

# JavaScript 1

JavaScript introductie 1ste leerjaar (gegeven aan alle 1ste klassen in de 2de periode 2019).

Standaard: Variabelen, if-then-else, true/false, functions, arrays

Plus: loops en DOM

- [JavaScript](#)
- [Les 1, Variabelen](#)
- [Les 2, Operatoren](#)
- [Deel 1, Oefenen voor toets](#)
- [Deel 1, Samenvatting](#)
- [Les 3, Vergelijkingen en if statement](#)
- [Les 4, Functions](#)
- [Les 5, Arrays](#)
- [Plusopdracht](#)

# JavaScript

*Intro - wat is JavaScript?*

Powerpoint.

## Wat is JavaScript

In 1995 Netscape maakte LiveScript, een jaar later JavaScript (gebaseerd op de naam Java; Java en JavaScript is heel wat anders).

Er zijn veel **back-end** talen: Java, PHP, Python, c#, node.js (ook JavaScript gebaseerd).

**Front-end** zijn er veel minder: JavaScript en is ingebouwd in alle moderne browsers en vanaf.

Er zijn veel libraries en frameworks gebaseerd op JavaScript zoals jQuery, AngularJS, React, Vue.js

JavaScript is Object oriented.

HTML is het skelet, CSS is de aankleding en JavaScript maakt alles dynamisch. HTML, CSS en JavaScript zijn allen front-end technologieën.

## Set-up

Web browser (Chrome/Firefox) en editor (Brackets, Atom, Sublime Text, ...).

Download Google Chrome Browser	<a href="https://www.google.com/chrome/index.html">https://www.google.com/chrome/index.html</a>
Download Brackets Code Editor	<a href="http://brackets.io/">http://brackets.io/</a>

# Les 1, Variabelen

*In deze les leren we wat een variabele is en hoe we deze in JavaScript kunnen gebruiken.*

*We gaan ook leren hoe je JavaScript kunt toevoegen aan je web pagina, dit kan op drie verschillende manieren.*

## Wat is een variabele?

In elke programmeertaal zijn variabelen erg belangrijk. In JavaScript dus ook. Misschien ken je ze al van wiskunde, maar in deze les wordt uitgelegd wat het in JavaScript zijn en hoe ze werken. Maar wat is zo'n variabele nou precies?

Een variabele is een plekje in het geheugen van de computer, en met de naam van die variabele (dus bijvoorbeeld x of y) kun je dan bij het geheugen op dat plekje. Ergens in het geheugen van de computer staat dus een waarde opgeslagen, en die kan jij dan met JavaScript veranderen of gebruiken.

De naam van dit geheugenplekje moet uniek (= mag niet al bestaan) zijn, want als je bijvoorbeeld twee plekjes hebt die beide X heten dan weet de computer niet welk plekje je bedoeld. Verder zijn er in JavaScript nog nog een aantal kenmerken aan een variabele:

1. De naam moet beginnen met een kleine- of hoofdletter of een underscore (\_).
2. De naam mag daarna letters (hoofdletters of kleine letters), underscores(\_) en cijfers bevatten.
3. De naam mag *niet* een naam van een JavaScript commando zijn.

Variabele namen zijn case sensitive. Dat betekent dat X en x verschillende variabelen zijn.

Hoe gebruik van een variable in JavaScript gaat as volgt:

```
var nummer;  
nummer = 42;
```

Hier staat eigenlijk: (1) maak een variable, een plekje in het geheugen, en noem deze *nummer*, (2) stop dan 42 in dit stukje geheugen. Dus de waarde van nummer wordt 42.

Dit kan ook korter:

```
var x = 42;
```

# Code

JavaScript is een ClientSide taal en kan tussen de HTML code worden geplaatst. We maken nu een sprongetje en gaan de volgende code op onze laptop uitproberen.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="nl" lang="nl">
<head>
  <title>Variabelen (hoofdstuk 4)</title>
  <script type="text/javascript">
    var x = 42;
    document.write(x);    // Uitvoer: 42
    document.write("<br />");
    document.write('x'); // Uitvoer: x (niet 42!)
    document.write("<br />");
    x = 11;
    document.write(x);    // Uitvoer 11
  </script>
</head>
<body>
</body>
</html>
```

Met de `document.write` wordt er iets naar het browser scherm geschreven. Je zult zien dat overal waar `x` staat, dit nu wordt vervangen door 42. Als iets echter tussen `""` of `"` staat dan ziet JavaScript dit niet als een variabele. Op regel 9 wordt de `x` dus niet vervangen door 42.

## <script>

Verder zie je ook dat de JavaScript tussen `<script type="text/javascript">` en `</script>` tags staan. Meestal zet je de JavaScript code ergens in de `<head>` sectie van de html pagina.

## JS File

Nog beter is de javascript in een aparte file te plaatsen. Dat houdt het lekker overzichtelijk. Dat doen we als volgt; plaats de onderstaande code in de head sectie van je html pagina en maak

vervolgens ene nieuwe file die je myScript.js noemt. Plaats daar de JavaScript code nu in.

```
<script src="myScript.js"></script>
```

Je kunt JavaScript niet alleen via een file laden maar ook via een URL: probeer het volgende maar eens:

```
<script src="https://www.yr.no/place/Netherlands/North_Holland/Amsterdam/external_box_stripe.js"></script>
```

## Prompt

Variabelen blijken pas echt nuttig te zijn wanneer je invoer van de gebruiker wil vragen. Dan weet je namelijk niet van tevoren wat er ingevoerd gaat worden.

Plaats het onderstaande stukje JavaScript code in een HTML pagina zoals je dat eerder ook hebt gedaan. Plaats dus de code in de <head> sectie en plaats de code tussen de juiste tags <script type="text/javascript"> en </script>.

Voer het uit en kijk wat er gebeurt.

```
var antwoord;  
antwoord = prompt("Wat is jouw naam?", "");  
document.write("Jouw naam is: ");  
document.write(antwoord);
```

Je ziet dat er nu met het prompt commando naar een naam wordt gevraagd. Deze wordt in een variabele opgeslagen en vervolgens afgedrukt.

Kun je je naam nu ook bold afdrukken? Tip: je hebt doorvoor de <b> en </b> tags nodig.

## Opgaven

*De volgende les gaan we het huiswerk bespreken. Maar daarom het huiswerk in een Word document zodat we dit samen in de klas kunnen nakijken.*

### Opgave 1

Geef van elke regel aan of de variabele goed of fout is.

```
var x;  
var X;
```

```
var naam;  
var 2eNaam;  
var tweede_naam;  
var tweedeNaam;  
var naam2;  
var _Naam;  
var var;  
var $x;  
var function;
```

## Opgave 2

Is regel 1 en 2 hetzelfde?

## Opgave 3

Maak een JavaScript dat aan de gebruiker vraagt welke dag het is en die dan bijvoorbeeld op het scherm afdrukt:

Vandaag is het maandag.

## Opgave 4

Maak de opgaven op:

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_variables1](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables1)

--

# Les 2, Operatoren

In deze les leren we wat een operator is

## Variabele

We hebben in de vorige les gezien hoe we een variabele gebruiken.

Met het commando

```
var myVar;
```

maken we een variable met de naam *myVar*. We noemen dit het *declareren* van de variabele.

## Operators op twee variabele

Een operator is een functie die je op (meestal) twee variabele kan uitvoeren. De meest eenvoudige operatoren zijn optellen, aftrekken, vermenigvuldigen, en delen. Bekijk het volgende stukje code:

```
var a = 10;  
var b = 12;  
var c = a + b;  
document.write(c);
```

Je declareert hier de variabele a, b en c. De variabele a krijgt de waarde 10, de variabele b krijgt de waarde 12 en c krijgt de waarde a + b. De variabele c wordt dus 12 en deze variabele wordt afgedrukt.

## Opdracht

Voer de code uit en verander het + teken stuk voor stuk in de operator `- , * , /`

## Operators op één variabele

Je kunt ook een operator op één variable uitvoeren, bijvoorbeeld:

Operator is zelfde	als
--------------------	-----

myVar++	myVar = myVar + 1
myVar--	myVar = myVar - 1
myVar+=2	myVar = myVar + 2
myVar-=b	myVar = myVar - b
document.write(a++)	document.write(a); a=a+1;
document.write(++a)	a=a+1; document.write(a);

## Modulo %

Ten slotte heb je nog een operator %, deze berekent de rest van een deling van twee variabelen. Bijvoorbeeld  $14\%5$  resulteert in 4 omdat 14 gedeeld door 5, 2 is en er dan 4 over blijven.

Nog wat voorbeelden:

Modulo	is gelijk aan
$23\%5$	3
$101\%10$	1
$9\%3$	0
$50\%7$	1

## Strings

Tot nu toe hebben we het gehad over operatoren op variabelen die een getal bevatten. Variabelen kunnen ook een string bevatten. Een string is een "ketting" van karakters, bijvoorbeeld: "Dit is een string", "Hello World!", "!^&YDJL:-(". Dit zijn dus drie strings. Een string staat altijd tussen "" of " (dubbele quotes of enkele quotes).

Strings kun je ook optellen met een + teken. Wat doet dat dan?

```
var a = "Hel";
var b = "lo";
var c = a + b;
document.write(c);
```

Probeer de code en zie wat het + teken met twee strings doet.

## Data Types



JavaScript -en ook andere computertalen- kennen datatypes. Een data type geeft het soort variabele aan. Tot nu toe hebben we twee soorten variabelen behandeld: getallen en strings. Later zullen we nog meer data types leren kennen. We hebben gezien dat we operatoren hebben voor verschillende data types. Zo konden we twee getallen optellen met de + operator en we konden twee strings aan elkaar plakken met de plus-operator.

Wat gebeurt er nu als we een string en een getal bij elkaar willen optellen, bijvoorbeeld:

```
var x = 16 + "Volvo";
```

We tellen 16 bij de string "Volvo" op. Dat slaat op zich nergens op. JavaScript lost dit op door te denken dat de 16 ook een string is. De opdracht wordt dus eigenlijk vertaald naar:

```
var x = "16" + "Volvo";
```

# Opdrachten

## Opdracht 1

Maak een scriptje waarin je drie variabele declareert en een waarde geeft. Verander dan in het script de waarde van alle drie de variabele.

## Opdracht 2

Wat is de waarde van b aan het einde van dit script?

```
var b = 4;  
var c = 5;  
c++;  
b=c;
```

## Opdracht 3

Wat is de waarde van a aan het einde van het script?

```
var a = 3;  
var b = 4;  
var c = 5;  
a += b+c;
```

## Opdracht 4

Wat wordt er in het volgende script afgedrukt?

```
var a = "1";  
var b = "2";  
document.write( a + b );
```

## Opdracht 5

Wat doet de volgende code?

```
var str = "Hello ";  
str += "world!";  
document.write(str);
```

## Opdracht 6

Wat doet de volgende code?

```
var a=10;  
var b='5';  
document.write(a+b);
```

## Opdracht 7

Maak een JavaScriptje met twee variabele: voorNaam en achterNaam. Initialiseer deze variabele en geef ze jouw eigen voor- en achternaam. Log (console.log) vervolgens jouw naam naar het console en controleer of je naam daar verschijnt. Zorg er voor dat je een spatie tussen jouw voor- en achternaam afdruckt.

## Opdracht 8

Maak een scriptje dat aan de gebruiker eerst zijn voornaam en daarna zijn achternaam vraagt. Druk vervolgens op een web pagina de boodschap "Welkom .....!!"af. Op de plaats van de puntjes moet dan jouw voor- en achternaam komen te staan. Let op de spatie en de uitroepteken aan het eind!

## Opdracht 9

Maak een scriptje dat aan de gebruiker twee getallen vraagt. Stel dat de gebruiker 12 en 10 invoert dan druk je op de web pagina af:  $12 \times 10 = 120$ .

## Opdracht 10

De Amerikaanse dollar is ongeveer 0.90 Euro waard. Dus als iets 100 US Dollar kost dan kost dat  $0.9 \times 100 = 90$  euro. Maak nu een scriptje die de prijs vraag in Amerikaanse dollars,. Stel je voert in dat iets 23 dollar kost. Dan programmeer jij een scriptje dat op de webpagina het volgende afdruckt: 23 Amerikaanse Dollar is 20.7 Euro

--

# Deel 1, Oefenen voor toets

## *Herhaling en oefening voor test*

Probeer eerst te voorspellen wat de code doet. Daarna kan je het proberen door de code uit te laten voeren op je laptop. Voor de toets kun je de antwoorden niet zelf controleren dus zorg ervoor dat je de antwoorden ook echt begrijpt.

## Opgave 1

Wat doet deze code?

```
var str = "Hello" + ' ' + "world!";  
document.write(str);
```

## Opgave 2

Wat wordt er afgedrukt?

```
var a = 1;  
var b = 11;  
a++;  
document.write(b%a);
```

## Opgave 3

Wat wordt er afgedrukt?

```
var a = 10;  
var b = 11;  
var c = a++;  
document.write( b - a );
```

## Opgave 4

Wat wordt er afgedrukt (let op er worden twee getallen afgedrukt)?

```
var a = 6;  
var b = a - 1;
```

```
document.write( a*b-2 );  
document.write( a*(b-2) );
```

## Opgave 5

Wat wordt er afgedrukt?

```
var a = 6;  
var b = 2;  
a+ = 2;  
b+ = 2;  
a = a + a;  
document.write( a + b );
```

## Opgave 6

Wat wordt er afgedrukt?

```
var a = "1";  
var b = "2";  
document.write( a + b + a + a );
```

## Opgave 7

Wat wordt er afgedrukt?

```
var piet=101;  
var jan=102;  
var chor=103;  
jan = chor + piet  
document.write( chor - piet );
```

## Opgave 8

Wat wordt er afgedrukt?

```
var auto = "Tesla";  
var prijs = 60000;  
document.write( auto + prijs );
```

## Opgave 9

Wat wordt er afgedrukt?

```
var getal_1=1;
var getal_2=1;
getal_1++;
getal_2 = getal_1 + getal_2;
document.write( ( getal_2-getal_1 ) *( getal2+3 ) );
```

## Opgave 10

Wat wordt er afgedrukt?

```
var getal_1=10;
var getal_2=getal_1;
getal_1--;
getal_2+=1;
document.write( getal_2 % getal_1 );
```

## Opgave 11

Wat doet de volgende code?

```
var a = 10;
var b = '5';

document.write(a * b);
document.write(a + b);
```

Let uit wat er gebeurt.

## Opgave 12

Maak een script dat vraagt hoeveel bier je hebt verkocht. Vraag de gebruiker dan de prijs van één biertje. Laat het scrpt dan uitrekenen hoeveel geld er in de kassa moet zitten. Dit bereken je dus door het aantal verkochte biertjes maal de prijs te berkenen. Je mag er vanuit gaan dat de ingevoerde getallen geldige getallen zijn.

Tip: maak gebruik van het [prompt](#) commando.

--

# Deel 1, Samenvatting

We hebben geleerd:

## HTML, CSS, JavaScript

- dat JavaScript een WebPagina dynamisch kan maken: HTML is het skelet, CSS is de aankleding en JavaScript is de 'beweging'.

## Front-end

- dat JavaScript een front-end taal is. Dat betekent dat JavaScript wordt uitgevoerd op de front-end (front-end is jouw computer waar de web browser op draait en back-end is de web server). JavaScript is de enige script taal die door alle browsers wordt ondersteund. De browser interpreteert de Javascript en voert die zelfstandig uit.

## Variabele

- dat een variabele een verwijzing is naar een plekje in computergeheugen. In dit plekje staat een getal of een string (woord of zin). Variabele hebben een naam, die bestaat uit één woord. Om variabele leesbaar en berijpbaar te houden gebruiken we de camelCase standaard, bijvoorbeeld: *cijferEngels* of *studentenNummer*. Niet alle karakters mogen voorkomen in een variabele en variabelen zijn case sensitive in JavaScript.

## Strings en getallen

- dat getallen en strings kunnen in JavaScript door elkaar worden gebruikt en JavaScript probeert er het beste van te maken. Als je bijvoorbeeld een string bij een getal optelt dan doet JavaScript net alsof het twee strings bij elkaar 'optelt'.

## Operatoren

- dat er een aantal operatoren is die je op getallen en soms op strings kunt toepassen; de belangrijkste zijn +, -, \*, /, %. Deze zijn toepasbaar op twee getallen of strings, maar er is ook een aantal operatoren die je op één getal of string kunt toepassen, de belangrijkste zijn ++, --, +=, -=.

## Modulo

- dat er één operator is die we niet kennen vanuit de wiskunde, dit is de modulo, % operator. Dit is de operator die de rest van een deling uitrekent.

## JavaScript plaatsing

- hoe je javascript code kunt toevoegen aan je HTML pagina. Dat kan op drie manieren: inline, via een aparte file of via een andere website. De javascript verwijzing staat meestal in de header sectie van je HTML pagina, maar we zullen later zien dat dat niet altijd zo hoeft te zijn.

## Commentaar in JS

- In JavaScript plaats je commentaar met // of tussen /\* ... \*/ Commentaar is belangrijk zodat anderen of jijzelf later nog begrijpt wat de code doet.

--



# Les 3, Vergelijkingen en if statement

*In deze les gaan we leren wat conditionele statements zijn dit zijn statements met een if-then-else.*

## Vergelijkingen

Een vergelijking is wanneer je twee waarden vergelijkt: is de één groter dan de ander, of juist kleiner. Voor vergelijkingen kun je verschillende vergelijgings-operatoren gebruiken. hieronder staan de meest gebruikte.

Vergelijgings operator	Betekenis
==	waarde is gelijk
!=	waarde is ongelijk
>	waarde is groter dan
>=	waarde is groter dan of gelijk
<	waarde is kleiner dan
<=	waarde is kleiner dan of gelijk

Als de vergelijking klopt is die true (waar) als die niet klopt dan is die false (niet waar).

## Opdracht 1

Zet nu zelf true of false achter de vergelijkingen

Opdracht	Vergelijking	true of false?
a	(12 == 13)	
b	(12 > 13)	
c	(12 < 13)	
d	(13 >= 13)	
e	(12 <= 13)	
f	( 'a' != 'b' )	

Bij een if statement volgt een vergelijking als die true is dan wordt er uitgevoerd wat er achter de if staat. Als de vergelijking achter de if niet waar is dan wordt de code na de if niet uitgevoerd maar wordt eventueel wel de code die bij de else staat wordt uitgevoerd.

Dus:

```
// Voorbeeld 1
if (vergelijking) {
  // dit wordt uitgevoerd als de vergelijking true is
}

// Voorbeeld 2
if (vergelijking) {
  // dit wordt uitgevoerd als de vergelijking true is
} else {
  // dit wordt uitgevoerd als de vergelijking false is
}

// Probeer het nu zelf:
if ( 13 < 14 ) {
  document.write('De vergelijking was true');
} else {
  document.write('De vergelijking was false');
}
```

## Opdracht 2

Geef van elk van de volgende vergelijkingen aan of er true of false uit komt.

```
// opgave a
if ( 13!=12 ) { .... // true or false?

// opgave a
if ( 13>=12 ) { .... // true or false?

// opgave b
if ( 13!=12 && 13==12 ) { .... // true or false?

// opgave c
```

```
if ( 13!=12 || 13==12 ) { .... // true or false?

// opgave d
if ( 13>=12 && 13<=12 && 0<1 ) { .... // true or false?

// opgave e
if ( 13<=12 || 13<=12 || 0!=1 ) { .... // true or false?

// opgave f
if ( 13<=12 || 13<=12 || 0<=1 ) { .... // true or false?

// opgave g
if ( 13==12 || 13<=12 || 0<=1 ) { .... // true or false?
```

## Opdracht 3

En nu een ander voorbeeld. Vervang de .... in de inderstaande code met een vergelijking die true is wanneer het geslacht van het kind man is.

```
var kind="Alexis";
var geslacht="man";
if ( .... ) {
    document.write( kind + ' is mijn zoon');
} else {
    document.write( kind + ' is mijn dochter');
}
```

Een if statement kan nog complexer.

```
var leeftijd = 22;

if (leeftijd <= 6 ) {
    document.write('Je bent een kleuter');
} else if ( leeftijd <= 12 ) {
    document.write('Je bent een scholier');
} else if ( leeftijd <= 18 )
    document.write('Je bent nog steeds leerplichtig');
} else {
    document.write('Je bent bijna volwassen');
```

```
}
```

## Opdracht 4

Pas de bovenstaande code aan zodat wanneer je 21 of ouder bent je afdruckt 'Je bent volwassen!'.

## Opdracht 5

Je hebt drie variabelen:

```
var cijferVanJohn = 7;  
var cijferVanJane = 5;  
var voldoendeVanaf = 6;
```

Je kunt de cijfers variëren en je moet nu een stukje code maken dat bepaald wie er is geslaagd. Afhankelijk van het resultaat wordt er één van de volgende regesl afgedrukt:

1. Eén van de studenten is geslaagd
2. Beide studenten zijn geslaagd
3. Geen van de studenten is geslaagd

In dit voorbeeld (met de cijfers zoals die zijn gegeven), zou regel 1 dus worden afgedrukt. Maar verander nu het cijfer van Jane naar een 9 en kijk of dan boodschap 2 wordt afgedrukt. Ten slotte moet je ook nog testen of boodschap 3 wordt afgedrukt.

## Opdracht 6

Maak nu een scripje dat twee vragen aan de gebruiker stelt. Vraag 1 is welk jaar bent u geboren? Vraag 2 is bent u dit jaar jarig geweest. Aan de hand van de twee antwoorden druk je nu af hoe oud de persoon is.

- Voorbeeld ik ben geboren in 2002 en ik ben 31 december jarig dan ben ik dus 16.  
De computer vraagt eerst: "Wanneer bent u geboren?" en de gebruiker voert 2002 in.  
Dan vraagt de computer: "Bent u al jarig geweest (ja/nee)?" en de gebruiker voert *ja* in.  
Op het scherm verschijnt nu: "u bent 16 jaar oud".
- Voorbeeld ik ben geboren in 2000 en ben in juni harig dan ben ik dus 18.  
De computer vraagt eerst: "Wanneer bent u geboren?" en de gebruiker voert 2000 in.

Dan vraagt de computer: "Bent u al jarig geweest (ja/nee)?" en de gebruiker voert *nee* in.

Op het scherm verschijnt nu: "u bent 18 jaar oud".

## Opdracht 7

In plaats van te vragen of iemand al jarig is geweest met het prompt commando kan je ook het commando `confirm()` gebruiken. Tussen haakjes zet je de vraag van de gebruiker en de waarde die terugkomt is `true` or `false`. De waarde is `true` als de gebruiker `Yes` heeft ingedrukt en `false` als de gebruiker `Cancel` heeft ingedrukt.

Vul de onderstaande code aan en plaats op de plaats van de ..... de juistge code.

```
var leeftijd=prompt(.....);
var antwoord=confirm('Ben je al jarig geweest (Yes=ja en Cancel=nee)?');
if (.....) {
    document.write("U bent "+ .... + " jaar");
} else {
    document.write(.....);
}
```

## Opdracht 8

Vraag de gebruiker om een getal in te voeren en daarna nog een keer (zoals we dit eerder ook deden bij de opdracht waarbij we twee getallen moesten vermenigvuldigen). Maak nu een scriptje dat deze twee getallen vraagt en druk dan het grootste getal af (op de manier zoals in het voorbeeld hieronder is weergegeven). Als de getallen gelijk zijn dan informeert het scriptje de gebruiker dat de getallen gelijk zijn.

Voorbeelden;

- De gebruiker voert 12 en 10 in. Het scriptje drukt af: *12 is groter dan 10*
- De gebruiker voert 13 en 13 in. Het scriptje drukt af *13 is gelijk aan 13*.

## Opdracht 9

Breidt het scriptje van opgave 8 uit met drie getallen.

Voorbeeld van de output:

De gebruiker voert 12, 10 en 9 in. Het scriptje drukt af: *12 is groter dan 10 en 9*.

De gebruiker voert 11, 10 en 19 in. Het scriptje drukt af: *19 is groter dan 10 en 11*.

De gebruiker voert 12, 12 en 9 in. Het scripje drukt af: *er is geen getal dat groter is dan de anderen.*

De gebruiker voert 12, 12 en 12 in. *Het scripje drukt af: er is geen getal dat groter is dan de anderen.*

## Opdracht 10

Probeer te ontdekken wat de volgende code doet en vul de regels 7 en 9 aan. Zet code op de plaats van de puntjes zodat de code werkt.

```
getal1=Math.floor(Math.random() * 101); // neem een random getal tussen 1 en 100
getal2=Math.floor(Math.random() * 11); // neem een random getal tussen 1 en 10

antwoord=prompt("Wat is "+getal1+"%" +getal2+"?"); // geef de gebruiker een som

document.write("Wat is "+getal1+"%" +getal2+"?<br>"); // schrijf de som naar de browser
document.write("Gegeven antwoord "+.....+"<br>"); // schrijf het gegeven antwoord naar de browser

if ( ..... ){ // is het gegeven antwoord goed?
    document.write("Goed zo!")
} else {
    document.write("Helaas niet goed")
}
```

--

# Les 4, Functions

We kennen nu de basis van JavaScript en gaan nu de volgende stap maken in het leren van JavaScript. In de volgende lessen gaan we het hebben over functies, arrays, objects en loops.

*In deze les gaan we leren wat een functie is.*

## Functions / Functies

Zie ook: [https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

We hebben eigenlijk al kennis gemaakt met functies. We hebben in de eerste lessen de functie *prompt* gebruikt. Weet je nog? De functie *prompt* stelt een vraag aan de gebruiker, die dan iets moet invullen. De waarde die de gebruiker invoert wordt dan in een variable gestopt.

```
antwoord = prompt(vraag);
```

Wat hier staat is dat de functie *prompt* wordt opgeroepen en bij deze oproep wordt er een waarde meegegeven, deze waarde is de string "vraag". Een functie is een soort automaat; je gooit er wat in en je krijgt wat terug. In de *prompt* functie stop je de vraag erin en je krijgt het antwoord eruit.

Je kunt zelf ook functies maken, bijvoorbeeld een functie die twee getallen optelt.

```
function myFunction(a, b) {  
  var antwoord;  
  antwoord = a+b;  
  return antwoord;  
}
```

Een function begin je met *function*, dan komt de naam van de functie die je maakt en dat zet je tussen haakjes twee variabelen. In deze variabelen komt te staan wat je in de functie gooit (net als een automaat waar je munten in gooit). Dan begint de functie met een { en hij eindigt met een } . Daar tussenin staat de code van de functie; dus wat de functie doet. In dit geval wordt er een variabele *antwoord* gemaakt en daar wat de som (de optelling) van *a* en *b* in gezet. Vervolgens wordt de waarde van de variabele terug gegeven als uitkomst.

De variabelen die je aan de functie meegeeft hoeft je niet apart met *var* te declareren (initialiseren).

## parameters

Wat je in een functie 'gooit' heet ook wel de parameters

## return value

Wat een functie terug geeft wordt ook wel de return value genoemd.

Dus....



Een eigen gemaakte functie gebruik je net zoals een in JavaScript ingebouwde functie: de bovenstaande functie roep je aan door:

```
antw=myFunction(3, 4);
```

of

```
antw=myFunction(getal1, getal2);
```

of

```
document.write( myFunction(3,4) );
```

Waarom functies? Functies worden gebruikt om twee redenen:

1. Om lange programma's in nette blokjes op te delen en;
2. om veel voorkomende stukken code één keer te maken in plaats van elke keer opnieuw.

# Oefenen met functies

## Opgave 1

Maak een functie *myProduct* met twee parameters en geef als return value het product (vermenigvuldigen) van deze twee parameters. Test of de functie werkt.

Kopieer de script code op je husiwerkblad.

## Opgave 2

Maak een functie *uitroepteken* met een parameter. Deze parameter is een string. De functie returned dezelfde string maar dan met een uitroepteken er aan vastgeplakt. Test of de functie werkt.



Kopieer de script code op je husiwerkbld.

## Opgave 3

Maak een functie *getalGroterdanNul* met één parameter, een getal. De functie returned één van de drie zinnen: *het getal is kleiner dan nul*, *het getal is precies nul* of *het getal is groter dan nul*. Het zinnetje dat door de functie wordt teruggegeven hangt uiteraard af van de waarde van het getal wat je aan de functie meegeeft.

Kopieer de script code op je husiwerkbld.

## Opgave 4

Maak een functie *posOptellen* met twee parameters. Deze functie geeft de som (optelling) van de twee parameters terug maar als het getal negatief is dan wordt het getal eerst positief gemaakt. Dus, voorbeelden:

posOptellen(3,4) geeft 7 terug  
posOptellen(-3,4) geeft 7 terug  
posOptellen(3,-4) geeft 7 terug  
posOptellen(-3,-4) geeft 7 terug

Test de functie en laat zien dat die werkt.

Kopieer de script code op je husiwerkbld.

## Opgave 5

Als je in een vliegtuig vliegt dan hoor je de piloot altijd zeggen dat ze op bijvoorbeeld 9000 feet vliegen. Om dit om te rekenen in meters moet je dit delen door 3.28. Dus 9000 feet is  $9000/3.28 = 2744$  meter oftewel 2.7 kilometer hoog.

Maak nu een functie waarin je als parameter de feet stops en waarbij de output kilometers is. Dus als je er 9000 in stops dan krijg je er 2.7 uit.

Vraag vervolgens een met de prompt functie aan de gebruiker om de hoogte in feet in te vullen en druk daarna op het scherm (in je browser) af wat de hoogte is in kilometers.

Kopieer de script code op je husiwerkbld.

## Opgave 6- plusopdracht/optioneel

Bekijk onderstaande code en voorspel wat er gebeurt, wat wordt er afgedrukt? Kun je dit verklaren? Deze manier van programmeren heet recursief. Het is als het ware als je van jezelf een foto maakt in de spiegel, dan zie je op de foto de spiegel waarin je weer jezelf ziet in de spiegel waarin je weer jezelf ziet.....

Beschrijf op je huiswerkbld wat er gebeurt.

```
<script>
function myFunction(a) {
  antwoord = a * 2;
  if (antwoord < 100) {
    myFunction(antwoord);
  }
  return (antwoord);
}

document.write( myFunction(5) );
</script>
```

--

# Les 5, Arrays

*In deze les gaan we het over arrays hebben en gaan we leren wat een array is en hoe je er mee kan werken.*

Bestudeer: [https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)

## One Dimensional Array

Array is een verzameling variabelen. De meest eenvoudige vorm van een array is een één dimensionaal array. Dit is eigenlijk niets meer als een lange rij variabelen. Bijvoorbeeld:

```
var myArray = ['auto', 'fiets', 'brommer', 'bus', 'vliegtuig', 'trein'];
```

Hiermee maak je een groep variabelen en die noem je myArray. In de groep variabelen staan 6 strings.

```
document.write(myArray[0]); // je zult auto in je browser zien  
document.write(myArray[5]); // je zult trein in je browser zien
```

Je ziet in deze voorbeelden dat je een element uit je array kan aanroepen met een zogenaamde index. De index is het nummer dat aangeeft welk element je wilt gebruiken. Je ziet ook dat het eerste element is element 0. Dat is in de meeste programmeertalen zo.

Je kun array elementen ook veranderen.

```
myArray['1']='vrachtauto';
```

Element 1 (dat is het 2de element) wordt nu 'vrachtauto'. En daarmee is de waarde 'fiets' overschreven.

Als je wilt weten hoe groot je array is dan kan je dat opvragen met:

```
myArray.length;
```

De waarde hiervan zou in dit voorbeeld 6 zijn. Ons array had immers 6 elementen.

Let wel het laatste element van het array is **myArray[5]** en niet **myArray[6] !!** Dit komt omdat de **index bij 0 begint** te tellen.

## Opgave 1

Gebruik het array in het voorbeeld en druk alle elementen af in de browser.

## Opgave 2

Gebruik het array uit het voorbeeld en wissel element 0 en 1 om. (fiets wordt dus auto en auto wordt fiets). Maak de code zo dat dit de omwisseling altijd werkt ook als de array andere waarden zou hebben.

## Opgave 3

Maak een omwissel-functie waarbij je drie parameters doorgeeft: een array, indexNummer1 en indexNummer2. De return value van de functie wordt het array waarbij er de waarde van element met indexNummer1 en die van indexNummer2 is omgewisseld.

```
function wisselOm(array, index1, index2) {  
  ...  
  ...  
  ...  
  return(array);  
}
```

Plaats de juiste code om de plaats van de puntjes (je kunt meer of minder regels gebruiken).

## Opgave 4

Breidt je omwissel-functie uit zodat er wordt gecontroleerd of de nummers, index1 en index2 niet hetzelfde zijn. Het heeft immers geen zin om een element uit het array om te wisselen met zichzelf. Als de index1 en index2 nummers hetzelfde zijn dan print je een melding naar het console en voer je verder niets uit. Je returned het ongewijzigde array.

## Opgave 5

Breidt je omwissel-functie uit zodat er wordt gecontroleerd of de indexen die je meegeeft correct zijn. Het array heeft namelijk een aantal elementen en de index mag niet te groot zijn omdat je anders naar een element in het array verwijst dat niet bestaat.

Beveilig je functie zondanig dat als je een index zou opgeven die buiten de grensen van joue index zou vallen dat je dan een foutmelding naar het console afdruckt en dat je dan verder niets omwisselt.

--

# 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 afdruckt.

## 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.

--