

# Programmeren

## Module 4: JAVASCRIPT 3

Leerjaar:	1
Crebonummer:	25604
Sector:	Software-developer

## INTRODUCTIE

Je hebt geleerd wat variabelen zijn en je kunt deze gebruiken. Je kunt met lussen werken en je kunt op het canvas element werken. In deze module ga je verder met functies, variabelen en listeners. Ook maak je kennis met een Javascript bibliotheek

De leerdoelen voor deze module zijn:

Je kunt

- Een functie maken en aanroepen
- Variabelen binnen en buiten een functie gebruiken ('scope')
- Parameters en returnwaarden gebruiken in een functie
- Een functie gebruiken als event-handler
- Een library gebruiken (P5.js)
- Fouten in de eigen code opsporen ('debuggen')

## TIJDSPLANNING

<b>Week 1</b>		
BASIS1b, les 1	2 uur	Functies
BASIS1b, les 2	2 uur	Functies die functies aanroepen
BASIS1c	4 uur	Verwerkingsopdracht 1-1
<b>Week 2</b>		
BASIS1b, les 3	2 uur	Scope van variabelen
BASIS1b, les 4	2 uur	Anonieme functies (en scope)
BASIS1c	4 uur	Verwerkingsopdracht 2-1
<b>Week 3</b>		
BASIS1b, les 5	2 uur	Event handlers
BASIS1b, les 6	2 uur	Event listeners
BASIS1c	4 uur	Verwerkingsopdracht 3-1
<b>Week 4</b>		
BASIS1b, les 7	2 uur	P5.js
BASIS1b, les 8	2 uur	Verwerkingsopdracht 4-1
BASIS1c	4 uur	Eindopdracht casus A, B of C (kies er 1!)



# Funcities

## INLEIDING

Een **functie** kan je in Javascript gebruiken om een **groep** opdrachten te kunnen uitvoeren met een aangegeven naam. De opdrachten of statements die binnen de functie staan worden uitgevoerd wanneer de functie wordt aangeroepen. De opdrachten worden dus wel alvast ingelezen, maar nog niet gebruikt.

## EXTERNE BRONNEN

Zie <https://www.youtube.com/watch?v=IclUo13OUCg> voor een uitleg over functies.

Hierin komen events ook al aan de orde. Dat is geen punt; daar ga je volgende week zelf ook mee aan de slag.

Extra uitleg vind je op [https://www.w3schools.com/jsref/prop\\_img\\_src.asp](https://www.w3schools.com/jsref/prop_img_src.asp)

Wil je meteen een uitgebreide uitleg? Kijk dan bij LinkedIn Learning naar de cursus Learning the Javascript Language.

## THEORIE

Een functie is een serie bewerkingen. Door die bewerkingen samen te voegen kun je deze makkelijker nog een keer toepassen.

Een functie begint met het woord function en krijgt vervolgens een naam. Gebruik GEEN hoofdletter als eerste letter. Na de namen komen 2 accolades: () en krulhaken {}  
Tussen de krulhaken komt je code.

Bijvoorbeeld

```
function calculate(){  
    doe iets  
    en doe nog iets  
    en doe vooral nog heel veel meer  
}
```

Een functie kun je vervolgens gebruiken wanneer je op een knopje drukt  
<button onclick="calculate()">Klik</button>

of wanneer je een pagina gaat laden  
<body onload="calculate()">

## OEFENINGEN

### Oefening 1a:

Kijk <https://www.linkedin.com/learning/javascript-functions/what-are-functions?u=36995692>. Kopieer de code uit de video tutorial. Maak een tweede functie die 'min' heet en waarbij je twee getallen van elkaar kunt aftrekken.

### Oefening 1b: kleurswitch wijzigen

Open 1c-kleurswitch. Wanneer iemand op de knop drukt moet de tekst groen worden. Gebruik [https://www.w3schools.com/Jsref/prop\\_style\\_color.asp](https://www.w3schools.com/Jsref/prop_style_color.asp) om te zien hoe je kleuren kunt wijzigen met behulp van JavaScript.

### Oefening 1c: plaatjes aanpassen

Open 1d-plaatjes.zip. Hierin zie je een rollover plaatje (regel 19). Wanneer je in de HTML-code kijkt dan zie je 2 stukjes Javascript die hiervoor verantwoordelijk zijn, de onmouseover:

```
onmouseover="window.document.images[0].src='images/basset2.png';"
```

en onmouseout:

```
onmouseout="window.document.images[0].src='images/basset1.jpg';"
```

Dit kan netter. Pas deze pagina als volgt aan:

1. open het externe Javascript
2. Schrijf 2 functies om de rollover mee te regelen:
  - De 1e functie heet **mijnFunctie1()**. Deze functie moet het plaatje *basset1.jpg* wijzigen in *basset2.png*. Koppel deze functie aan de mouseover.
  - De 2e functie heet **mijnFunctie2()**. Deze functie moet het plaatje *basset2.png* weer wijzigen in *basset1.jpg*. Koppel deze functie aan de mouseout.

# Argumenten gebruiken

## EXTERNE BRONNEN

Zie LinkedIn Learning naar de cursus [Learning the Javascript Language](#) hoofdstukken 6.2 (Arguments in functions) en 6.3 (More on arguments).

## THEORIE: ARGUMENTEN

Omdat we liever lui dan moe zijn, willen we niet te veel hetzelfde doen. Dit is bij programmeren erg belangrijk!

In het voorbeeld hieronder zijn 2 functies gegeven die allebei hetzelfde doen, namelijk de achtergrondkleur wijzigen.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Oef 1 functies</title>
6 <script type="text/javascript">
7   function maakGeel(){
8     document.body.style.backgroundColor = "yellow";
9   }
10  function maakBlauw(){
11    document.body.style.backgroundColor = "blue";
12  }
13 </script>
14 </head>
15 <body>
16   <h1 id="myHeading">Hello world</h1>
17   <button onclick="maakGeel('yellow')">Maak geel</button>
18   <button onclick="maakBlauw('blue')">Maak blauw</button>
19 </body>
20 </html>
```

Voorbeeld 1

Je kunt je voorstellen dat het echt een hels karwei is, wanneer we iemand laten kiezen uit 265 kleuren. Dan moet ik 265 functies schrijven! De oplossing is, werken met argumenten. Dit zijn extra waarden die je tussen de haakjes meegeeft aan de functie.

Je maakt dan 1 functie en tussen de haakjes zet je de waarde die je wilt meegeven. In dit voorbeeld gaat het om een kleur:

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Oef 1 functies</title>
6  <script type="text/javascript">
7      function wijzigBgKleur(kleur){
8          document.body.style.backgroundColor = kleur;
9      }
10     </script>
11 </head>
12 <body>
13     <h1 id="myHeading">Hello world</h1>
14     <button onclick="wijzigBgKleur('yellow')">Maak geel</button>
15     <button onclick="wijzigBgKleur('blue')">Maak blauw</button>
16 </body>
17 </html>|

```

Voorbeeld 2

Het woord 'yellow' op regel 14 en het woord 'blue' op regel 15 worden tijdelijk even opgeslagen als argument 'kleur' (regel 7). Vervolgens wordt dit argument gebruikt om op regel 8 de actie uit te kunnen voeren.

Je mag meer dan 1 argument gebruiken in een functie (probeer maar uit):

```

function glazenBol(baan, locatie, kinderen) {

    let future = 'Je wordt een ' + baan + ' in ' + locatie + ' en hebt ' + kinderen + ' kinderen.';

    console.log(future);

}

glazenBol('beroemde voetbalspeler', 'Spanje', 12);

glazenBol('stunt double', 'Japan', 3000);

glazenBol('ethical hacker', 'Rusland', 0);

```

## OEFENINGEN

### Oefening 1d: Glazen bol

Typ de code van voorbeeld 3 hierboven over en test of deze werkt. Maak er zelf 1 bij. Maak ook eens expres een fout, door bijvoorbeeld een komma weg te halen of haakjes weg te halen. Lees de foutmelding en probeer deze te begrijpen.

### Oefening 1e: Background switch

Wijzig de Javascript zodat er maar 1 functie gebruikt hoeft te worden. Zowel de body als de heading luisteren dan naar dezelfde functie. Tip: deze functie heeft 2 argumenten nodig; 1 voor de kleur en 1 voor de id welke gewijzigd moet worden.

Maak een tweede functie, zodat ook de achtergrondkleur van de heading gewijzigd kan worden.



Voorbeeld 3

### Oefening 1f: debuggen

Open 1f-debuggen.html.

De buttons werken niet. Blijkbaar heb ik iets fout gedaan! Help! Zorg ervoor dat deze wel werken.

## VERWERKINGSOPDRACHT - LES 1 - 4 UUR

# Verwerkingsopdracht 1-1

### Animal age

Maak een spelletje waarbij geraden moet worden hoeveel 1 levensjaar van een dier te vergelijken is met 1 jaar van een mens. De afbeeldingen vind je bij animalAge.zip. Zie animal age.mp4 voor een voorbeeld.

Eisen:

1. Iemand moet een getal kunnen invoeren.
2. Daarna kan iemand op één van de vier buttons klikken om een keuze te maken welk dier hij wil raden.
3. In een uitvoerveld moet vervolgens tekstueel feedback worden gegeven.

### Benodigde kennis

- Je moet een getal uit een tekstveld kunnen uitlezen.
- Dit getal moet je kunnen vermenigvuldigen.
- Je moet een functie kunnen koppelen aan een button.
- Je moet met een if-statement kunnen werken.
- Je moet content in een leeg tekstveld kunnen plaatsen.
- Voor extra punten: je moet een afbeelding kunnen wijzigen.

### Uitbreiding

**Extra 1:** je maakt 1 functie voor alle 4 de knoppen en je gebruikt argumenten om te wisselen tussen de dieren (*zie oefeningen glazen bol en background switch*)

**Extra 2:** de afbeelding wijzigt ook

### Beoordeling:

Het werkt = 7 punten

Extra 1 = 1 punt extra

Extra 2 = 2 punten extra



**Hoe oud is mijn huisdier?**

1 mensenjaar is  jaar voor een:



Hond	Gecko
Olifant	Vlieg

**Resultaat:**

Dit is het startscherm. Iemand moet eerst een getal invullen en daarna op een dier klikken.

De goede antwoorden zijn overigens:

Hond = 5,5 jaar


Gecko = 11 jaar

Olifant = 1,6 jaar

Vlieg = 1216 jaar

**Hoe oud is mijn huisdier?**

1 mensenjaar is  jaar voor een:



Hond	Gecko
Olifant	Vlieg

**Resultaat:**  
Kies lager, voor een hond is dit niet 7 jaar

Wanneer iemand te hoog heeft geraden, krijgt hij/zij in het uitvoerveld te lezen dat het getal te hoog is. De afbeelding wijzigt dan in een pijl naar beneden.

**Hoe oud is mijn huisdier?**

1 mensenjaar is  jaar voor een:



Hond	Gecko
Olifant	Vlieg

**Resultaat:**  
Kies hoger, voor een hond is dit niet 1 jaar

Wanneer iemand te laag heeft geraden, wijzigt de pijl in een pijl omhoog.

**Hoe oud is mijn huisdier?**

1 mensenjaar is  jaar voor een:



Hond	Gecko
Olifant	Vlieg

**Resultaat:**  
Wanneer een hond 1 jaar oud is dan is deze in mensenjaren inderdaad 5.5 jaar!

Wanneer het goed is, krijg je de uitslag in een ander veld op de pagina en wijzigt de afbeelding links.

## De scope van variabelen

### INLEIDING

Wanneer een variabele wordt aangemaakt (dit noemen we declareren), bepaal je ook de levensduur van de variabele. Je moet als programmeur dus van tevoren beslissen wanneer (en hoe lang) je de variabele wilt kunnen aanspreken.

Bijvoorbeeld:

Ik speel graag Zelda. In dat spel weet ik pas hoe sterk mijn vijand is wanneer ik ga vechten. Dan krijg ik een balkje te zien bij mijn vijand. Wanneer ik van alle vijanden de hele tijd die balkjes te zien krijg, dan kan ik niet meer spelen! Door het aanroepen van een functie vechten (bijvoorbeeld) worden pas de variabelen 'leven' getoond.

### EXTERNE BRONNEN

1. <https://www.youtube.com/watch?v=kjdZ1h3CIs4>
2. <https://www.linkedin.com/learning/javascript-essential-training-3/variable-scope?u=36995692>
3. Hoofdstuk 6.7 (Variable Scope in Javascript) van de LinkedIn Learning cursus "Learning the Javascript Language"

### THEORIE

Variabelen bestaan in 2 smaken: lokaal en globaal.

Lokaal = klein/ bestaat alleen binnen een functie.

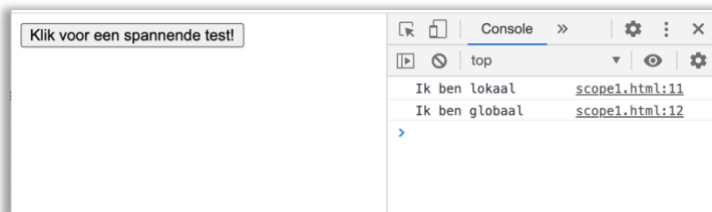
Globaal = groot/ kan door het hele script (en alle functies) gelezen worden.

In het spel Zelda zou het rode levensbalkje van een vijand prima door een lokale variabele beheerd kunnen worden. Deze is alleen nodig tijdens het gevecht. Wanneer het vechten klaar is, hoef ik 'm niet meer te zien.

In het spel Zelda zou een globale variabele bijvoorbeeld de weersomstandigheid kunnen zijn. Deze heeft invloed op zeer veel elementen in het spel. Bij regen kan ik slechter klimmen, kan ik minder zien, maar kan de vijand mij minder goed horen (en zien) en moeten er spetters onder mijn voeten te zien zijn enzovoorts. Kortom: heel veel functies (en karakters) worden beïnvloed door dit onderdeel.

```
<script type="text/javascript">
  let globaleVariabele = "Ik ben globaal";

  function test(){
    let lokaleVariabele = "Ik ben lokaal";
    console.log(lokaalVariabele);
    console.log(globaleVariabele);
  }
</script>
```

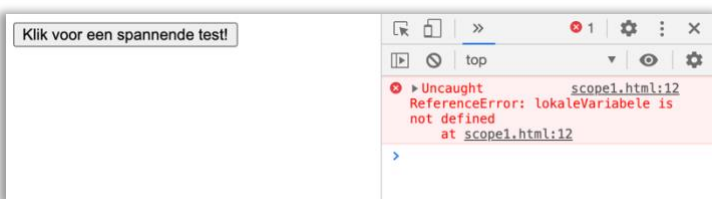


### Voorbeeld A

De lokale variabele en de globale variabele zijn zichtbaar wanneer de functie wordt aangeroepen. In dit geval wanneer er op de knop wordt geklikt.

```
<script type="text/javascript">
  let globaleVariabele = "Ik ben globaal";

  function test(){
    let lokaleVariabele = "Ik ben lokaal";
  }
  console.log(lokaalVariabele);
  console.log(globaleVariabele);
</script>
```



### Voorbeeld B

Wanneer ik de console.log functie BUITEN de functietest aanroep, dan heb ik een probleem.

De variabele lokaleVariabele kan niet gevonden worden. Dit komt omdat de functie is afgelopen. De variabele lokaleVariabele bestaat niet meer.

## OEFENINGEN

### Oefening 2a

Maak voorbeeld A hierboven na en kijk goed wat er gebeurt.

### Oefening 2b

Maak voorbeeld B hierboven na en kijk goed naar de foutmelding, zodat je deze herkent wanneer het je overkomt!

### Oefening 2c

Open 2d-scope.html in de browser en in je editor.

In dit documentje staan 2 functies die allebei getallen kunnen optellen. Beide functies schrijven de uitkomst naar de HTML-pagina.

Bekijk de code. Op regels 23 t/m 33 staan vragen die je moet beantwoorden. Probeer eerst de antwoorden uit je hoofd te berekenen (of met een kladblaadje) en test het dan door in je

console de functie (met parameter) aan te roepen. Je ziet dan op de pagina het goede antwoord.

## THEORIE

Waarom is het belangrijk om nu let te gebruiken en niet var?

Zie <https://www.linkedin.com/learning/javascript-essential-training-3/es2015-let-and-const?u=36995692>

### *Let op*

*Gebruik vanaf nu echt alleen maar **let** en niet **var** wanneer je een variabele declareert.*

## OEFENINGEN

### Oefening 2d

Typ onderstaande code eens over en test deze in de browser. Gebruik de console om het resultaat te zien. Als het goed is, zie je een 2 en een 1 verschijnen:

```
<script type="text/javascript">
  let x = 1;
  {
    let x = 2;
    console.log(x); // hier komt 2 uit
  }
  console.log(x); // hier komt 1 uit
</script>
```

### Oefening 2e

En wijzig nu let eens in var. Wat gebeurt er?

Hieruit blijkt dat, wanneer je een var wijzigt binnen een functie, deze een nieuwe waarde krijgt. Bij een let is dat niet. De scope van een let is preciezer dan van een var. De let werkt tussen de {} haken en de var kan ruimer werken.

Dit voorbeeld komt trouwens uit: <https://codeburst.io/javascript-learn-understand-scope-f53d6592c726>. Je kunt dit op eigen gelegenheid nog eens nalezen.

# Anonieme functies

## INLEIDING

Een anonieme functie is een functie die direct aan een variabele gekoppeld wordt. Wanneer je deze nodig hebt, roep je de variabele aan met () erachter. Wat is hier handig aan? Soms heb je een bewerking maar 1 keer nodig en is deze onderdeel van een andere functie. Om dan helemaal een nieuwe functie te schrijven kan wat onoverzichtelijk worden. Een anonieme functie is dan beter. Ook handig is om een anonieme functie te gebruiken in een loop.

## EXTERNE BRONNEN

<https://www.javascripttutorial.net/javascript-anonymous-functions/>

<https://dev.to/designpuddle/coding-concepts---anonymous-methods-ago>

## THEORIE

Hieronder zie je een functie.

De functie wordt aangeroepen VOORDAT de functie zelf wordt gedeclareerd. Dat is geen probleem.

```
<script type="text/javascript">
  sayHello();

  function sayHello(){
    alert("Hello");
  }
</script>
```

Hieronder hetzelfde voorbeeld, maar nu als een anonieme functie. Nu is het belangrijk dat eerst de variabele (met anonieme functie) wordt gedeclareerd, voordat je deze kunt gebruiken.

```
<script type="text/javascript">
  var sayHello = function {
    alert("Hello");
  }
  sayHello();
</script>
```



Wat een beetje gek is hierbij, is dat je een gewone variabele (sayHello) aanroept als een functie: sayHello();

Let daar dus op!

## Help?

Moet je nu met anonieme functies werken om een voldoende te halen? Nee.

Maar je komt anonieme functies best vaak tegen in tutorials op internet. Vandaar dat deze nu in de les wordt behandeld.

## OEFENINGEN

### Oefening 2f

```
<script type="text/javascript">
  function a() {
    console.log("Dit is een melding vanuit functie A");
  }
  let b = function() {
    console.log("Dit is variabele b met een anonieme functie");
  }
  a();
  b();
</script>
```

Kopieer de code van het voorbeeld.

Typ zelf een nieuwe anonieme functie die c heet en die ook iets in de console schrijft.

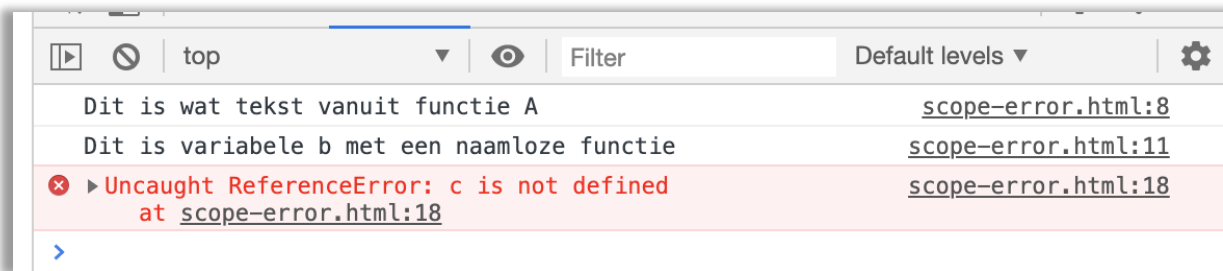
## THEORIE

### Anonieme functies debuggen

Met anonieme functies wordt het helemaal belangrijk om te letten op de scope van de variabelen. Kijk maar eens naar het onderstaande voorbeeld. Hierin worden 3 variabelen gedeclareerd (a, b en c) maar bij c gaat er iets mis:

```
<script type="text/javascript">
  function a() {
    console.log("Dit is wat tekst vanuit functie A");
  }
  let b = function() {
    console.log("Dit is variabele b met een naamloze functie");
    let c = function() {
      console.log("Dit is variabele c met een naamloze functie");
    }
  }
  a();
  b();
  c();
</script>
```

Ik krijg de volgende foutmelding:



### Wat gaat er mis?

De variabele c wordt binnen de naamloze functie van b gebruikt. Je kunt die niet rechtstreeks aanroepen. Alleen functie b kan gebruik maken van functie c.

## OEFENINGEN

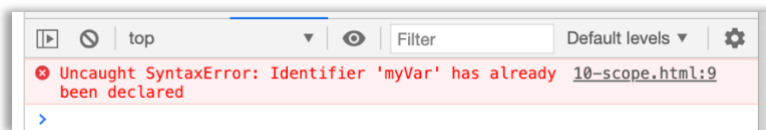
### Oefening 2g

Open 2g-debug.html in de browser (en open ook de console). Er verschijnt *undefined* in de console. Wat gaat er mis en waarom? Schrijf je antwoord als comment in de code erbij. Tip: typ de code even over, dat werkt makkelijker.

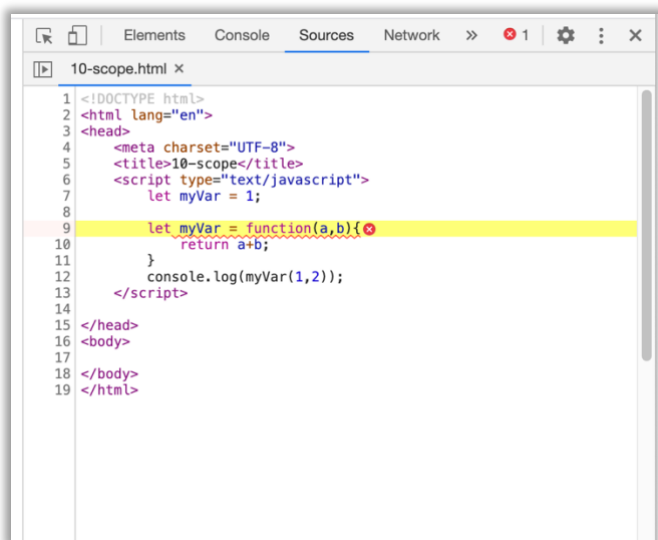
### Oefening 2h

Deze oefening gaat over het debuggen van een anonieme functie.

Open 8-debug in de browser (en open ook de console). Je ziet een foutmelding.



Lees de foutmelding goed en probeer deze voor jezelf te vertalen.




Klik op het regelnummer achter de foutmelding om te zien waar het probleem precies zit.

De foutmelding dat identifier (= variabele) myVar al is gedeclareerd op regel 9, wordt getoond. Los dit op. Zoek eventueel op Internet op wat de foutmelding betekent.







## VERWERKINGSOPDRACHT - LES 2 - 4 UUR

### Verwerkingsopdracht 2-1


Een groentespecialist wil een online winkel beginnen. Als test wil hij appels, peren en bananen verkopen. Jij moet een opzetje maken. De HTML-pagina wordt aangeleverd, jij moet het met Javascript werkend krijgen. Zie winkelwagentje.zip.






#### Uw groentespecialist

Product	Omschrijving		Aantal	Totaal
 <p>€ 1</p> <p>Sappiger kan gewoon niet! Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse id sem mollis, tincidunt risus sed, vulputate risus.</p>		<input type="text"/>	<input type="text"/>	
 <p>€ 2</p> <p>Vooral lekker als ijsje! Nulla eget erat orci. Maecenas blandit lacus sed risus efficitur ultricies. Quisque ornare massa non congue congue</p>		<input type="text"/>	<input type="text"/>	
 <p>€ 3</p> <p>Mooi krom en geel! Duis vel sem lacus. Fusce efficitur suscipit ultrices. Etiam ullamcorper laoreet justo</p>		<input type="text"/>	<input type="text"/>	
<input type="button" value="Bereken totaalprijs"/>			<input type="text"/>	

Het is de bedoeling dat iemand op de +-button achter het fruit kan klikken waardoor er een aantal en een totaalprijs van dat fruit te zien is. Onderaan worden alle totaalprijzen bij elkaar opgeteld.



**Uw groentespecialist**

Product	Omschrijving		Aantal	Totaal
	€ 1 Sappiger kan gewoon niet! Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse id sem mollis, tincidunt risus sed, vulputate risus.	+	<input type="text" value="2"/>	<input type="text" value="2"/>
	€ 2 Vooral lekker als ijsje! Nulla eget erat orci. Maecenas blandit lacus sed risus efficitur ultricies. Quisque ornare massa non congue congue	+	<input type="text" value="1"/>	<input type="text" value="2"/>
	€ 3 Mooi krom en geel! Duis vel sem lacus. Fusce efficitur suscipit ultrices. Etiam ullamcorper laoreet justo	+	<input type="text" value="3"/>	<input type="text" value="9"/>
Bereken totaalprijs			<input type="text" value="13"/>	

Let op: het hoeft alleen maar te werken wanneer er op de button geklikt wordt. Iemand hoeft geen aantallen op te kunnen geven.

Voor de overzichtelijkheid in de code wil je werkgever dat je *in ieder geval* gebruik maakt van 2 functies. Waarschijnlijk gebruik je 3 functies, dat is ook goed. Schrijf één functie voor de plus-knop, één voor de min-knop en één voor de "Bereken totaalprijs"-knop.

Je mag zelf beslissen of je aparte functies schrijft voor de appels, de peren en de bananen. In dat geval krijg je meer dan 3 functies. Je werkgever heeft een voorkeur voor één functie waarbij je met behulp van argumenten bekijkt of iemand appels, peren of bananen koopt.

#### Eisen

- er zijn in ieder geval 2 functies (meer mag ook).
- de optellingen zijn correct.

#### Benodigde kennis


- Je moet een getal uit een tekstveld kunnen uitlezen.
- Dit getal moet je kunnen vermenigvuldigen (*zie oefening favosnack*).
- Je moet een functie kunnen koppelen aan een button.
- In die functie heb je 1 of meer argumenten nodig (*zie oefeningen glazen bol en background switch*).
- Je moet content in een leeg tekstveld kunnen plaatsen.
- Je moet kunnen beslissen of een variabele een lokale of globale scope heeft.






## Uitbreiding

**Extra 1:** zorg dat er ook een werkende min-knop is. Deze min-knop mag nooit onder de nul komen.

**Extra 2:** maak 1 functie voor het toevoegen en verwijderen van de producten. Je project heeft dan in totaal maximaal 2 functies: één voor het toevoegen en verwijderen van de producten en één voor de totaaloptelling.



### Uw groentespecialist

Product	Omschrijving		Aantal	Totaal
 <p>€ 1</p> <p>Sappiger kan gewoon niet!                      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse id sem mollis, tincidunt risus sed, vulputate risus.</p>	<div>+</div> <div>-</div>	<input type="text" value="5"/>	<input type="text" value="5"/>	
 <p>€ 2</p> <p>Vooral lekker als ijsje!                      Nulla eget erat orci. Maecenas blandit lacus sed risus efficitur ultricies. Quisque ornare massa non congue congue</p>	<div>+</div> <div>-</div>	<input type="text" value="1"/>	<input type="text" value="2"/>	
 <p>€ 3</p> <p>Mooi krom en geel!                      Duis vel sem lacus. Fusce efficitur suscipit ultrices. Etiam ullamcorper laoreet justo</p>	<div>+</div> <div>-</div>	<input type="text" value="0"/>	<input type="text" value="0"/>	
<div>Bereken totaalprijs</div>			<input type="text" value="7"/>	

## Beoordeling:

Het werkt = 7 punten

Extra 1 = 1 punt extra

Extra 2 = 2 punten extra

## Event handlers

### INLEIDING

**Events** vormen het hart van een Javascriptpagina. Event handling staat toe dat de Javascriptopdrachten op het juiste moment worden uitgevoerd, wanneer de juiste gebeurtenis heeft plaatsgevonden, bijvoorbeeld:

- Een berekening wordt uitgevoerd wanneer een gebruiker op een knop klikt.
- Een figuur verandert wanneer de gebruiker er met de muis over beweegt.
- Een animatie wordt gestart wanneer de pagina geladen is.
- Een plaatje wijzigt wanneer een gebruiker er met de muis overheen rolt.

In dit hoofdstuk ga je zelf met event handlers aan de slag.

### EXTERNE BRONNEN

De volgende tutorials geven meer uitleg en voorbeelden over events:

- Video: <https://www.youtube.com/watch?v=ZvoKeYGrdDc>
- [https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)
- [https://www.w3schools.com/jsref/dom\\_obj\\_event.asp](https://www.w3schools.com/jsref/dom_obj_event.asp)
- <https://javascript.info/introduction-browser-events>

### THEORIE

Je kunt Javascript laten reageren op bepaalde acties van de bezoeker. Hiermee kan je voor meer interactiviteit zorgen op een bepaalde pagina.

Een event kan bijvoorbeeld zijn:

- Een bezoeker die met de muis over een link beweegt.
- Een pagina die in de browser van de bezoeker geladen wordt.

Nadat zo'n event zich voordoet, wordt er een Javascript code uitgevoerd in de browser. Deze code wordt ook wel een **event handler** genoemd.

Hieronder zie je een overzichtje van enkele veelgebruikte events in Javascript. Een aantal van deze events zal je misschien al bekend voorkomen.

Event-naam:	Omschrijving:
onClick = "..."	De muis klikt ergens op (bijv. een link of knop)
onLoad = "..."	Als de pagina geladen is
onUnload = "..."	Als een ander bestand wordt geladen in de browser
onMouseOver = "..."	Als de muis ergens over beweegt (link, plaatje, ...)

<b>onMouseOut = "..."</b>	Als de muis ergens van af beweegt (link, plaatje, ...)
<b>onSelect = "..."</b>	Als een tekst(-veld) wordt geselecteerd
<b>onSubmit = "..."</b>	Als een formulier wordt verstuurd
<b>onChange = "..."</b>	Als de inhoud van een (tekst-)veld wordt aangepast
<b>onBlur = "..."</b>	Als je een (tekst-)veld verlaat

Let op: een event begint altijd met *on*.

### Properties, Methods en Event types

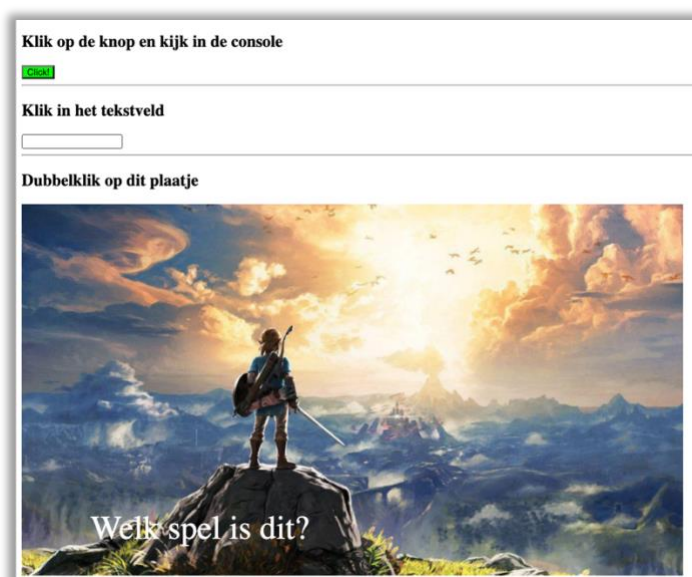
Een event 'kan' bepaalde zaken. Bijvoorbeeld een MouseEvent kan doorgeven wanneer er met de muis geklikt wordt of wanneer de muis bewogen wordt. Dit heten **event types**. Echter, het MouseEvent kan ook waarden teruggeven, bijvoorbeeld de x- en y-positie van waar de muis staat in het scherm of doorgeven of de gebruiker een Shift-toets indrukt terwijl hij/zij met de muis klikt. Dit zijn properties of methods. Een property is een waarde (die x-positie), een method is een actie.

## OEFFENINGEN

### Oefening 3a: drie handlers

Open 3a-events.html. In deze pagina zie je al 3 event handlers staan:

- Op de knop (er verschijnt tekst in de console.log)
- Op het tekstveld (de kleur van de input field wijzigt)
- Op het plaatje (de tekst in het divje wijzigt)



Maak er zelf 3 event handlers bij. Dit mogen niet dezelfde event handlers zijn als die al gebruikt zijn. Je moet er ook zelf de extra HTML-elementen aan toevoegen.

Zie [https://www.w3schools.com/jsref/dom\\_obj\\_event.asp](https://www.w3schools.com/jsref/dom_obj_event.asp) voor een overzicht van events. In dit overzicht staat het woord on nog niet voor de events. Klik op een event om naar de pagina met meer informatie te gaan; daar vind je de volledige code.

### Oefening 3b: events

Open 3b-events.html.

Zorg ervoor dat de functie goedemorgen gaat werken nadat de pagina geladen is. Bekijk zelf welk event je daarvoor moet gebruiken.

### Oefening 3c: codepen

Open <https://codepen.io/prosetech/pen/oRxMmZ>

Hier zie je een HTML, een CSS en een Javascript. Met het CSS-bestand hoef je niets te doen, maar de juiste Javascript**functies** moeten aan de HTML-elementen gekoppeld worden zodat het verhaal klopt. Zoek zelf welke event je daarvoor moet gebruiken.

Test je werk door in het voorbeeld onderin op de knoppen te klikken en je naam in te vullen.

## Debuggen

### OEFENINGEN

#### Oefening 3d:

Open 3d-event.html. Het is de bedoeling dat de achtergrondkleur van de body groen wordt wanneer je op de knop drukt. Ik zie geen fouten in de console, maar toch werkt het niet!

Help!

# Event listeners

## THEORIE

Het is netjes en handig om alle code bij elkaar te hebben staan. Met behulp van event listeners hoef je geen events aan te roepen in je HTML-pagina, maar roep je een gebeurtenis aan in het Javascriptbestand.

Nog een voordeel: wanneer je items dynamisch gaat laden, dan weet je van tevoren niet hoeveel elementen je op de website zult hebben. Bijvoorbeeld een winkelwagentje: een gebruiker kan 1 item kopen, maar ook 100. Echter, alle knopjes zullen een rollover moeten hebben. Met een event listener kun je dat makkelijker regelen, dan met losse event handlers.

## EXTERNE BRONNEN

Voor extra informatie kan je gebruik maken van de volgende links:

1. Video: <https://www.linkedin.com/learning/javascript-events/understanding-event-registration?v=36995692>.
2. Video: <https://www.youtube.com/watch?v=EaRrmOtPYTM>

## 2 VOORBEELD

Hier zie je 2 event listeners in het script. Kopieer de code maar eens en probeer het zelf uit.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>EventListeners</title>
    <meta name="description" content="eventListeners">
  </head>
  <body>
    <button id="button">Klik!</button>
    <script type="text/javascript">
      window.addEventListener("load", function(){
        document.body.style.backgroundColor = "red";
      })
      document.getElementById("button").addEventListener("click", function(){
        document.body.style.backgroundColor = "green";
      })
    </script>
  </body>
</html>
```



## OEFENINGEN

### Oefening 3e: rollover

Rollover afbeeldingen worden heel veel gebruikt op het Internet. Bij de volgende oefening ga je 3 rollover afbeeldingen maken.

Pak 3e-rollover.zip uit. Hierin vind je een HTML-pagina, een Javascript pagina en 6 afbeeldingen.

1. Maak voor iedere afbeelding een rollover waarbij de kleurenfoto zichtbaar is. Je moet een event listener gebruiken!
2. Wanneer de gebruiker weer van de afbeelding afrolt, moet de zwart-wit foto weer getoond worden.

Tip: voor het aanroepen van een afbeelding op een pagina kun je natuurlijk getElementById gebruiken maar je kunt ook het volgende gebruiken

```
window.document.images[0].src="mijnPlaatje";
```

waarbij plaatje nummer nul het eerste plaatje op je pagina is.

### Oefening 3f:

Open 3f-rangeSlider.html

De code is afkomstig van voorbeeld 3 van

[https://www.w3schools.com/howto/howto\\_js\\_rangeslider.asp](https://www.w3schools.com/howto/howto_js_rangeslider.asp) maar wat extra is, is het rode vierkantje. Wanneer je de slider beweegt, beweegt het rode vierkantje van links naar rechts. Het is de bedoeling dat dit rode vierkantje ook omhoog en naar beneden gaat wanneer je de slider beweegt. Zorg daarvoor.

## Debuggen

### OEFENINGEN

### Oefening 3g:

Open 3g-debug eventlistener.html in de browser en open de console. Rol over de afbeelding om een foutmelding te krijgen. Los deze op.

## VERWERKINGSOPDRACHT - LES 3 - 4 UUR

### Verwerkingsopdracht 3-1

Tijdens deze opdracht ga je een scriptje schrijven waarmee je geluid kunt afspelen met behulp van Javascript. Wanneer iemand de A, de S, de D of de F-toets indrukt op het toetsenbord klinkt er een geluid. Dit is steeds een ander geluidje.

Er is geen HTML-file aangeleverd, wel muziekfiles.

Tip om te starten:

1. Maak een array met de 4 sounds. Zie <https://stackoverflow.com/questions/30289217/how-do-i-loop-through-a-javascript-array-of-audio-files> maar je hebt daar lang niet alles van nodig; typ dus niet klakkeloos over!
2. Maak een functie die op een key press wacht. Gebruik hiervoor NIET de functie `keyCode`, deze is ouderwets!
3. Deze functie bekijkt of iemand op de 'a', de 's', de 'd' of de 'f' heeft gedrukt.
4. Zo ja, dan speelt geluid nummer 1, 2, 3 of 4.
5. Zo nee, dan hoeft er niets te gebeuren.

#### Eisen

- Het is de bedoeling dat je event listeners gebruikt (of event handlers).
- Wanneer iemand op de 'a' drukt, gaat geluid 1 afspelen.
- Wanneer iemand op de 's' drukt, gaat geluid 2 afspelen.
- Wanneer iemand op de 'd' drukt, gaat geluid 3 afspelen.
- Wanneer iemand op de 'f' drukt, gaat geluid 4 afspelen.
- Géén gebruik van de functie `keyCode`.

#### Benodigde kennis

- Gebruik van event handlers of event listeners.
- Gebruik een array (zie lesmateriaal vorige periode).
- Invoegen van een audio-element op de HTML-pagina (zoek zelf naar een tutorial).

#### Uitbreidingen

**Extra 1:** Er moet ook een knop zijn die het geluid weer laat stoppen.

**Extra 2:** Maak de pagina mooi op bijvoorbeeld door een mooie play list erbij te maken (zodat ik weet op welke knoppen ik moet drukken) en maak er knoppen bij die ook een sound afspelen.

**Beoordeling**

Het werkt = 7 punten

Extra 1 = 1 punt extra

Extra 2 = maximaal 2 punten extra

## Javascript libraries – P5.js

### THEORIE

P5 is een library (= een verzameling) Javascriptjes waarmee je vrij makkelijk grafische effecten op je website kunt maken. Het kan goed gebruikt worden om animaties en afbeeldingen mee te maken (of te bewerken).

P5.js moet je downloaden. Je krijgt dan een serie Javascriptbestanden en een HTML-bestand. Daarnaast is het de bedoeling dat je je eigen Javascriptbestand schrijft; je werkt NIET verder in de bestanden die aangeleverd worden. Die laat je met rust.

P5.js heeft een goede documentatie en er zijn voldoende video's en tutorials te vinden om je op weg te helpen. Laten we eerst eens kijken hoe P5 eruitziet.

### OEFFENINGEN

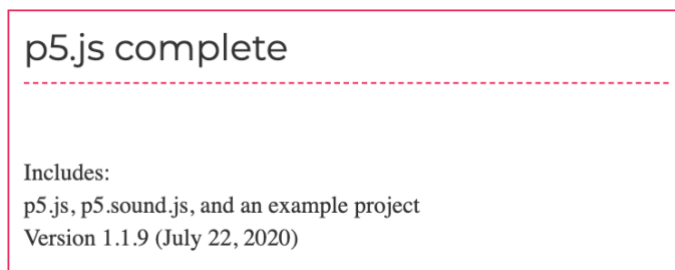
#### Oefening 4a:

Ga naar <https://p5js.org/reference/#/p5/setup> Je ziet op die pagina een live editor. Hiermee kun je het geanimeerde vierkantje aanpassen. Verander de waarde eens bij de background, de fill en de rect functie. Wat gebeurt er?

#### Oefening 4b:

Pak 2-P5.zip uit en bewaar het zipbestand. Dan heb je altijd een werkende basis indien het mis gaat.

NB: je kunt P5 ook downloaden via <https://p5js.org/download/>



Hierin vind je een opzet voor een website. In het Javascriptbestand vind je een setup en draw functie. Let op: de index pagina staat niet in de root en je krijgt (bij offline werken) foutmeldingen over sound.js. Je moet daar dus nog iets aan knutselen.

### Oefening 4c:

Gebruik <https://p5js.org/get-started/#settingUp> om een cirkel te tekenen.

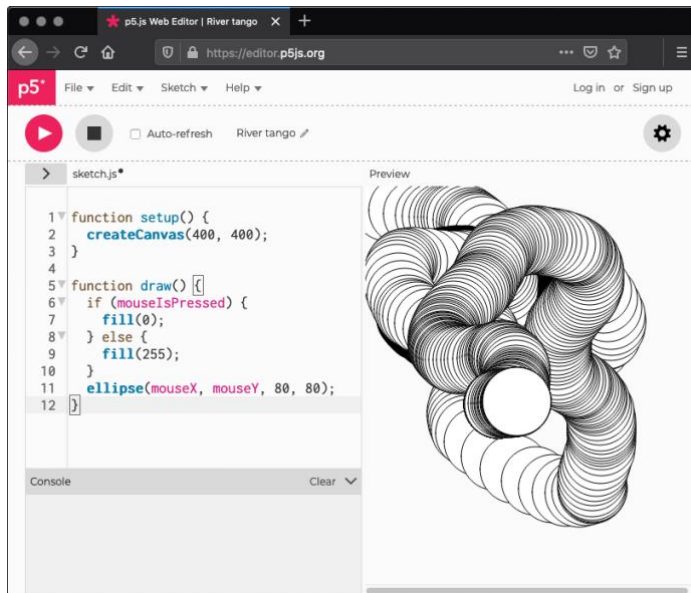
### Oefening 4d:

Ga naar <https://p5js.org/get-started/> en volg de tutorial op de pagina. Let op: maak een kopie van je basismap en noem die map 4d-started. Werk verder in deze map.

Lukt het je om de cirkeltjes op je pagina te krijgen?

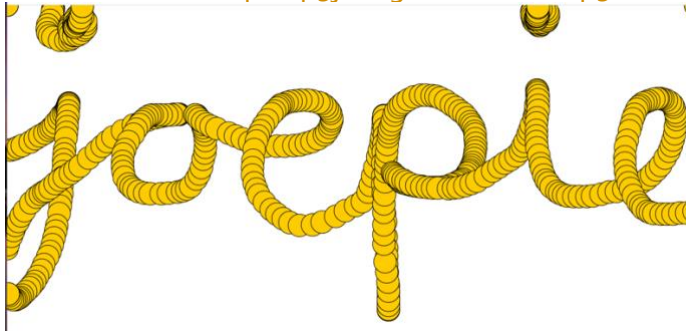
*Tip: wanneer je de inspect element open hebt staan, werkt het iets anders.*

*Let op dat je (wanneer de inspect element open staat) kiest voor 'Laptop with HiDPI screen'.*



### Oefening 4e:

Wijzig eens wat waarden. Maak het tekenveld groter, de cirkels kleiner, pas de kleur van de cirkels aan (zie <https://p5js.org/reference/#/p5/color>).



### Oefening 4f:

Lukt het je ook om de oefening zo te wijzigen dat je alleen kunt tekenen wanneer iemand op de muis klikt en anders niet? En om die vervelende zwarte rand weg te krijgen?



### Tip:

Ben je die cirkeltjes nu wel zat? Kijk dan eens bij <https://p5js.org/reference/> voor meer vormen. Voeg aan je document nog 2 vormen toe.

## THEORIE

### Animatie in js.

Per seconde wordt de drawfunctie (ongeveer) 60 keer aangeroepen door de browser. Dit betekent dat er ongeveer 60 keer per seconde iets op het scherm wordt getekend. Dat lijkt dan op een animatie. Dit betekent niet dat er altijd iets beweegt op het scherm; er hoeft natuurlijk geen animatie te zijn. Echter, onthoud wel dat de drawfunctie 60 keer per seconde wordt aangeroepen. Ik merkte zelf dat dit soms tot een flikkerend beeld kan leiden; mijn code was toen ook niet helemaal goed.

## EXTERNE BRONNEN

Daniel Shiffman heeft een kanaal vol met tutorials over P5. Wanneer je deze allemaal wilt kijken, dan ben je zeker 6 uur bezig. We zullen er dus kleine stukjes van kijken.

<https://www.youtube.com/playlist?list=PLRqwX-V7Uu6Zy51Q-xgtMWlv9cueOFTFA>

## OEFENINGEN

### Oefening 4g - animatie

Open oefening 4g-animatie.html. Je ziet dat de cirkel van links naar rechts beweegt. Speel eens met de code; wijzig getallen om te zien wat er gebeurt en probeer de cirkel eens van rechts naar links te laten bewegen.

### Extra oefening 4h

Tijd over? Kijk [https://www.youtube.com/watch?v=LO3Awjn\\_gyU&list=PLRqwX-V7Uu6Zy51Q-xgtMWlv9cueOFTFA&index=15](https://www.youtube.com/watch?v=LO3Awjn_gyU&list=PLRqwX-V7Uu6Zy51Q-xgtMWlv9cueOFTFA&index=15) en open 4h-bounce.

Je ziet daarin al het basisproject waarmee de tutorial ook begint. In de tutorial wordt je uitgelegd hoe je het balletje moet laten stuiteren tegen de rechterzijde van het browserscherm. Zorg er zelf voor dat de bal ook tegen de linkerkant kan stuiteren.

Hier een andere opdracht maken. Een simpele

INHOUD WEEK 16 - LES 4B - 2 UUR

## P5 3D (als extra: hoeft niet)

### THEORIE

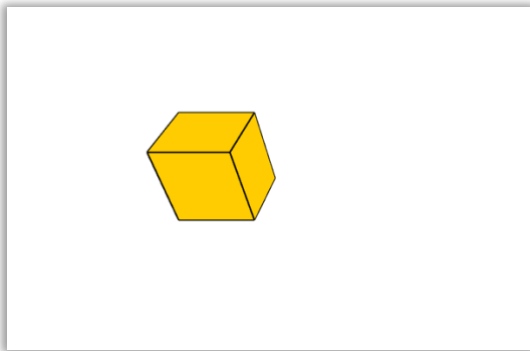
Met P5 kunt je 3D tekenen in de browser. Zie

<https://github.com/processing/p5.js/wiki/Getting-started-with-WebGL-in-p5>

### OEFENINGEN

#### Oefening 4i:

Maak met behulp van [translate en rotate](#) (alleen het onderdeel 'translate, rotate') en [fill](#) de volgende tekening:



#### Oefening 4j

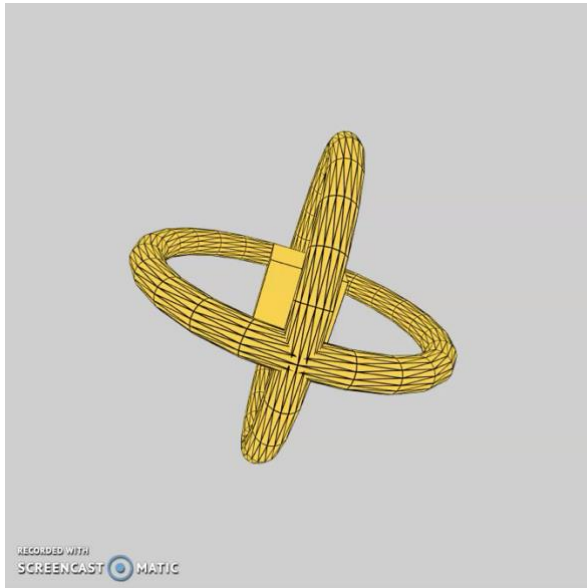
En maak dan ook een torus



#### Oefening 4k - 3D-animatie

Gebruik de volgende linkjes om onderstaande animatie te maken:

- <https://p5js.org/reference/#/p5/box>
- <https://p5js.org/reference/#/p5/rotateY>



## Oefening 4I

Ga naar <https://p5js.org/reference/#/p5/createCapture> en probeer alle drie de codes eens uit. Je kunt 4I-capture gebruiken als startbestand.

Wil je meer weten over video capture? Kijk dan <https://youtu.be/bkGf4fEHKak>.

## GEWOON FUN

Ben je benieuwd wat de groten der aarde ;- ) nu zoal kunnen maken met processing? Kijk dan eens op de profielen van [Saskia Freeke](#), [Joshua Davis](#), [Dave Whyte](#) of [Denial of Service](#) ([muziekvideo's](#)). Wie zegt er nu dat programmeren niet creatief is?

Wil je lezen hoe Saskia aan zoveel werk komt? [Hier vertelt ze over haar project](#) om iedere dag iets te maken en te posten (op social media). Wellicht inspireert het je om zelf ook zoiets te doen!

Goed te volgen voorbeelden vind je ook op: <https://genekogan.com/code/p5js-transformations/>

## Verwerkingsopdracht 4-1

Teken met P5 een huisje met een wolkje erboven.  
Je mag zelf kiezen of je het in 2D of in 3D tekent.

Tip om te starten:

- Zie <https://p5js.org/reference/> voor een overzicht van shapes en functies.

### Eisen

- Je hebt een huisje getekend
- Je hebt een wolkje getekend

### Benodigde kennis

- Basiskennis van P5.

### Uitbereidingen

**Extra 1:** De wolk moet bewegen.

**Extra 2:** Hoe mooier je huisje en hoe meer er animeert, hoe hoger je cijfer. Kortom: voeg bomen toe, konijnen, rook uit de schoolsteen, een dinosaurus, een UFO...

### Beoordeling

Het werkt = 7 punten

Extra 1 = 1 punt extra

Extra 2 = maximaal 2 punten extra

## EINDOPDRACHT - 4 UUR

De eindopdracht voor BASIS 1C is een individuele opdracht. Er zijn 3 opdrachten (casussen), daar kies je er 1 van. Je krijgt voor deze opdracht een cijfer maar ook een werkproces beoordeling.

Voor alle 3 de casussen gelden dezelfde werkprocessen:

B1-K1-W3: realiseert (onderdelen van) software

B1-K1-W4:

## Beoordeling werkprocessen

Bij alle casussen word je op de volgende werkprocessen beoordeeld. De werkprocessen worden individueel beoordeeld. Op de laatste pagina staat het gebruikte beoordelingsformulier afgedrukt.

In totaal zijn er 9 indicatoren te behalen voor werkproces B1-K1-W3.  
Je moet 7 indicatoren behaald hebben voor een voldoende.

B1.-K1-W3

- 1-3-1: kies de juiste materialen en middelen (gebruikersinterface, software/softwaretools - editors en compilers) en gebruikt deze effectief.
- 1-3-2: hanteer de code conventies volgens de voorgeschreven wijze.
- 1-3-3: realiseer software die netjes en goed leesbaar is.
- 1-3-4: realiseer de software nauwgezet conform de eisen uit opdracht en ontwerp.
- 1-3-5: presteer onder (tijds)druk en/of in een stressvolle omgeving effectief en productief.

B1.-K1-W4

- bewaakt de kwaliteit conform de eisen die in het ontwerp vermeld staan en toetst conform gangbare methoden of deze te zijn verwezelijken voordat hij start aan de realisatie.
- kiest een passende softwareontwikkelmethode.
- pakt zijn werkzaamheden op een ordelijke en systematische manier aan.
- signaleert en rapporteert tijdig over afwijkingen.

## Casus 1 - Enigma

### Maak een geheimschrift typemachine (Enigma)

- Maak een HTML-pagina met 1 invoerveld en 1 uitvoerveld.
- Een gebruiker kan tekst invoeren in het invoerveld.
- Schrijf een event handler die luistert naar welke toets wordt ingedrukt.
- Vervolgens wordt de volgende letter in het alfabet afgedrukt in het uitvoerveld. Bijvoorbeeld het woord AAP wordt dan BBQ en GLR wordt HMS.
- Het invoerveld en het uitvoerveld moeten leeggemaakt kunnen worden. Schrijf hier een aparte functie voor die wordt aangeroepen wanneer er op een "reset" knop gedrukt wordt. Dit mag een event listener met een anonieme functie zijn, maar het mag ook een 'gewone' functie zijn.

Eisen:

- Zie de omschrijving hierboven.
- Er mag **NIET** gewerkt worden met `keyCode`. Deze is deprecated (wordt niet meer ondersteund door browsers).
- Het hoeft alleen maar te werken bij de letters uit het alfabet. Cijfers en leestekens doen niet mee.
- Wanneer de letter z ingetoetst wordt, moet de a geschreven worden.
- Het mag alleen maar werken wanneer de tekst in het invoerveld is ingevuld! Het is niet goed wanneer iemand naast het invoerveld klikt en toch iets kan invullen.
- Spaties moeten ook werken. Je mag zelf beslissen of dit spaties blijven of dat dit iets anders wordt.
- Maak een verslag met daarin:
  - Screenshot van je mapstructuur.
  - Een stukje tekst over je debug activiteiten.Zorg dat je hierin ten minste 2 screenshots van fouten opneemt en vertel hoe je deze fouten hebt opgelost.



The screenshot shows a web application titled "Mijn enigma". It has two main sections: "Invoer" (Input) and "Gecodeerde tekst" (Encoded text). The "Invoer" section contains a text input field with the text "hallo wereld" and a blue border. The "Gecodeerde tekst" section contains a text output field with the text "ibmmp xfsfme".

### OPDRACHT UITBREIDING 1 (maximaal 2 punten)

- Zorg dat de backspace ook werkt: de letter uit het gecodeerde veld wordt dan ook weggehaald.
- Cijfers en/of leestekens doen wel mee

## OPDRACHT UITBREIDING 2 (maximaal 1 punt)

- Maak ook een ontvangende pagina waarop iemand jouw code weer kan ontcijferen: De ontvanger kan de gecodeerde tekst dan simpelweg kopiëren en plakken in een veld en kan in een 2e veld de goede tekst lezen.

## HOE BEGIN JE?

- Maak een array met alle letters erin.
- Zoek uit hoe je kunt achterhalen op welke toets iemand heeft gedrukt.
- Let op: je zult nog een aantal variabelen nodig hebben. Bekijk goed waar je die declareert!
- Zoek op Internet hoe je kunt vergelijken of een ingevulde waarde overeenkomt met een item in de array.
- Toon vervolgens het volgende item uit de array in het uitvoerveld.
- Ga van daaruit verder met deze opdracht.

### Handige linkjes:

<https://keycode.info/>

## BEOORDELING

Je wordt beoordeeld op:

- Het is op tijd af.
- Je code is netjes.
- Je werkt netjes in aparte files.
- Je gebruikt variabelen.
- Je gebruikt functies.
- Je gebruikt event listeners.
- Je code is goed leesbaar; je gebruikt logische functienamen en voegt commentaar toe.
- Je hebt enige CSS-styling toegepast, zodat de site prettig werkt voor een gebruiker.
- Je volgt de code-conventie DRY (*Don't Repeat Yourself*) zo goed als mogelijk.
- Je gebruikt de console (of iets anders) om te debuggen.
- Extra punten kunnen verdiend worden door de uitbreidingen te maken.

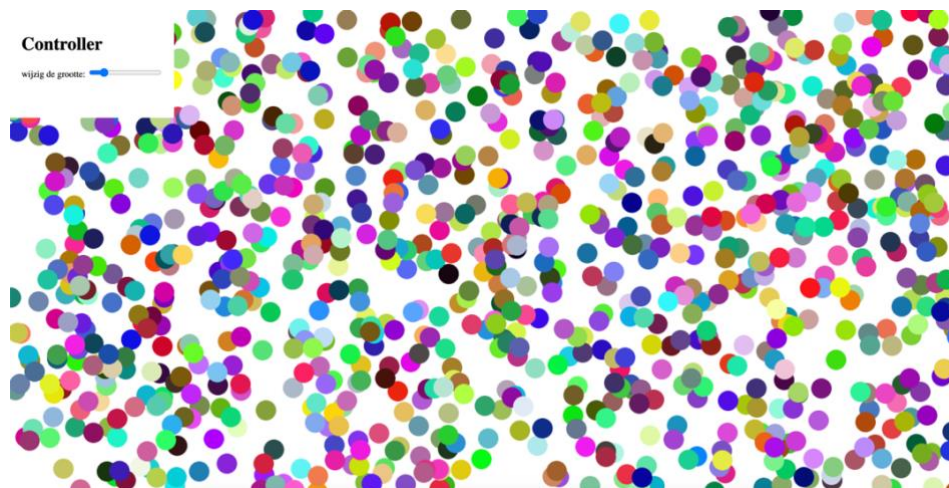


## Casus 2 - confetti

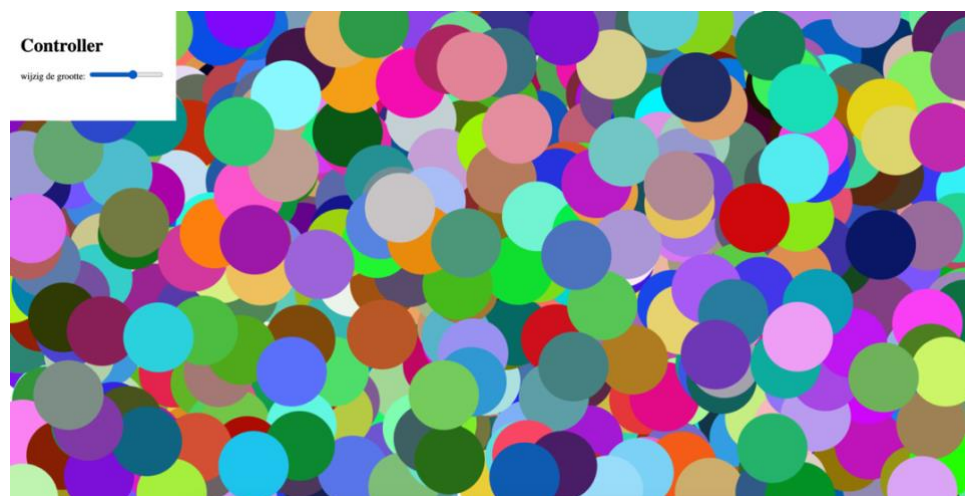
### Maak een confetti achtergrond

Maak een HTML-pagina met een canvas element er op. Dit canvas moet 100% breed en hoog zijn. In dit canvas komen 200 cirkels te staan. Kies zelf een kleur. Door middel van een schuifje (een range slider) kan een gebruiker de grootte van de cirkels wijzigen. Dit wordt dan direct doorgevoerd (de gebruiker hoeft dus niet meer op een knopje te drukken). Je hebt hier een event listener voor nodig.

Let op: de cirkels worden pas getekend NADAT de gebruiker de slider heeft gebruikt. Wanneer iemand voor het eerst op de pagina komt dan zal deze er nog leeg uitzien, dat is niet erg.



Figuur 1 - Confetti opdracht



Figuur 2 - Confetti opdracht met slider

## Eisen

- Er staan 200 cirkels op het canvas element.
- De cirkels staan op een random positie.
- Met behulp van een range slider is de grootte van de cirkels aan te passen.
- Bij het openen van de pagina hoeven er nog geen cirkels in het canvas te staan; dit hoeft pas te gebeuren nadat iemand de range slider heeft gebruikt.
- Je hebt gebruik gemaakt van een event handler.
- Maak een verslag met daarin:
  - Screenshot van je mappenstructuur.
  - Een stukje tekst over je debug activiteiten. Zorg dat je hierin ten minste 2 screenshots van fouten opneemt en vertel hoe je deze fouten hebt opgelost.

## OPDRACHT UITBEREIDING 1 (maximaal 2 punten)

- Maak een range slider waarmee je de radius van de cirkels instelt.

## OPDRACHT UITBEREIDING 2 (maximaal 1 punt)

- De cirkels hebben een random kleur.

## HOE BEGIN JE?

- Zorg eerst dat je 1 cirkel random in je canvas krijgt.
- Maak er dan 200.
- Zorg dat de range slider een event listener krijgt, waarbij de waarde van de range slider wordt opgevraagd.
- Vervolgens wordt binnen deze event listener de functie van de cirkels aangeroepen, waarbij de waarde van de range slider wordt meegegeven aan de functie.
- Ga van daaruit verder.

### Handige linkjes:

Over de range slider: [https://www.w3schools.com/howto/howto\\_js\\_rangeslider.asp](https://www.w3schools.com/howto/howto_js_rangeslider.asp)  
(zie stap 3 voor een voorbeeld)

Over de Math.random() functie:

[https://youtu.be/1Rq\\_LrpcgIM](https://youtu.be/1Rq_LrpcgIM) (EN).

<https://youtu.be/gTfjA95ONSQ> (EN). Tot circa 6:30.

## BEOORDELING

Je wordt beoordeeld op:

- Het is op tijd af.
- Je code is netjes.
- Je werkt netjes in aparte files.
- Je gebruikt variabelen.
- Je gebruikt functies.
- Je gebruikt event listeners.
- Je code is goed leesbaar; je gebruikt logische functienamen en voegt commentaar toe.
- Je hebt enige CSS-styling toegepast zodat de site prettig werkt voor een gebruiker.
- Je volgt de code-conventie DRY (*Don't Repeat Yourself*) zo goed als mogelijk.
- Je gebruikt de console (of iets anders) om te debuggen.
- Extra punten kunnen verdiend worden door de uitbreidingen te maken.

## Casus 3 - Selfie website met P5

### Beschrijving van de casus

Maak een website waarop iemand met behulp van de webcam een snapshot kan maken. Met behulp van 1 of meer knoppen kan er een (kleur)effect over de foto heen gelegd worden.

Zie <https://www.youtube.com/watch?v=oLiaUEKsRws> over het maken van een foto met behulp van de video. Zie [https://idmnyu.github.io/p5.js-image/Filters/vid\\_filter.html](https://idmnyu.github.io/p5.js-image/Filters/vid_filter.html) voor het toevoegen van een filter.

### Eisen

- Er is een knop waarmee een snapshot gemaakt kan worden.
- De video moet gestopt kunnen worden.
- De video moet gestart kunnen worden.
- Er kan tenminste 1 kleureffect over de foto heen gelegd worden.
- Maak een verslag met daarin:
  - Screenshot van je mappenstructuur.
  - Een stukje tekst over je debug activiteiten. Zorg dat je hierin ten minste 2 screenshots van fouten opneemt en vertel hoe je deze fouten hebt opgelost.

### OPDRACHT UITBEREIDING 1 (maximaal 2 punten)

- Er is meer dan 1 kleureffect mogelijk.

### OPDRACHT UITBEREIDING 2 (maximaal 1 punt)

- Doe op internet een onderzoek naar privacy.  
Bereid je verslag uit met het volgende:
  - Wat is dat eigenlijk, privacy?
  - Waarom zou je rekening moeten houden met de privacy?
  - Mag ik zomaar een opname maken van iemand met mijn website?
  - Wat kan ik doen wanneer ik wel opnames wil maken?

## BEOORDELING

Je wordt beoordeeld op:

- Het is op tijd af.
- Je code is netjes.
- Je werkt netjes in aparte files.
- Je gebruikt variabelen.

- Je gebruikt functies.
- Je gebruikt event listeners.
- Je code is goed leesbaar; je gebruikt logische functienamen en voegt commentaar toe.
- Je hebt enige CSS-styling toegepast, zodat de site prettig werkt voor een gebruiker.
- Je volgt de code-conventie DRY (*Don't Repeat Yourself*) zo goed als mogelijk.
- Je gebruikt de console (of iets anders) om te debuggen.
- Extra punten kunnen verdiend worden door de uitbreidingen te maken.

## Beoordelingsformulier casus

Module	Module 4: Javascript 3
Naam beoordelaar	
Naam student	
Datum	
Leerdoelen:	
De student kan een functie maken en aanroepen --> (wordt beoordeeld in alle opdrachten en casussen)	
De student kan variabelen binnen en buiten een functie gebruiken ('scope') --> (wordt beoordeeld in verwerkingsopdracht 2-1 en iedere casus)	
De student kan parameters en returnwaarden gebruiken in een functie --> (wordt beoordeeld in alle opdrachten en casussen)	
De student kan een functie gebruiken als event-handler --> (wordt beoordeeld in verwerkingsopdracht 3-1)	
De student kan een library gebruiken --> (wordt beoordeeld in verwerkingsopdracht 4-1 en in casus 3)	
De student kan fouten in eigen code opsporen ('debuggen') --> (wordt beoordeeld in alle opdrachten en casussen)	
Opmerkingen:	
Normering:	Score:
Leerdoelen voldoende	7
Uitbereiding 1 van de casus gemaakt	2 (maximaal)
Uitbereiding 2 van de casus gemaakt	1 (maximaal)
<i>Bij te laat inleveren</i>	<i>Maximum cijfer: 6</i>
<i>Niets inleveren</i>	<i>Cijfer: 1</i>
Cijfer:	

<b>Werkproces</b>	<b>O/V</b>
B1-K2-W1: Realiseert (onderdelen van) een product	
Feedback:	