



# Handboek

Red de wereld  
met code!

in P5

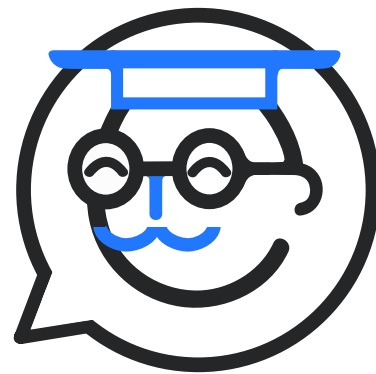
# Introductie

Leuk dat je meedoet aan de workshop programmeren! Tijdens deze workshop ga je de basis leren zodat je straks helemaal zelf kunt gaan programmeren. De vaardigheid die we hiermee trainen is probleemoplossend denken. Dit is een belangrijke vaardigheid in je toekomst! Het is uiteindelijk de bedoeling dat je een anti-virus gaat programmeren voor het wereldwijde corona virus. Allereerst ga je leren hoe een assenstelsel werkt en waarom dit van belang is bij het programmeren. Hierna ga je de basisvormen leren die bij programmeren gebruikt worden en hoe je deze kan kleuren. Als het goed is heb je de tutorial video's al bekeken!

Je gaat dus uiteindelijk een instructie schrijven in programmeertaal die de computer moet begrijpen. Een van de belangrijkste dingen om te onthouden bij programmeren is om niet op te geven! Soms lukt het even niet, maar je moet blijven uitproberen om erachter te komen hoe alles werkt.

We gaan straks werken in de online editor genaamd P5.js. Dat programma klinkt heel ingewikkeld maar het is eigenlijk gewoon een moeilijke naam voor een tekenprogramma in de browser.

Het is vandaag de bedoeling dat je creatief bezig gaat zijn!



Heel veel plezier!

# Ontdek het assenstelsel

Als het goed is heb je de tutorial videos bekeken. Hierin wordt uitgelegd hoe een assenstelsel werkt. Om deze kennis even op te frissen volgt er een korte opdracht.

Zoek op welke coördinaten de punten **a** t/m **d** hebben. Zet vervolgens de punten **e** t/m **h** in hetzelfde assenstelsel (deze punten maken geén vorm maar is puur bedoeld om te oefenen).

a( , )

b( , )

c( , )

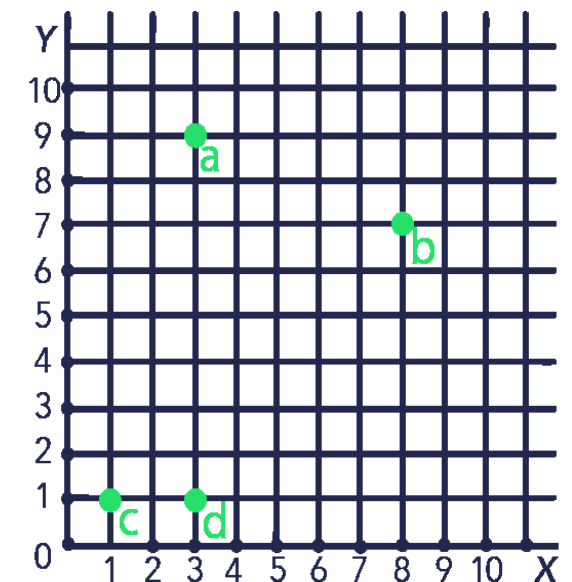
d( , )

e(1,10)

f(10,1)

g(6,5)

h(1,7)



Bij programmeren is het assenstelsel zeer belangrijk. Zoals je misschien al weet ga je hierin alles een plek geven. Nu hebben we steeds gewerkt met hokjes. Bij programmeren werkt dit met zogenaamde pixels. Je scherm is helemaal opgebouwd uit kleine blokjes die door rood, groen en blauw te mengen een bepaalde kleur kunnen maken. Meerdere van deze blokjes bij elkaar vormen samen bijvoorbeeld een letter of een afbeelding. Bij oudere beeldschermen kan je de pixels vaak met het blote oog zien. Kijk maar eens goed als je er een tegenkomt!

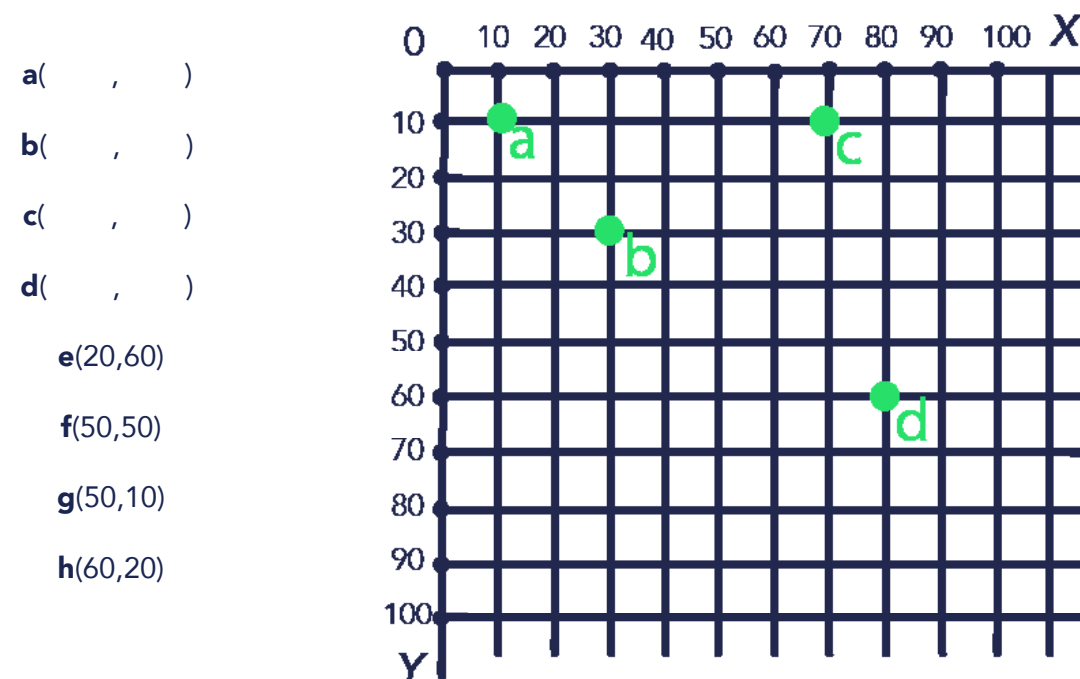
Bij het programmeren ga je in het assenstelsel aangeven waar die blokjes(pixels) neer komen te staan. Het is belangrijk om te weten dat bij het programmeren de oorsprong van het assenstelsel niet van linksonder begint, maar linksboven. Als je dus een punt wilt aangeven in programmeertaal begin je vanuit boven te tellen.

# Assenstelsel in programmeertaal

De x-as staat dus niet onderaan het raster, maar erboven. Wel nog steeds horizontaal. Als je het x coördinaat wilt weten tel je dus vanaf links naar rechts. Om het y coördinaat te vinden moet je van boven naar onder tellen. Je begint ook linksboven.

Het is dus een kwestie van omdenken. In plaats van onder naar boven te tellen op de y-as begin je bovenaan en tel je naar onder toe. In de volgende opdracht kun je hiermee oefenen.

Zoek op welke coördinaten de punten **a** t/m **d** hebben. Zet vervolgens de punten **e** t/m **h** in het assenstelsel. De stappen(cijfers) tussen de hokjes zijn dit keer groter. Laat je hierdoor niet in de war brengen.



Je weet nu hoe je een punt precies kunt aangeven binnen het assenstelsel van pixels. De volgende stap is om niet één punt te kunnen teken op een bepaalde positie, maar een hele vorm.

# Basisvormen

Hieronder zie je een aantal vormen met de benaming die gebruikt worden in het programma P5.js.



Rechthoek = **rect()**;



Cirkel = **ellipse()**;



Driehoek = **triangle()**;



**Let op!**

Een **ellipse** wordt vanuit het midden van de cirkel getekend.  
Een **rectangle** wordt vanuit de linkerbovenhoek getekend.  
Hier leggen we zometeen meer over uit.



**Wees creatief!**

# Rechthoek

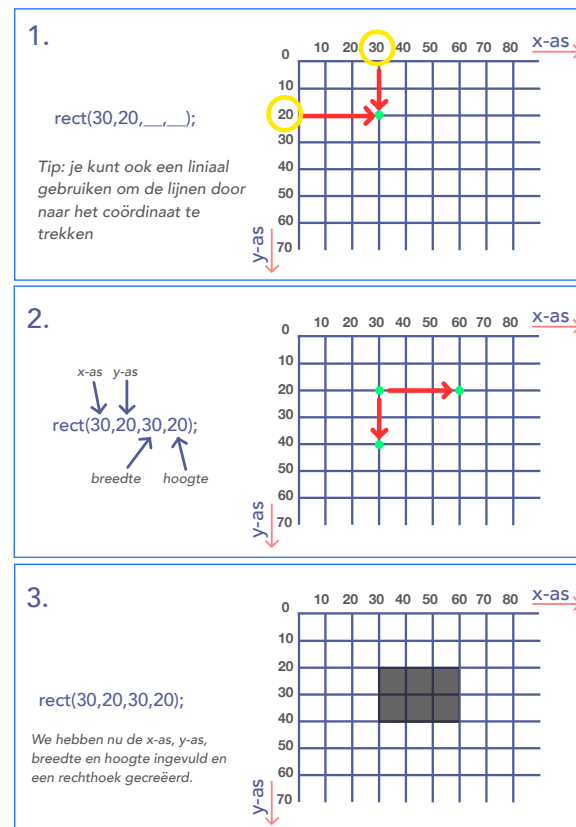
We gaan de vormen ontdekken die je kunt programmeren in het programma P5.js, dat programma klinkt heel ingewikkeld maar het is eigenlijk gewoon een moeilijke naam voor een tekenprogramma in de browser! Allereerst is het belangrijk om nog even te herinneren dat bij programmeren het nulpunt van het assenstelsel altijd linksboven staat.

Er zijn een aantal basis vormen die je gaat leren. Dit zijn een rechthoek, cirkel en een driehoek. Allereerst beginnen we met de rechthoek. De rechthoek in programmeertaal noemen we een rectangle en schrijf je zo: **rect()**; Tussen de haakjes geven we aan waar de vorm moet komen te staan en hoe groot de vorm moet worden. Bij een rechthoek moet je 4 cijfers invullen, namelijk de x-as, y-as, breedte en hoogte. Voorbeeld: **rect(x,y,breedte,hoogte)**; Het is belangrijk dat tussen elke waarde een komma staat en dat de code wordt afgesloten met een punt komma, anders werkt hij niet! Je kunt altijd de coördinaten nog wijzigen in P5 wanneer je een foutje hebt gemaakt.

De rechthoek wordt getekend vanuit de linkerbovenhoek van de rechthoek. Dit betekent dat je de x-as en y-as moet invullen van deze hoek. Dus allereerst zorgen we dat de we de coördinaten van de linkerbovenhoek hebben. Vanuit dit punt wordt de breedte en hoogte getekend.

Hiernaast zie je een voorbeeld met de stappen hoe dit werkt.

Je kunt deze voorbeelden ook als spiekblaadje gebruiken tijdens het programmeren waar je zometeen mee aan de slag gaat! :)



# Cirkel

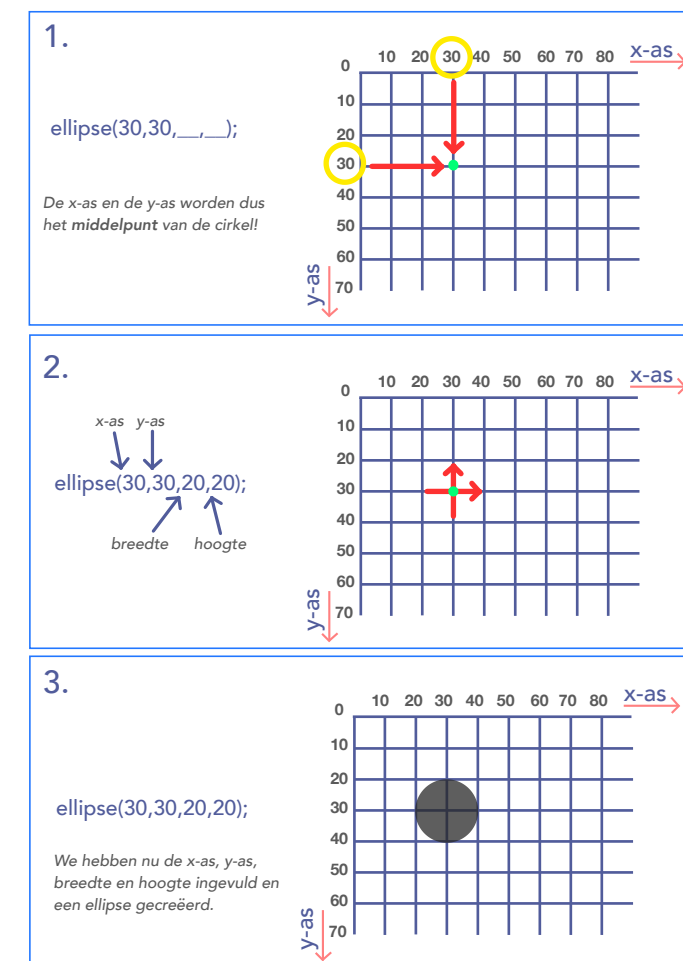
Hierna volgt de cirkel vorm. Deze noemen we in programmeertaal een ellipse en schrijven we zo: **ellipse()**;

Bij een ellipse zijn er ook 4 cijfers nodig tussen de haakjes.

Alleen wordt bij de ellipse de x-as en y-as vanuit het midden getekend.

Vanuit hier bepaal je dus de coördinaten, breedte en hoogte van de cirkel. De code van de ellipse ziet er hetzelfde uit als de rechthoek:

**ellipse(x,y,breedte,hoogte);**



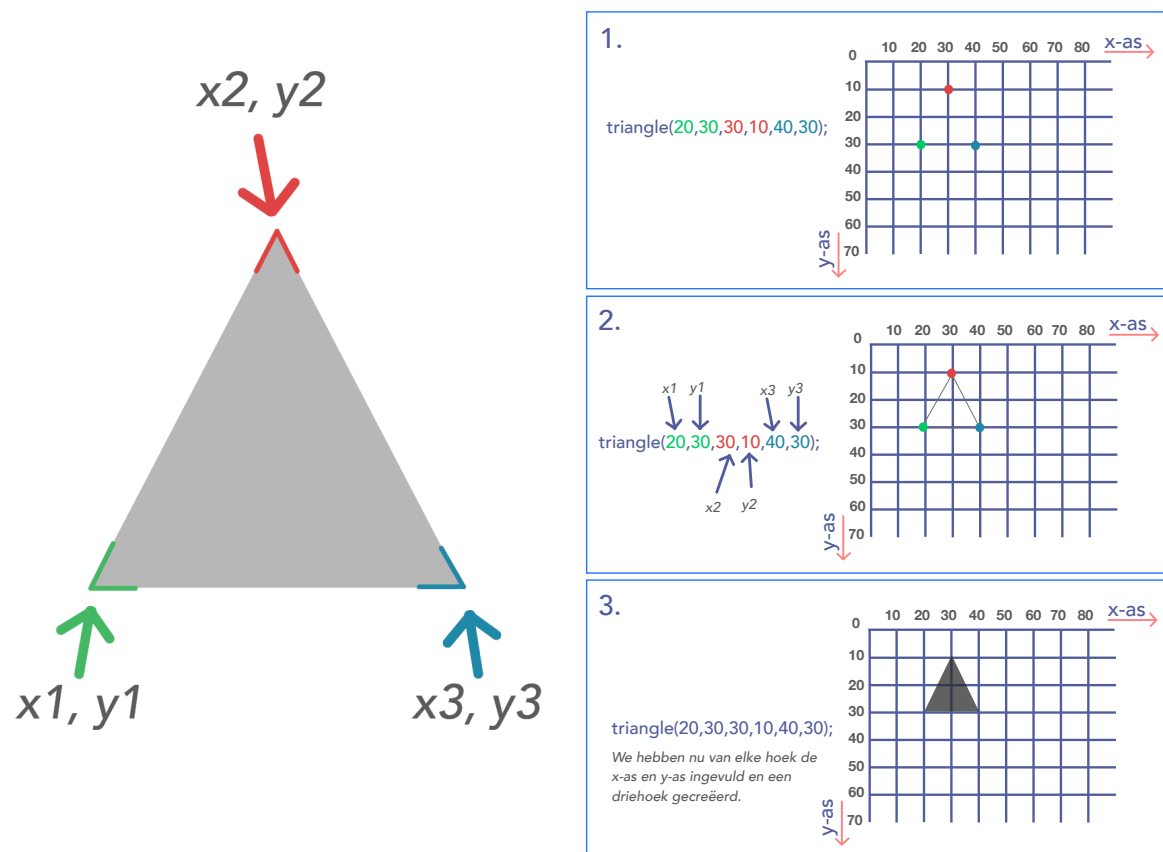
# Driehoek

Dan hebben we ook nog de driehoek. Deze noemen we in programmeertaal een triangle en schrijven we zo: **triangle();**

De driehoek is wat lastiger te maken dan de andere vormen. Deze is namelijk opgebouwd uit 6 waardes in plaats van 4. We schrijven de triangle in code als volgt: **triangle(x1,y1,x2,y2,x3,y3);**

Zoals je kunt zien staat bij de triangle geen breedte of hoogte. Dat komt omdat de driehoek wordt gemaakt vanuit 3 losse punten. x3,y3 staan voor het derde punt.

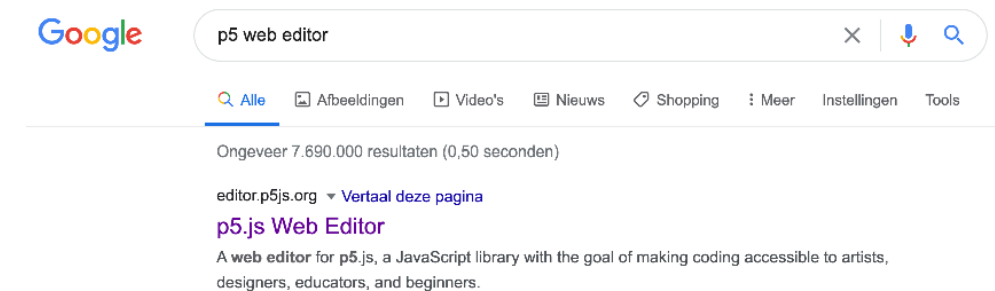
Hieronder kun je zien waar de punten zich bevinden. De x3 en y3 bevinden zich dus bijvoorbeeld bij de rechterhoek beneden



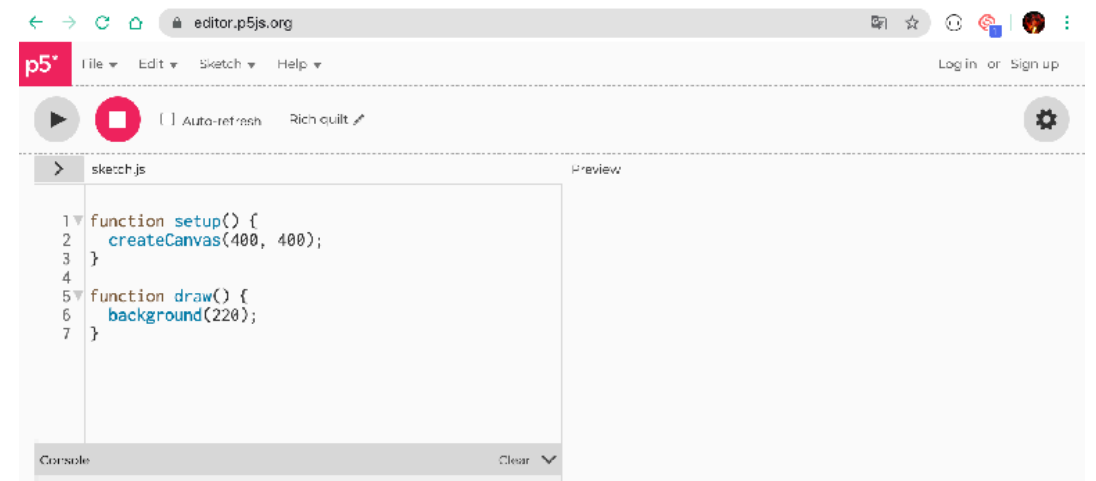
# Werken in P5

Tot nu toe heb je steeds losse stukjes van een code geschreven. Jullie hebben uitleg gekregen over de coördinaten in het assenstelsel en de termen voor vormen. Ook weten jullie nu hoe de code van deze vormen in elkaar zit. Al deze onderdelen gaan we nu samenvoegen om een volledige code te kunnen schrijven. Hier voor gaan we P5 web editor gebruiken zoals eerder benoemd.

Als je laptop opgestart is open je Google Chrome of Internet Explorer. Typ in de zoekbalk van Google 'P5 web editor' in en druk op enter. Als het goed is krijg je het volgende te zien(zie onderstaand voorbeeld). De eerste link die tevoorschijn komt is als het goed is **www.editor.p5js.org/**, deze open je.



Als het goed is krijg je na het openen van de link het volgende in beeld.



# Werken in P5

Aan de linker kant komt je code te staan en aan de rechter kant kun je zien wat je maakt. Eerst gaan we ontdekken wat er aan de linkerkant staat, dit is namelijk belangrijk want hier komt je code te staan. Als het goed is zie je daar *setup* en *draw* staan, je ziet een *play* en *stop* knop. De setup voor nu is niet zo heel belangrijk maar kort samengevat: hier komen dingen in die je een keer wil creëren en die niet veranderen zoals de createcanvas. De createcanvas is je werkveld en deze kan je altijd groter of kleiner maken, maar voor nu is de standaard grootte van 400 bij 400 pixels perfect. Als je op de play knop klikt zal aan de rechterkant een grijs vierkant tevoorschijn komen.

Dan komen we aan bij de functie draw. Deze is heel belangrijk. Deze functie is waar we nu voornamelijk in zullen gaan werken. Hier komt de code in voor alles wat je wilt maken. Voordat we beginnen zijn er een paar dingen belangrijk om te weten en te herhalen:

- Tijdens het programmeren werkt het assenstelsel van boven naar beneden zoals eerder al werd verteld. Alles wordt gecreëerd vanuit de linkerbovenhoek.
- De grootte van je werkveld is ook de grootte van je assenstelsel. Je kan dus op 400 punten iets neer zetten. Zowel horizontaal (x) als verticaal (y).
- Je ziet bij de draw 2 van deze haakjes {} en je code moet tussen deze haakjes staan anders werkt de code niet. In ons geval staat er al een background dus moet je daaronder je code schrijven.
- De code die je schrijft begint altijd met een object zoals een rect(rechthoek), hier komen 2 haakjes achter te staan () en daar komt een punt komma achter.
- Tussen de haakjes () van je regel code komen de waarden van de positie, breedte en hoogte te staan.
- Tussen waardes van je code moet een komma.  
Voorbeeld `rect(120,80,40,40);`



**Wat zijn jullie al goed op weg!**

# Oefenopdracht in P5

Om alvast wat handigheid in het programmeren en P5 te krijgen gaan we wat opdrachtjes maken!

**Opdracht 1:** Schrijf de code voor een rechthoek.

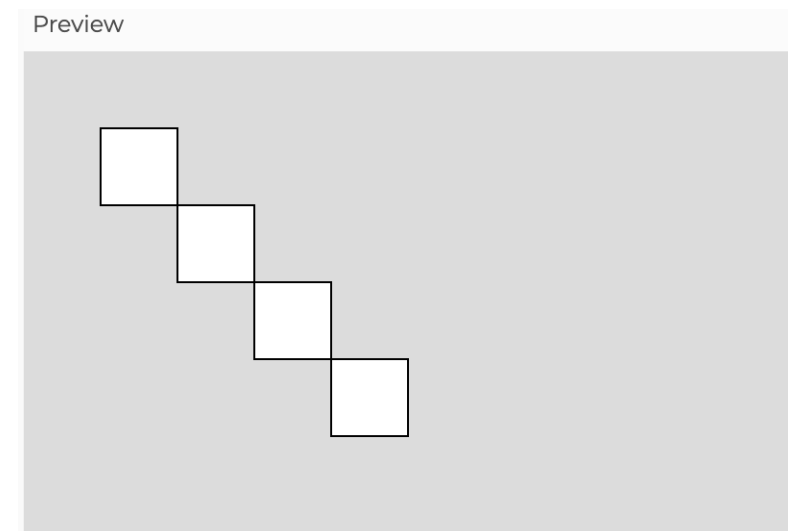
**Opdracht 2:** Vul de code van de rechthoek met coördinaten(tussen de haakjes). Het maakt niet uit waar de rechthoek in je canvas komt te staan.

**Opdracht 3:** Schrijf nu de grootte op de juiste plaats in de regel van de code, zorg dat je de rechthoek 40 breed en 40 hoog wordt (een vierkant dus).

**Opdracht 4:** schrijf nu de code voor een vierkantje op 2 andere plaatsen binnen je canvas.

*Extra opdracht:* probeer dit nu ook voor de driehoek en de cirkel.

Tip: als het nog niet wil lukken kijk dan nog eens terug naar de uitleg van basisvormen! Hieronder zie je een voorbeeldje van wat je zou kunnen maken in P5.





# Kleuren maken

We hebben nu wat vierkantjes gemaakt maar ze zijn allemaal wit, en dat is best saai. Daarom gaan we kleur toevoegen. Om dit te kunnen doen moeten we deze vierkantjes vullen met kleur.

Dat kunnen we doen met een code, en deze noemen we: **fill()**. Nu zou je denken dat deze **fill()** onder de regel code komt die we zojuist hebben geschreven. Maar nee dat is niet het geval.

Code werkt van boven naar beneden net als het assenstelsel. Dus de vulling van die vierkant komt boven de code van de vierkant zelf te staan. Ga maar na als je een glazen bak wil vullen met rode verf. Dan hou je bus verf er ook boven. Dat is bij code ook zo. Maar waar vullen we de vierkantjes mee? We vullen die met kleur. In programmeren wordt gewerkt met **RGB**. Dit staat voor **Red**, **Green** en **Blue**. Oftewel Rood, Groen en Blauw. We kunnen vrijwel alle kleuren maken die we willen met deze drie kleuren. Om een bepaalde kleur te krijgen moeten we deze van een waarde voorzien. De waarde loopt van 0 tot 255. Dus als we de kleur rood willen krijgen dan vullen we rood met 255, groen met 0 en blauw met 0. 255 staat dus puur voor rood. Bij groen zeggen we 0, dus geen groen. Dit geldt hetzelfde voor blauw. Hieronder zie je de kleurcodes de bij de kleuren horen.

Rood = (255,0,0);

Groen = (0,255,0);

Blauw = (0,0,255);



**Tip: gebruik dit later als spiekblaadje!**

# Kleuren in P5

Nu jullie de kleuren weten kunnen we een vierkantje maken en aan de slag gaan. Je mag dit stukje code overnemen in de functie draw, of je mag zelf 3 vierkantjes tekenen.

```
fill(255,255,255);  
rect(40,160,40,40);
```

```
fill(255,255,255);  
rect(200,80,40,40);
```

```
fill(255,255,255);  
rect(280,280,40,40);
```

Let op: dit komt onder de *background* te staan. Om dit makkelijk te doen klik je met de muis achter de code van background en druk je 2 keer op enter. Als je nu weer op play klikt zal je zien dat je 3 witte vierkantjes hebt. Hieronder volgen een aantal opdrachtjes:

**Opdracht 1:** We gaan nu proberen om de vierkantjes in te kleuren. Maak het eerste vierkantje rood, de tweede groen en de derde blauw. Kijk nog even terug naar de uitleg als het nodig is.

**Opdracht 2:** Probeer nu om het eerste vierkantje paars te maken. Denk nog eens na over welke kleuren je nodig hebt en waar die moeten staan. Bij de waarde 0 krijg je geen kleur. 255 geeft je de volledige kleur.

**Opdracht 3:** Probeer nu geel te maken, hiervoor heb je rood en nog een kleur nodig. `fill(0,groen,0);`

**Opdracht 4:** Probeer nu lichtblauw te maken, hiervoor heb je alleen groen en nog een kleur nodig. `fill(000,groen,000);`

*Extra opdracht:* Probeer nu zelf wit en zwart te maken. Hier heb je alle drie de kleuren voor nodig, denk er aan dat je waardes van 0 tot 255 kan gebruiken.

# Tekst in P5

Ondertussen kunnen we meerdere vormen maken en die een kleur geven maar we kunnen nog meer doen met programmeren. Je kan ook tekst invoegen met code. Helaas kan je niet gewoon een stukje tekst in typen. Je hebt een net zoals bij vormen een bepaalde functie voor nodig. Hiervoor heb je de text functie. Deze bestaat ook weer uit drie delen. tekst, 2 haakjes (), en een punt komma. Voorbeeld: **text();**

Hier komt wel wat extra's bij kijken. Je moet tussen de 2 haakjes nog 2 aanhalingsteken plaatsen. Als je dat niet doet wordt je tekst niet weer gegeven. Voorbeeld: **text('tekst', x, y);**

Tekst kan je op verschillende manieren op maken, maar voor nu zullen we het vrij simpel houden. We gaan de tekst voorzien van een kleur en een lettergrootte. Tekst kan op meerdere manieren worden opgemaakt. We kunnen zowel kleur als lettergrootte gebruiken. Zoals we zojuist zagen kwam de kleur vóór de code van een vorm te staan. Dit geldt ook voor tekst.

Lettergrootte is iets moeilijker. Deze moet natuurlijk ook voor de tekst staan. We gaan de tekst grootte aanpassen en dit doen we met de code: **textSize();**. Let even goed erop dat je text met kleine letters schrijft en size met een hoofdletter en zonder spatie. Nogmaals er komen 2 haakjes () achter en een punt komma ;. Dus: **textSize();**



# Opdrachten tekst

**Opdracht 1:** Neem de volgende code over en zet op de juiste plek je eigen naam. **text("",x,y);**

**Opdracht 2:** Pas nu de x en de y aan zodat de tekst precies in het midden komt te staan.

**Opdracht 3:** Geef de tekst een kleur naar keuze.

**Opdracht 4:** Voeg de regel code voor de tekstgrootte toe. Kijk nog even goed naar de voorbeelden in de uitleg.

Jullie hebben nu alles geleerd wat je moet weten om het anti-virus te maken. Natuurlijk kun je het anti-virus maken zoals je zelf wilt, maar dit is veel informatie en we willen natuurlijk dat het antivirus ook echt goed werkt dus geven jullie een steuntje in de rug. Je kunt het stappenplan gebruiken ter ondersteuning. Wees creatief en voeg op het einde nog meer cirkels toe als je dat wilt, of andere kleuren. Misschien wil je hem wel een naam geven. Veel plezier! :)

