een bepaald interval (eg. 3 seconden), gebruiken we `setInterval()`. Deze methode heeft een uniek ID terug, zodat we dit kunnen verwijderen met `clearInterval()`.

./js_essentials/time-based/setTimeout-params.js

```js
function sayHello(name) {
  alert(`Hello ${name}, this script was loaded 5 seconds ago!`)
}

const myTimeout = setTimeout(sayHello, 5000, "Jannes");
```

```js
myInterval = setInterval(function, milliseconds);
```

Je kan ook extra argumenten meegeven aan de te uitvoeren functie.

```js
setInterval(functionRef, delay, param1, param2, /* … ,*/ paramN)
```

Zo kan je bijvoorbeeld een boodschap meegeven aan de te tonen melding.

./js_essentials/time-based/setInterval-params.js

```js
function sayHello(name) {
  alert(`Hello ${name}, this script will execute every 3 seconds!`)
}

const myInterval = setInterval(sayHello, 3000, "programmer");
```

`requestAnimationFrame()`

Met de methode `requestAnimationFrame()` kan je aan de browser vertellen dat je een animatie wilt uitvoeren en vraagt aan de browser om een specifieke functie op te roepen die de animatie bijwerkt voor de volgende repaint. De methode neemt als argument een callback die moet worden aangeroepen voor de repaint.

./js_essentials/time-based/requestAnimationFrame.js

```js
// Setting the start point for animation
let start = null;
let element = document.getElementById('forward');

function startAnim(timestamp) {

    // Timestamp is an integer that represents the number
    // of seconds elapsed since January 1 1970.
    if (!start) start = timestamp;

    // Setting the difference between timestamp
    // and the set start point as our progress
    var progress = timestamp - start;

    // Moving our div element
```

```
    element.style.transform =
    `translateX(${Math.min(progress / 10, 700)}px)`;
    window.requestAnimationFrame(startAnim);
}

window.requestAnimationFrame(startAnim);

```

Deze code zorgt ervoor dat de tekst van links naar rechts beweegt over het scherm, merk op dat er geen controle is of dit beeld nog zichtbaar is. Dus deze beweging gaat blijven doorgaan.

We kunnen deze beweging (of een animatie in het algemeen) stoppen met `cancelAnimationFrame()`, hiervoor moeten we dan wel `requestAnimationFrame()` toekennen aan een ID.

```js
let myReq = requestAnimationFrame(func);
cancelAnimationFrame(myReq);
```

Dit is dus niet correct:

```js
let myReq = requestAnimationFrame(func);
cancelAnimationFrame(func);
```

← Geïndexeerde collecties

Coding guidelines →

Laatst bijgewerkt: 8/9/2022 13:39:31