

# Plusopdracht

*In deze les ga je leren wat een loop is. Dan gaan we kijken naar het HTML DOM en hoe we met JS de HTML kunnen aanpassen. In deze les gaan we langzaam, stapje voor stapje, een soort memory spelletje maken.*

Bestudeer de volgende JS code.

```
for (i = 0; i < 5; i++) {  
  text += "The number is " + i + "<br>";  
}
```

Probeer deze code uit.

In de (Engelstalige) Youtube movie wordt de loop ook nog een keer in detail uitgelgd:

<https://www.youtube.com/watch?v=s9wW2PpJsmQ>

In deze les gebruiken ze het commando *let*. Het JS comamndo *let* is bijna hetzelfde als het commando *var*. Het verschil is op dit moment nog niet zo belangrijk; met beide initialiseer je een variabele.

## Opdracht 1

Maak nu een loop die de getallen 10 tot en met 25 afdrukt.

## Opdracht 2

Vanuit de HTML lessen weet je waarschijnlijk nog wel hoe je een tabel maakt.

Maak een tabel met een kolom en 10 regels, zet in deze regels de getallen 1 t/m 10 en maak deze code met behulp van een loop.

## Opdracht 3

Maak nu een tabel met twee kolommen en zet hierin de getallen 1 tot en met 20 op de volgende manier:

1	2
3	4
5	6
7	8
etc.	...

Maak deze code en gebruik hiervoor een loop.

## Opdracht 4 (extra punten)

Gebruik nu twee loops in elkaar en maak de volgende 10 bij 10 tabel af (het voorbeeld is een 6x6 tabel afgedrukt en je ziet op elke regel telkens de tafel van 1, 2, 3, ...etc).

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36
etc.					

Maak een JS scrip dat deze tabel automatisch genereerd.

## Opdracht 5

Kijk en bestudeer: [https://www.w3schools.com/howto/howto\\_js\\_toggle\\_hide\\_show.asp](https://www.w3schools.com/howto/howto_js_toggle_hide_show.asp)

Met de functie die he daar maakt, maar je een HTML element onzichtbaar. Je kunt ook de inhoud van een element veranderen:

```
function myFunction(myID) {
  var x = document.getElementById(myID);
  if (x.innerHTML=="_") {
    x.innerHTML=11;
  } else {
```

```
    x.innerHTML=" _";  
  }  
}
```

Je kunt een Javascript aanroepen vanuit een HTML link. Weet je nog de standaard code van een link

```
<a href="www.gogole.com">link</a>
```

Nu kun je de link aanpassen zodat je JavaScript functie wordt aangeroepen:

```
<a id="11" href="javascript:myFunction(11);">_</a>
```

Test het uit. Plaats de link in de body van de code en zet het JavaScript in de head sectie.

Zie je wat er gebeurt?

## Opdracht 6

Maak nu een tabel (met de HTML dus niet genereren vanuit JavaScript). en zet in de table in elke TD een link met een nummer zoals en een ID zoals hierboven in het voorbeeld. Zorg ervoor dat je alle nummers in de tabel kunt wegclicken en ook weer kunt aanclicken.

We kunnen nu een soort memory spel maken, de eerste stap is gezet.

Stuur de hele HTML pagina met HTML code en script op.

## Opdracht 7

Voor de volgende stap heb je kennis nodig van twee zaken:

1. Arrays: [https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)
2. Random nummers: [https://www.w3schools.com/js/js\\_random.asp](https://www.w3schools.com/js/js_random.asp)

Zorg dat je deze twee concepten begrijpt.

Maak nu een array met zoveel elementen als je memory-spel groot is en zorg ervoor dat je memory-spel een even aantal vakjes heeft. Stel je hebt een spel van 4 x 4, dan heb je 16 kaartjes. Je maakt dan een array met 16 elementen. Je kiest dan 16 elementen maar je zet deze in paren neer en zorgt ervoor dat alle items even lang (aantal karakter/letters) zijn. Dus bijvoorbeeld:

```
myArray = [ 'rood', 'rood' , 'peer', 'peer', 'drol', 'drol' ];
```

Nu komt de truuk en de opdracht. Maak dit array en kiest dan twee willekeurige getallen tussen 0 en de lengte van je array-1. Dus in het voorbeeld van het array met 16 elementen kies je twee getallen tussen 0 en 15 (minimaal 0 en maximaal 15). Stel je noemt deze getallen random1 en random2. Dan gebruik je deze twee nummers om myArray[random1] en myArray[random2] om te draaien. Dus myArray[random1] krijgt de waarde van myArray[random2] en andersom, myArray[random2] krijgt de waarde van myArray[random1]. Dus stel random1=0 en random2=4 dan zou het bovenstaande array er als volgt uitzien:

```
myArray = [ 'drol', 'rood', 'peer', 'peer', 'rood', 'drol'];
```

Zie je dat array element 0 en 4 zijn omgewisseld?

Ok, als we dit nog een paar keer doen dan schudden we het array als het ware door elkaar. Gebruik nu een loop om deze wissel-truuk 20 keer te doen. We hebben dan het hele array door elkaar geschud.

Het kan helpen om ervoor te zorgen dat random1 en random2 niet hetzelfde zijn want als dat zo is dan wisselt er niets.

Maak jouw array, en 'schud' het array door elkaar zoals hierboven is beschreven en druk het af om te controleren of het allemaal werkt. Als je het helemaal goed wilt maken dan maak je een functie 'schudden' die als parameter het array mee krijgt en die het door elkaar geschudde array returnd.

--

---

Revision #12

Created 25 November 2019 21:04:53 by Admin

Updated 12 December 2019 10:33:34 by Admin