
Figure 1: 1: Tekenen

# Contents



# Voorwoord


Figure 1: Het logo van De Jonge Onderzoekers


Figure 2: Het logo van Codestarter

Dit is het Processing boek van de Dojo. Processing is een programmeertaal. Dit boek leert je die programmeertaal.

### Over dit boek



Figure 3: De licensie van dit boek



Het is nog een beetje een slordig boek. Zo staat bijvoorbeeld het plaatje dat eigenlijk op de kaft moet staan op pagina twee. Er zitten tiepvauten in en de opmaak is n`iet altij`d *even moo*i.

Daarom staat dit boek op een GitHub. Om precies te zijn, op https://github.com/richelbilderbeek/Dojo . Hierdoor kan iedereen die dit boek te slordig vindt minder slordig maken.

## Processing opstarten op cursuslaptop

We programmeren in Processing. Processing is een programma. Dit programma moeten we dus eerst opstarten.

### Terminal starten

Eerst moeten we een terminal starten.

Wij hebben twee soorten laptops:

- Debian (spreek uit 'Debie-jen')
- Lubuntu (spreek uit 'Loe-boen-toe')

Start een Terminal:

- Debian: druk op de `WIN` toets (deze zit linksonder, tussen CTRL en ALT). Type dan 'terminal' en dan ENTER
- Lubuntu: druk op `CTRL + ALT + T`

### Processing starten

In de terminal, type:

```
cd Programs/processing-3.2.1
```

En type dan:

```
./processing
```

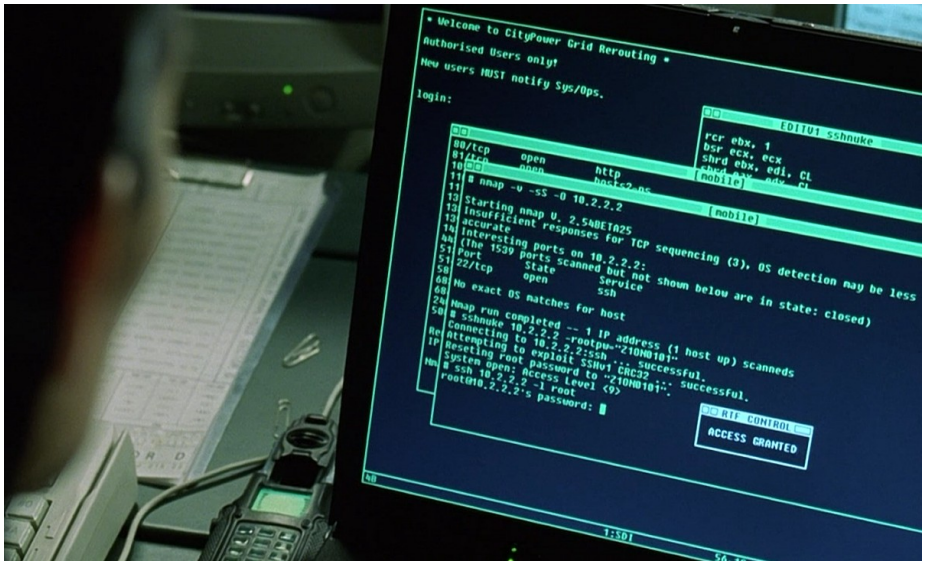


Figure 4: Hackers werken met een terminal

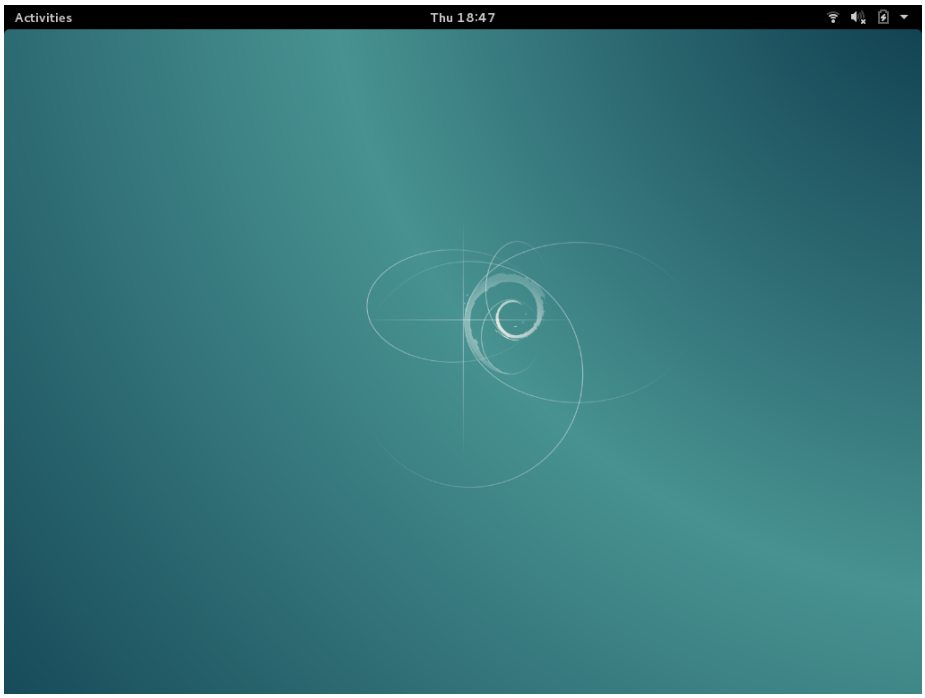


Figure 5: Debian: druk op WIN


Figure 6: De WIN toets zit tussen CTRL en ALT

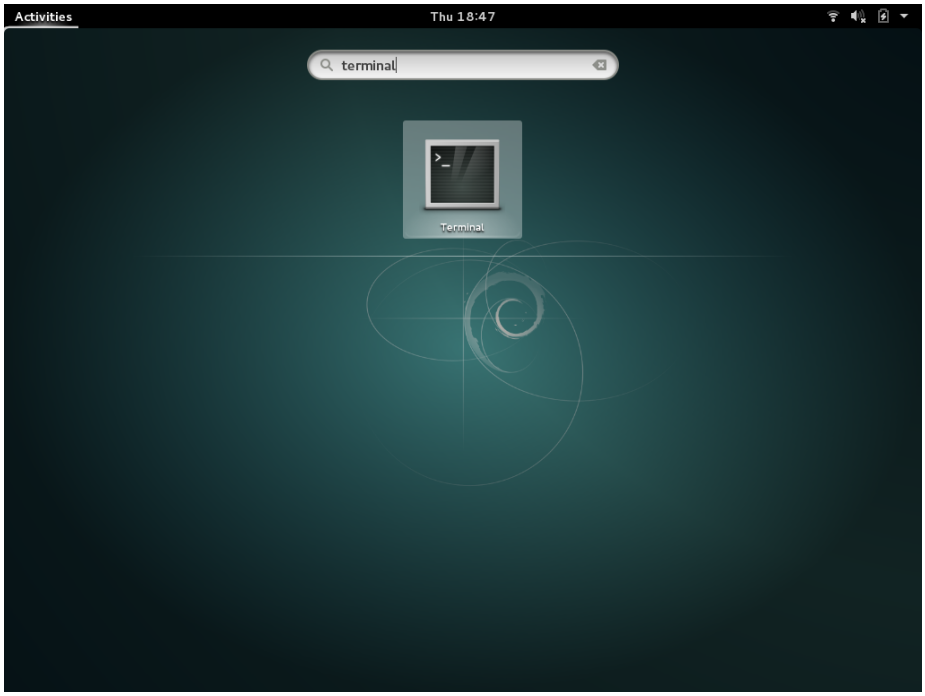


Figure 7: Debian: druk op WIN en type dan ‘terminal’

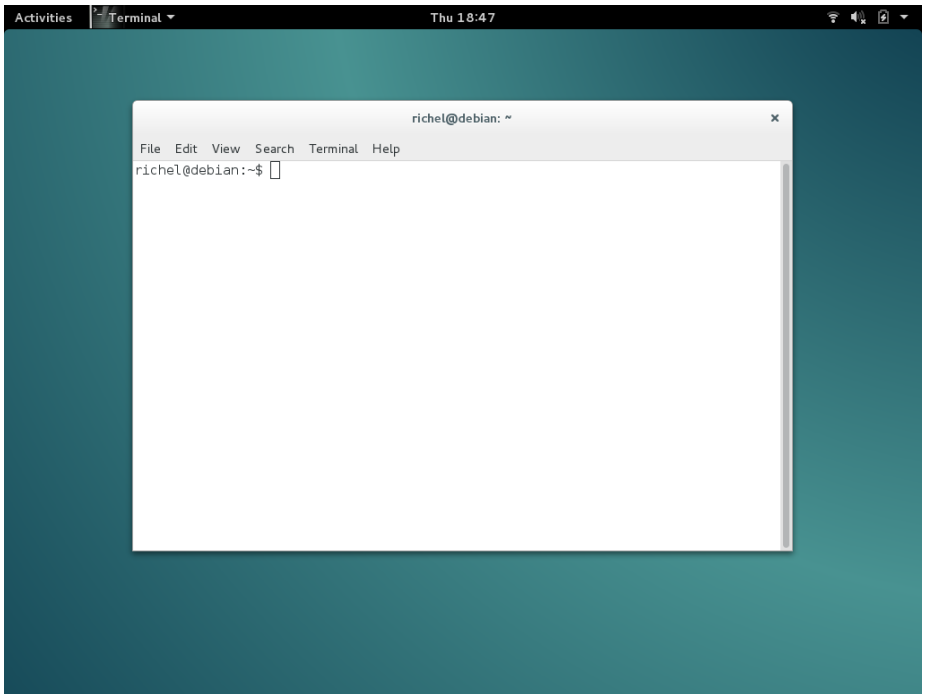


Figure 8: Debian: een terminal


Figure 9: Lubuntu: een terminal

Klaar!

## Processing slimmer starten

Je hoeft niet alles helemaal te typen. Een terminal kan je ook helpen. Als je in een terminal op Tab drukt, maakt deze het woord af.

In de terminal, type:

`cd Progr`

en druk dan op 'Tab'.

Nu wordt staat er op je scherm:

`cd Programs/`

Maak ervan

`cd Programs/pr`

en druk dan op 'Tab'.

Nu staat er:

`cd Programs/processing-3.2.1`

druk dan op `Enter`.

Doe nu:

`./p`

en druk dan op `Tab`. Nu staat er:

`./processing`

Druk op `Enter`.

Klaar!

## Updaten

Het kan gebeuren dat er een venster komt met een Update. Het venster vraagt of je een nieuwe versie van Processing wilt installeren. Klik op 'No'. 'No' is Engels voor 'nee'.

## Eindppdracht

- Start Processing helemaal zonder hulp

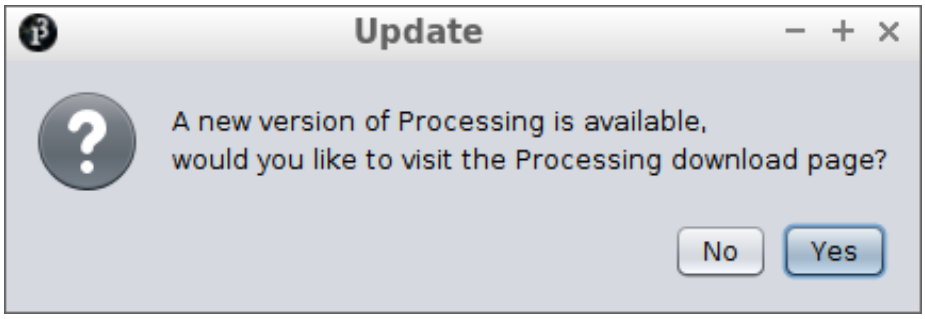


Figure 10: Processing Update venster, klik op 'No'

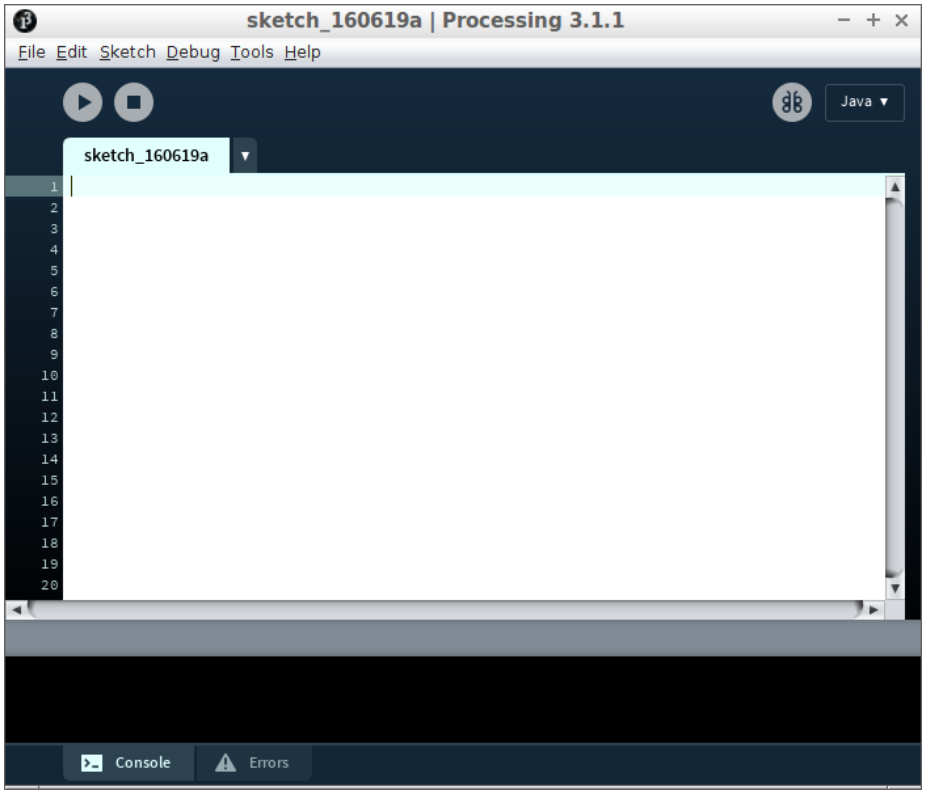


Figure 11: Processing zonder code

# Een Mooi Programma

Processing is een programmeertaal ontwikkeld voor kunstenaars en erg geschikt om games en mooie dingen mee te maken.

In deze les gaan we leren

- hoe we Processing opstarten
- hoe je code naar Processing kopieert
- hoe je het programma start

Zo ziet het programma eruit:


Figure 12: EenMooiProgramma

## Wat je nodig hebt

Je moet Processing op kunnen starten. Hoe dat moet, hangt af van het besturingssysteem:

- Processing opstarten op cursus laptop

- Processing installeren op eigen laptop met GNU/Linux
- Processing installeren op eigen laptop met Windows

## Code kopieeren

Processing begint met een leeg programma zonder code:

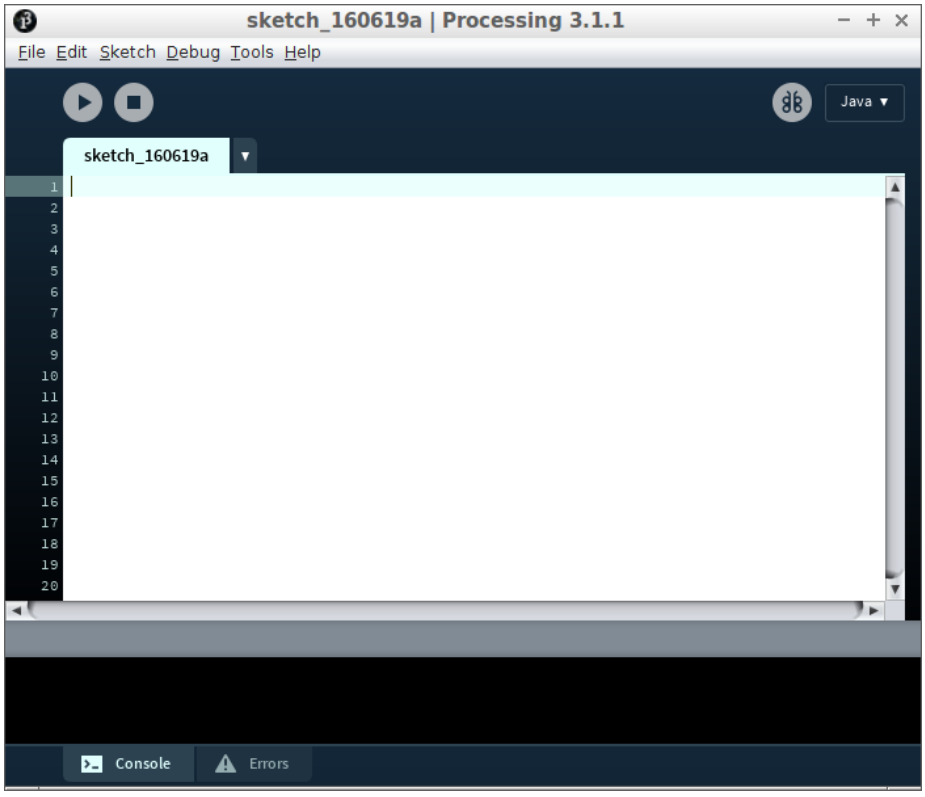


Figure 13: Processing zonder code

Dit is de programmeercode die we gaan gebruiken:

```
void setup()
{
  size(256,256);
}

void draw()
{
  fill(mouseX, mouseY, mouseX + mouseY);
  ellipse(mouseX, mouseY, 50, 50);
```

```
  fill(mouseY, mouseX, 255);
  ellipse(mouseY, mouseX, 50, 50);
}
```

Wat de code precies doet, leggen we later uit. Voor nu is het genoeg te weten dat het iets moois doet.

Om code te kopieeren gebruik je sneltoetsen:

- `SHIFT + pijltjes`: selecteren
- `CTRL + A`: alles selecteren
- `CTRL + C`: kopieren van selectie
- `CTRL + X`: knippen van selectie
- `CTRL + V`: plakken van selectie
- Start Processing
- Kopieer deze code naar Processing

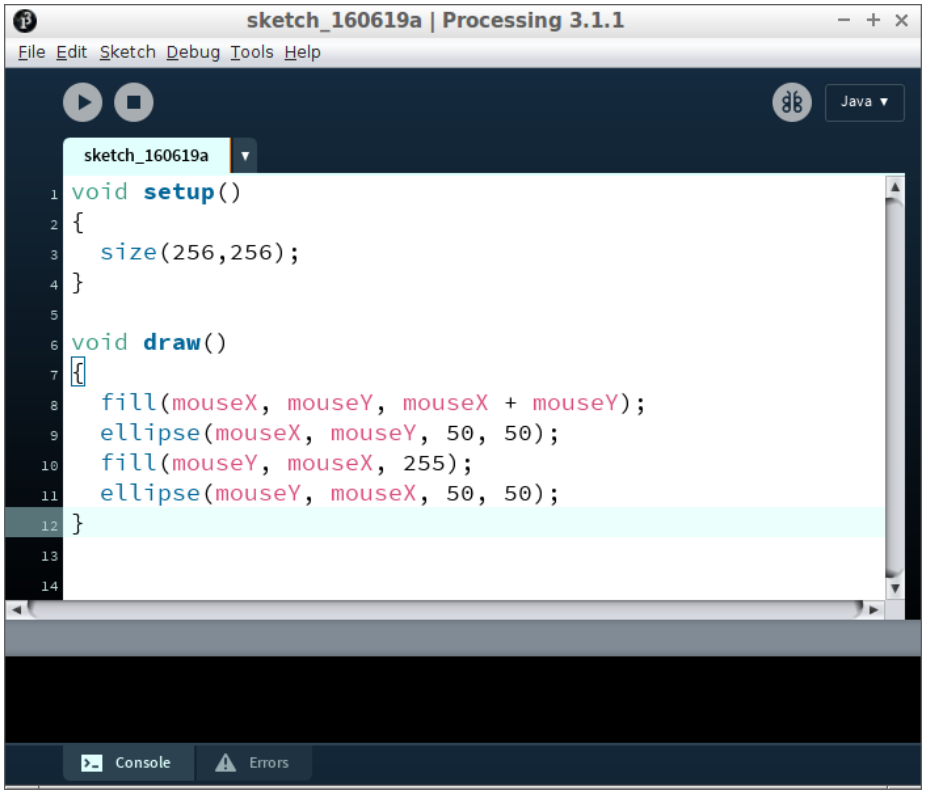


Figure 14: Processing met code

## Programma uitvoeren

- Klik op de 'Run' knop

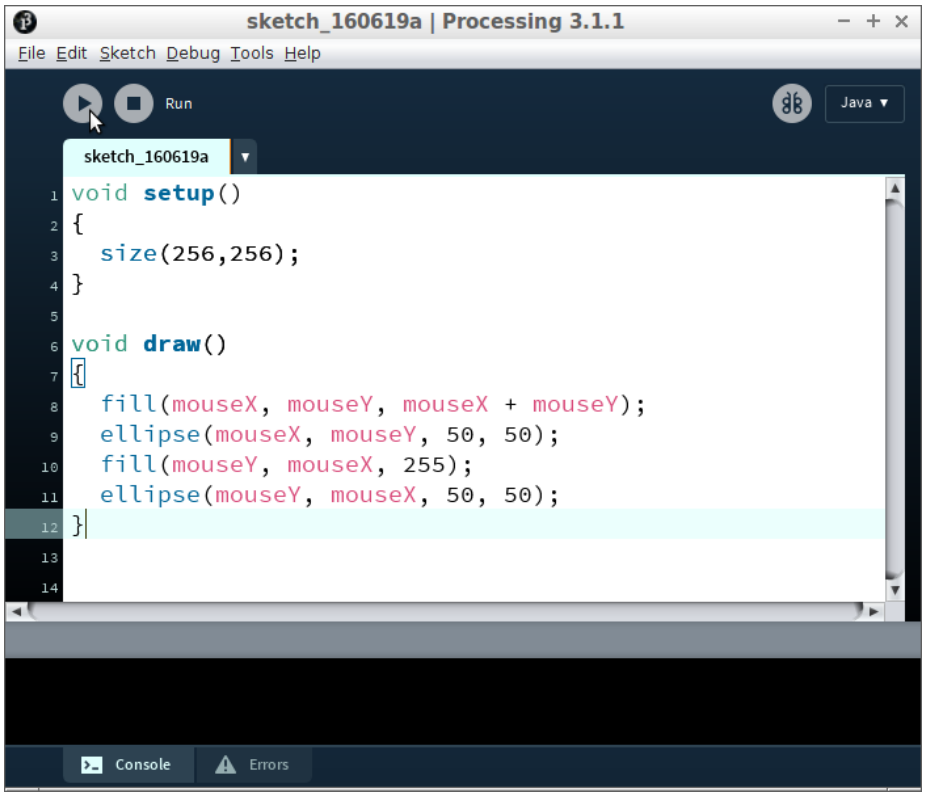


Figure 15: De Run knop

Als het goed is, zie je nu het programma!

## Sneltoetsen oefenen

- Werk met iemand samen. Hussel de code van de andere door de war, door deze te knippen/kopieren en te plakken. Repareer dan de code op je eigen computer

# `point`

Processing is een programmeertaal ontwikkeld voor kunstenaars en erg geschikt om games en mooie dingen mee te maken.

In deze les gaan we leren

- wat pixels zijn
- hoe de pixels op een beeldscherm zitten
- hoe je puntjes tekent

## Pixels

`Pixel = een vierkantje op je beeldscherm`

Pixels zijn de vierkantjes waaruit je beeldscherm is opgebouwd. Hoe meer pixels je scherm heeft, hoe scherper het beeld eruit ziet. Dat zie je goed bij oude games: die hebben minder pixels:

## Coordinaten

Elke pixel op je beeldscherm heeft een soort adres. Dit adres noemen we een coordinaat. Een coordinaat bestaat uit twee getallen.

`Coordinaat = een plek`

Van een coordinaat zijn de twee getallen:

- eerste getal: hoeveel pixels naar rechts
- tweede getal: hoeveel pixels naar onder

De pixel linksbovenin heeft als coordinaat `(0,0)` (spreek uit 'nul komma nul'). Dat klopt: je gaat nul naar rechts, dan nul naar onder!

Plaatje `Keiveel coordinaten` laat keiveel coordinaten zien.

## Een puntje tekenen

We gaan een puntje tekenen op (2,1). We maken het scherm 4 pixels breed en 3 pixels hoog.

Dat ziet er als tekening zo uit:

Hier zie je de code:

```
void setup()
{
  size(4, 3);
}

void draw()
{
```

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| (0,0) | (1,0) | (2,0) | (3,0) | (4,0) | (5,0) | (6,0) | (7,0) | (8,0) | (9,0) |
| (0,1) | (1,1) | (2,1) | (3,1) | (4,1) | (5,1) | (6,1) | (7,1) | (8,1) | (9,1) |
| (0,2) | (1,2) | (2,2) | (3,2) | (4,2) | (5,2) | (6,2) | (7,2) | (8,2) | (9,2) |
| (0,3) | (1,3) | (2,3) | (3,3) | (4,3) | (5,3) | (6,3) | (7,3) | (8,3) | (9,3) |
| (0,4) | (1,4) | (2,4) | (3,4) | (4,4) | (5,4) | (6,4) | (7,4) | (8,4) | (9,4) |
| (0,5) | (1,5) | (2,5) | (3,5) | (4,5) | (5,5) | (6,5) | (7,5) | (8,5) | (9,5) |
| (0,6) | (1,6) | (2,6) | (3,6) | (4,6) | (5,6) | (6,6) | (7,6) | (8,6) | (9,6) |
| (0,7) | (1,7) | (2,7) | (3,7) | (4,7) | (5,7) | (6,7) | (7,7) | (8,7) | (9,7) |
| (0,8) | (1,8) | (2,8) | (3,8) | (4,8) | (5,8) | (6,8) | (7,8) | (8,8) | (9,8) |
| (0,9) | (1,9) | (2,9) | (3,9) | (4,9) | (5,9) | (6,9) | (7,9) | (8,9) | (9,9) |

Figure 16: Keiveel coordinaten

| | | | |
|---|---|---|---|
| (0,0) | (1,0) | (2,0) | (3,0) |
| (0,1) | (1,1) | | (3,1) |
| (0,2) | (1,2) | (2,2) | (3,2) |

Figure 17: Een puntje op (2,1)

```
  point(2, 1);
}
```

Het commando `size` werkt als volgt:

```
size(4, 3);
eerste getal: hoeveel pixels breed
tweede getal: hoeveel pixels hoog
```

Het commando `point` werkt als volgt:

```
point(2, 1);
eerste getal: hoeveel pixels naar rechts
tweede getal: hoeveel pixels omlaag
```

## Opdrachten

1. Type de code over en run het programma. Zie je de pixel zitten?
2. Wat is de coodinaat boven (2,1)?
3. Wat is de coodinaat rechts van (2,1)?
4. Wat is de coodinaat onder (2,1)?
5. Wat is de coodinaat links van (2,1)?

## Oplossingen

1. Je kunt een heel klein zwart puntje zien
2. (2,0)
3. (3,1)
4. (2,2)
5. (1,1)

## Twee puntje tekenen

We gaan een puntje tekenen op (0,3) en (1,2). We maken het scherm 5 pixels breed en 4 pixels hoog.

Dat ziet er als tekening zo uit:

Hier zie je de code:

```
void setup()
{
  size(5, 4);
}

void draw()
```

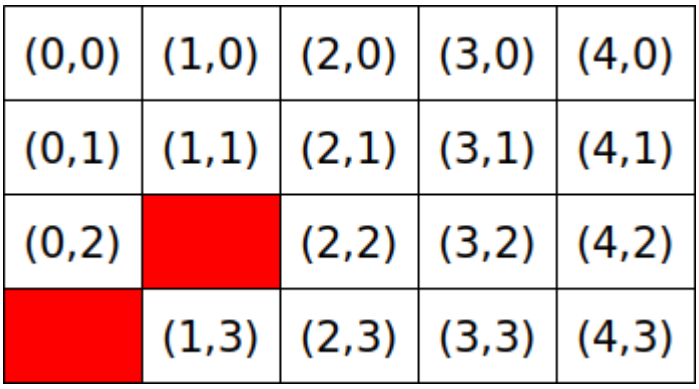


Figure 18: Een puntje op (0,3) en (1,2)

```
{
  point(0, 3);
  point(1, 2);
}
```

## Opdrachten

1. Type de code over en run het programma. Zie je de pixels zitten?
2. Maak de lijn af door twee nieuwe pixels te tekenen

## Oplossingen

1. Je kunt twee hele kleine zwart puntjes zien
2. Hier de code:

```
void setup()
{
  size(5, 4);
}

void draw()
{
  point(0, 3);
  point(1, 2);
  point(2, 1);
  point(3, 0);
}
```

### Een korte lijn tekenen

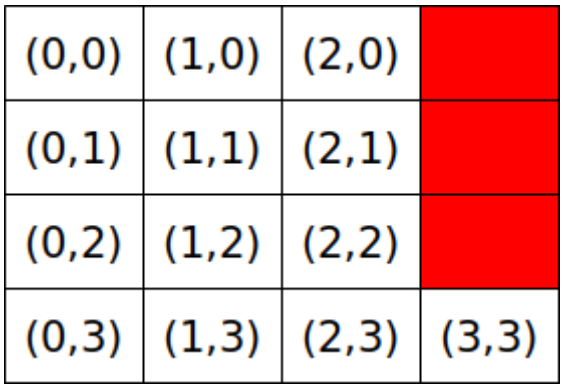


Figure 19: Een korte lijn

Op de tekening `Een korte lijn` zie je drie pixels die rood zijn gekleurd.

Programmeer deze lijn na. Maak ook het scherm net zo groot als op de tekenening.

### Oplossing

```
void setup()
{
  size(4, 4);
}

void draw()
{
  point(3, 0);
  point(3, 1);
  point(3, 2);
}
```

### Een ‘i’ tekenen

Plaatje `Een i` laat de pixels zien van een letter ‘i’:

Teken deze ‘i’ na. Maak ook het scherm groot genoeg

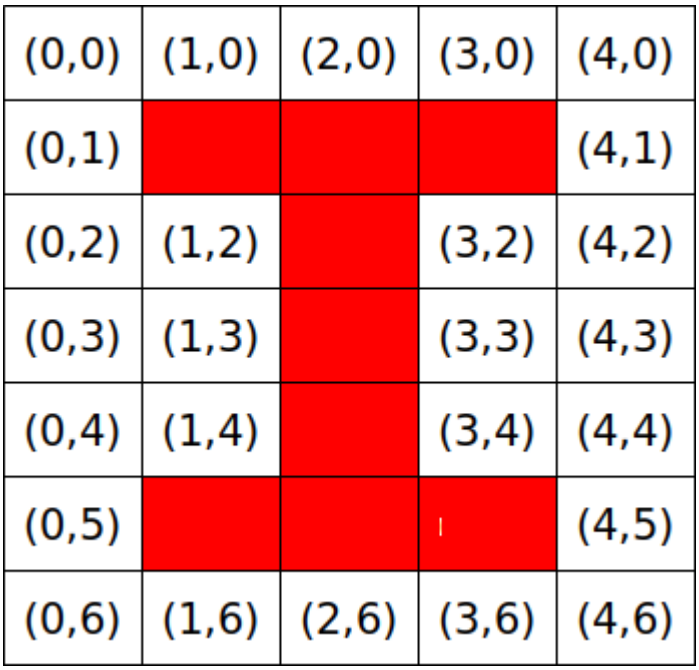


Figure 20: Een i

### Een hartje tekenen

Plaatje `I hartje` laat de pixels zien van de letter 'i' en een hartje:

Teken deze 'i' en het hartje. Maak ook het scherm groot genoeg

### Eindopdracht

Schrijf in pixels de tekst 'I hartje P':

## `line`

Zonder lijnen kun je bijna geen games maken. Een van de allereerste successgames, Asteroids, bestond vooral uit lijnen:

Je kunt een lijn tekenen met een boel puntjes, maar de `line` functie werkt gemakkelijker.

In deze les gaan we leren

- hoe je lijnen tekent

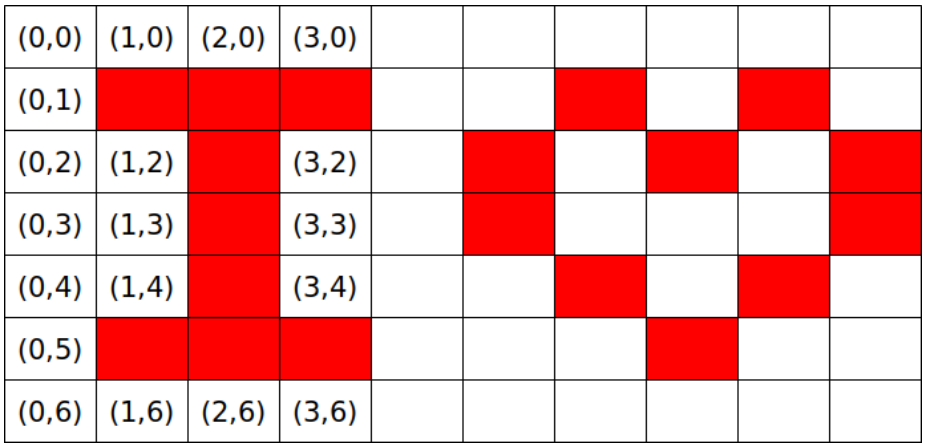


Figure 21: I hartje

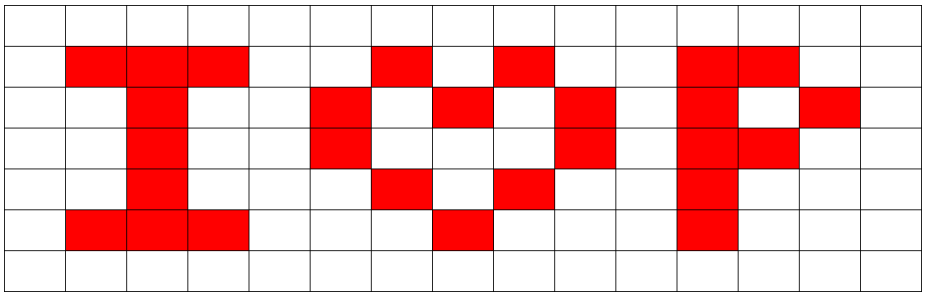

Figure 22: Point eindopdracht

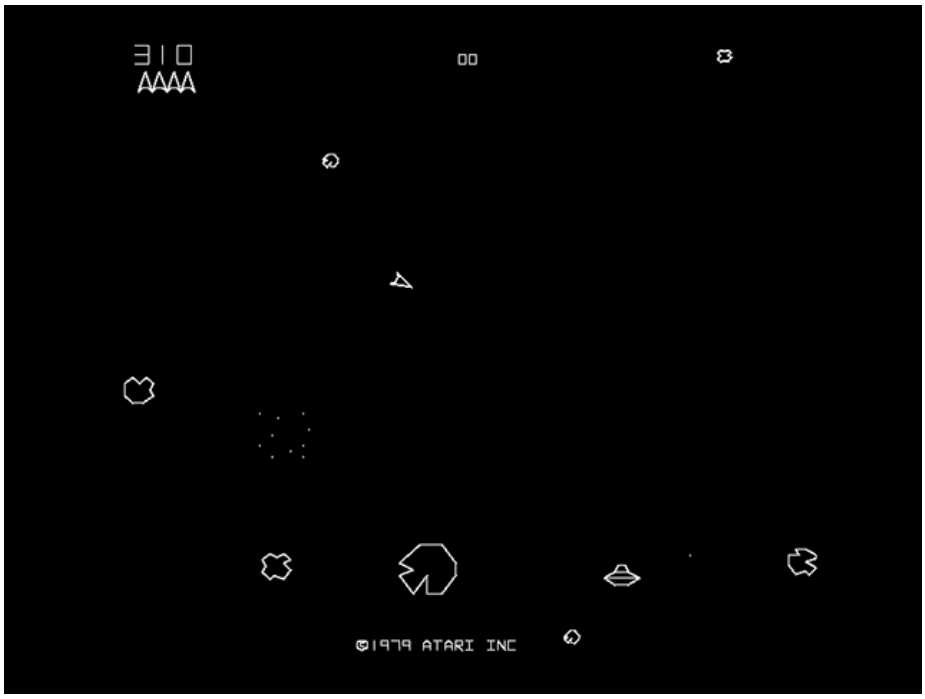


Figure 23: Asteroids

Kun je nog geen puntjes tekenen? Ga dan naar de les waarin je puntjes leert tekenen

## Lijnen

Een lijn bestaat uit pixels. Om een lijn te tekenen, moet je een beginpixel en eindpixel kiezen. Processing tekent dan zelf de pixels ertussenin.

## Een lijn tekenen

Op het plaatje staat een lijn die van (1,2) naar (3,0) gaat:

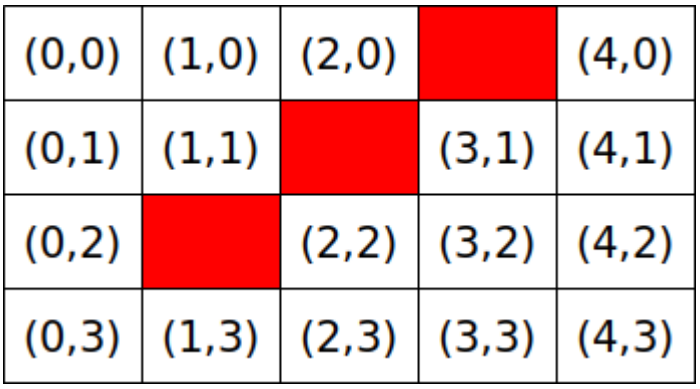


Figure 24: Lijn van (1,2) naar (3,0)

In Processing programmer je dat zo:

```
void setup()
{
  size(5,4);
}

void draw()
{
  line(1,2,3,0);
}
```

## Opdracht

1. Type de code over. Kun je de lijn zien?
2. Laat de lijn nu gaan van (1,2) naar (3,2). Wat zie je?
3. Draai de eerste lijn nu om: van (3,0) naar (1,2). Wat zie je?

## Oplossing

1. De lijn kun je wel zien. Hij is dikker dan drie puntjes
2. De lijn moet worden getekent met `line(1,2,3,2)`. De lijn is nu wel mooi dun
3. De lijn moet worden getekent met `line(3,0,1,2)`. De lijn is weer dik

## Een kruis tekenen

Op het plaatje staat een kruis:

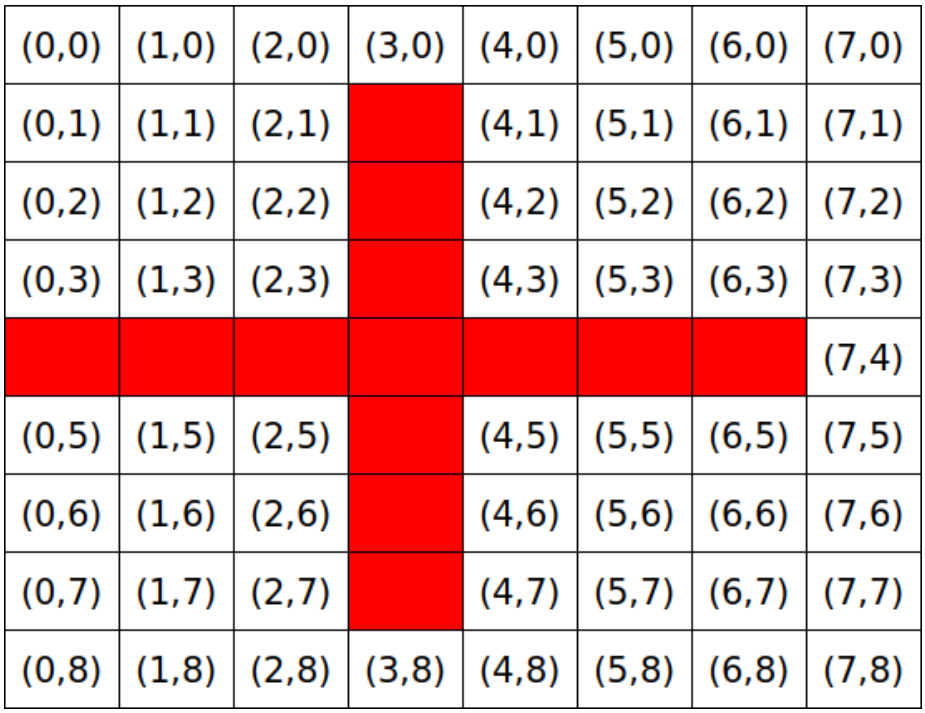


Figure 25: Een kruis

In Processing programmer je dat zo *ongeveer* zo:

```
void setup()
{
  size(30,20);
}

void draw()
{
```

```
  line(0,4,20,4);
  line(4,1,3,8);
}
```

## Opdracht

1. Type de code over. Welke `line` tekent het liggende streepje? Welke `line` tekent het neergaande streepje?
2. Maak het kruis mooi. Maak het scherm net zo groot als op het plaatje

## Oplossing

1. De eerste lijn is het liggende streepje.
2. Hier is de code:

```
void setup()
{
  size(8,9);
}

void draw()
{
  line(0,4,6,4);
  line(3,1,3,7);
}
```

## Een driehoek tekenen

Lijnen kunnen ook samen vorm worden.

Plaatje `Een driehoek` laat een driehoek zien:

De driehoek heeft drie hoeken met drie coordinaten.

## Opdrachten

1. Wat zijn de drie coordinaten van de punten van de driehoek?
2. Teken de driehoek na. Maak het scherm de juiste grootte

## Oplossing

1. (1,1), (9,1) en (1,9)

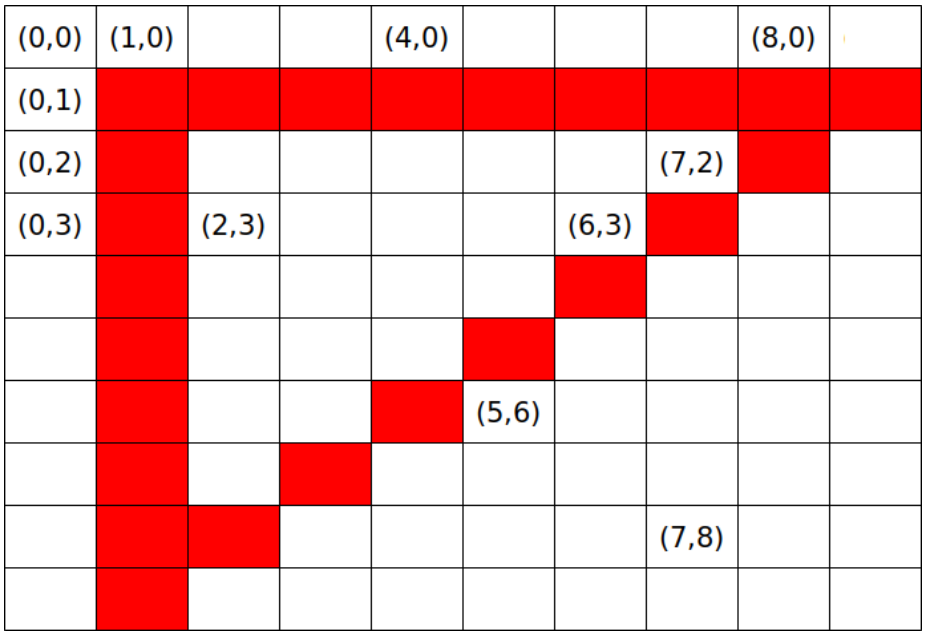


Figure 26: Een driehoek

2. Hier de code:

```
void setup()
{
  size(8,9);
}

void draw()
{
  line(1,1,9,1);
  line(1,1,1,9);
  line(1,9,9,1);
}
```

## Eindopdracht

Hierboven staat een tekening van een huisje. Maak deze tekening na in Processing. Je hebt acht lijnen nodig! Hij wordt niet zo mooi als op het plaatje, omdat de schuine lijnen niet mooi zijn.

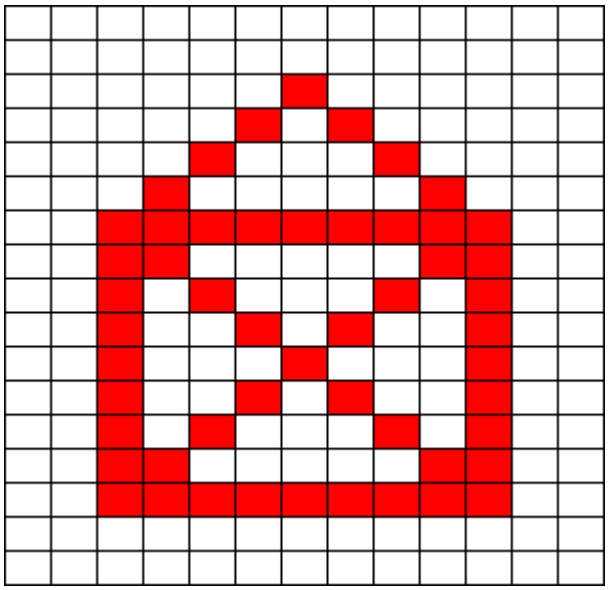

Figure 27: Line eindopdracht

## backGround

Zonder kleur zien games er minder mooi uit.

Een van de allereerste games met kleur heette Moria:

Om goed kleuren te kunnen gebruiken, moeten we leren hoe kleuren werken.

In deze les gaan we leren

- hoe kleuren werken

Zo gaat het eruit zien:

Je hoeft maar weinig te weten, behalve hoe je een mooi programma maakt

### Kleuren

Als je goed naar een beeldscherm kijkt, zie je dat elke pixel uit drie lampjes bestaat:

De lampjes hebben de kleuren rood, groen en blauw.

Omdat de lampjes zo klein zijn, ziet ons oog van afstand de menging van deze drie lampjes. Zo zien de lampjes rood en groen er samen uit als geel. Hier zie je hoe de kleuren mengen:


Figure 28: Moria

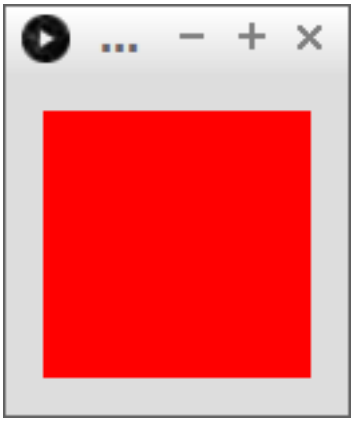

Figure 29: background

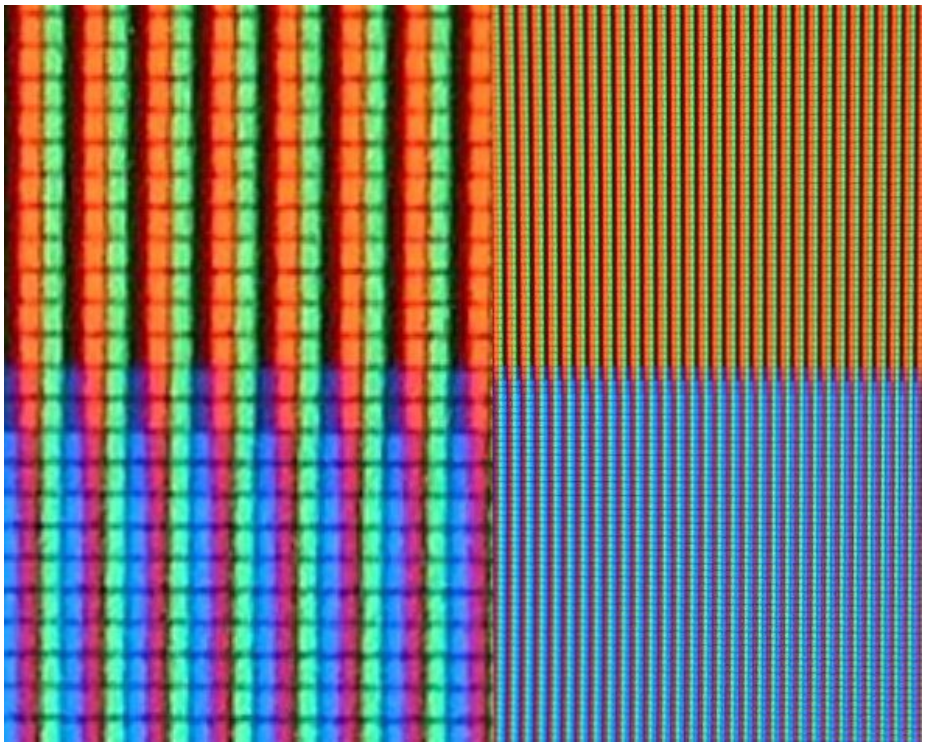

Figure 30: RGB pixels

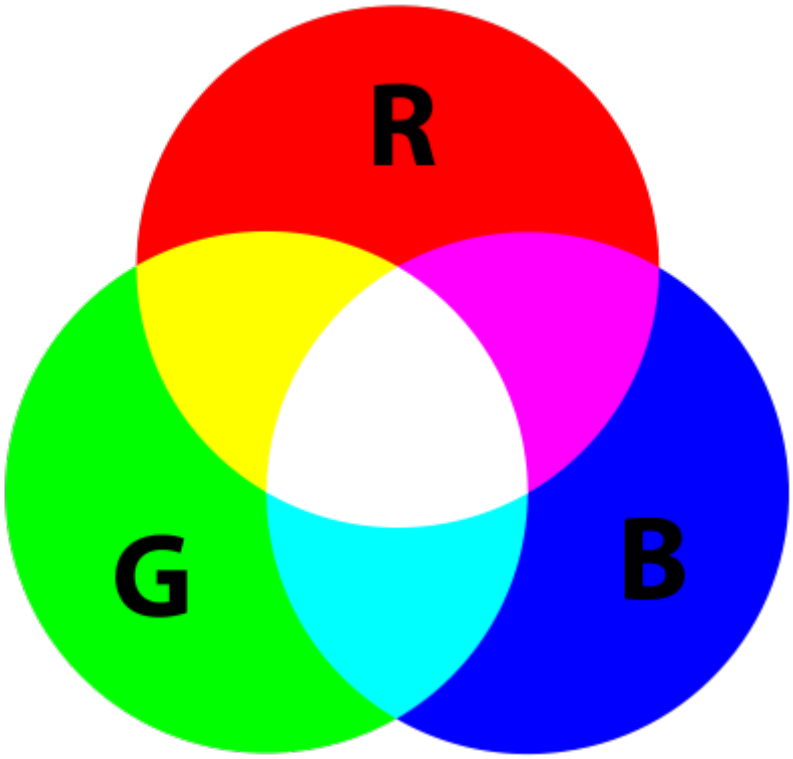


Figure 31: Additieve kleuren

Om wit te krijgen, heb je alledrie de kleuren nodig.

## Vragen

1. Welke drie kleuren lampjes heeft een pixel?
2. Met welke lampjes samen maak je geel?
3. Met welke lampjes samen maak je cyaan/lichtblauw?
4. Met welke lampjes samen maak je magenta/paars?
5. Met welke lampjes samen maak je wit?
6. Met welke lampjes samen maak je zwart?
7. Met welke lampjes samen maak je grijs?
8. Met welke lampjes samen maak je oranje?

## Antwoorden

1. Rood, groen en blauw
2. Rood en groen
3. Groen en blauw
4. Rood en blauw
5. Rood en groen en blauw
6. Met geen lampjes: als alle lampjes uit zijn, is het zwart
7. Met rood en groen en blauw, maar dan moeten ze niet op hun hardst branden
8. Met rood op z'n hardst en groen op halve kracht

### `background`

In Processing is er een functie om de achtergrond een kleur te geven. Deze functie heet `background`. `background` is Engels voor 'achtergrond'. `background` is een functie die drie getallen nodig heeft. Deze drie getallen bepalen hoe hard de rode, groene en blauwe lampjes gaan schijnen. Deze drie getallen noemen we de RGB waarde. Met het getal nul zeg je dat een lampje uit staat. Met het getal 255 zeg je dat een lampje aan staat. Met getallen tussen nul en 255 kun je het lampje ertussenin laten branden.

Met deze Processing code krijg je een rode achtergrond:

```
void setup()
{
  size(100,100);
}

void draw()
{
```

```
  background(255,0,0);
}
```

## Opdracht

1. Kopieer deze code in Processing en start de code
2. Wat is een RGB waarde?
3. Wat is de RGB waarde van groen? Maak in Processing een groene achtergrond
4. Wat is de RGB waarde van blauw? Maak in Processing een blauwe achtergrond
5. Wat is de RGB waarde van geel? Maak in Processing een gele achtergrond
6. Wat is de RGB waarde van cyaan/lichtblauw? Maak in Processing een cyane/lichtblauwe achtergrond
7. Wat is de RGB waarde van magenta/paars? Maak in Processing een magenta/paarse achtergrond
8. Wat is de RGB waarde van wit? Maak in Processing een witte achtergrond
9. Wat is de RGB waarde van zwart? Maak in Processing een zwart achtergrond
10. Wat is de RGB waarde van grijs? Maak in Processing een grijze achtergrond
11. Wat is de RGB waarde van donkerrood? Maak in Processing een donkerrode achtergrond
12. Wat is de RGB waarde van oranje? Maak in Processing een oranje achtergrond

## Oplossingen

1. OK
2. De Rood-Groen-Blauw waarde. Dit zijn drie getallen van nul tot en met 255 die bepalen hoe hard de drie kleurenlampjes branden
3. background(0,255,0)
4. background(0,0,255)
5. background(255,255,0)
6. background(0,255,255)
7. background(255,255,0)
8. background(255,255,255)
9. background(0,0,0)
10. background(128,128,128), maar andere getallen tussen 0 en 255 zijn ook goed. Als ze maar alledrie gelijk zijn
11. background(128,0,0), maar het eerste getal mag ook een ander getal tussen de 0 en 255 zijn
12. background(255,128,0), maar het tweede getal mag ook een ander getal in de buurt van 128 zijn

### Eindopdracht

Maak de achtergrond Thijs zijn favoriete kleur.

## `stroke`

Zonder kleur zien games er minder mooi uit.

Een van de allereerste games met kleur heette Moria:


Figure 32: Moria

In deze les gaan we leren

- hoe je een lijn een kleur kan geven.

Zo gaat het eruit zien:

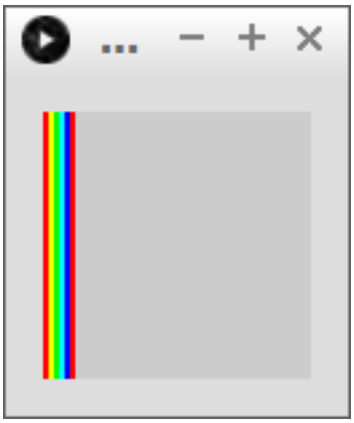

Figure 33: Stroke

Weet je nog niet hoe kleuren werken, ga dan naar de les background

## Een lijnkleur instellen

In Processing is er een functie om lijnen een kleur te geven. Deze functie heet `stroke`. `stroke` is Engels voor '(penseel)streek'. `stroke` is een functie die drie getallen nodig heeft. Deze drie getallen zijn de RGB waarden.

Met deze Processing code krijg je een rode lijn:

```
void setup()
{
  size(100,100);
}

void draw()
{
  stroke(255,0,0);
  line(10,20,30,40);
}
```

Met `stroke` zeg je: 'vanaf nu wil ik deze lijnkleur'. Hieronder zie je hoe je twee groene en een blauwe lijn tekent:

```
void setup()
{
  size(100,100);
}
```

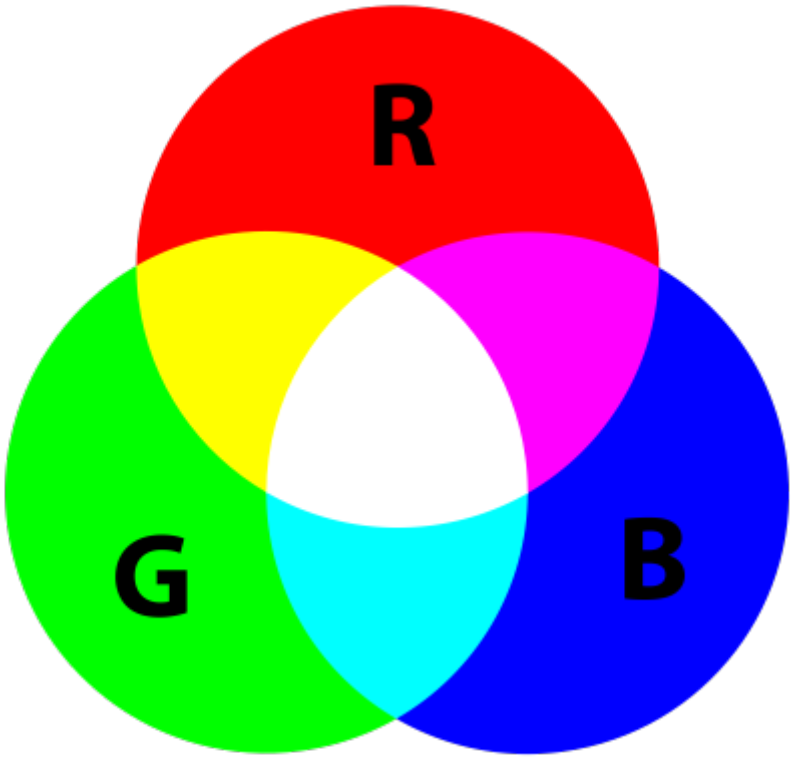


Figure 34: Kleurencirkel

```
void draw()
{
  stroke(0,255,0);
  line(10,20,30,40);
  line(50,60,70,80);
  stroke(0,0,255);
  line(90,10,20,30);
}
```

## Opdracht

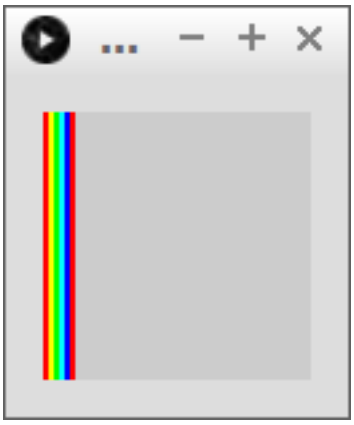

Figure 35: Stroke

- 1. Maak in Processing bovenstaande tekening na. Het venster is honderd bij honderd pixels. Elke kleur is met twee lijnen getekend. De kleuren zijn rood, geel, groen, cyaan, blauw, magenta

## Oplossing

```
void setup()
{
  size(100,100);
}

void draw()
{
  stroke(255,0,0);
  line(0,0,0,100);
```

```
  line(1,0,1,100);
  stroke(255,255,0);
  line(2,0,2,100);
  line(3,0,3,100);
  stroke(0,255,0);
  line(4,0,4,100);
  line(5,0,5,100);
  stroke(0,255,255);
  line(6,0,6,100);
  line(7,0,7,100);
  stroke(0,0,255);
  line(8,0,8,100);
  line(9,0,9,100);
  stroke(255,0,0);
  line(10,0,10,100);
  line(11,0,11,100);
}
```

### Eindopdracht

Maak nog vier kleuren bij het voorbeeld hierboven.

## `rect`

Vierkanten worden veel gebruikt in games.

Hier zie je een van de beroemdste games ooit:

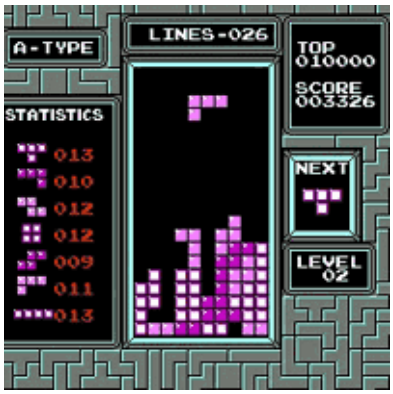


Figure 36: Tetris

Je kunt een vierkant tekenen met vier lijnen, maar de `rect` functie werkt gemakkelijker.

In deze les gaan we leren

- hoe je lijnen tekent

Zo gaat het eruit zien:

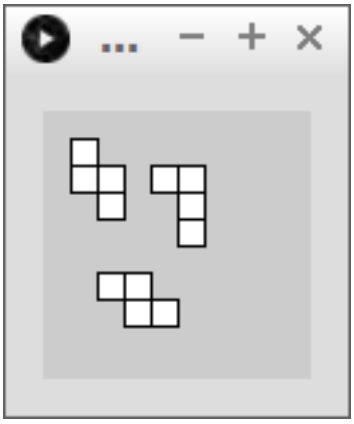

Figure 37: Rect

Kun je nog geen lijnen tekenen? Ga dan naar de les waarin je lijnen leert tekenen

## Rechthoeken

Een rechthoek bestaat uit vier lijnen. Om een rechthoek te tekenen, moet je een coordinaat, breedte en hoogte geven.

Om in Processing een rechthoek te tekenen, gebruik je de functie `rect`. De functie `rect` heeft vier getallen nodig. De eerste twee getallen zijn de coordinaat van de linkerbovenhoek van de rechthoek. Het derde getal is de breedte van de rechthoek. Het vierde getal is de hoogte van de rechthoek.

Hier zie je een rechthoek met coordinaat (1,2), een breedte van drie pixels en hoogte van vier pixels:

In Processing teken je deze rechthoek met:

```
rect(1,2,3,4);
```

Hier is nog een rechthoek:

De linkerbovenhoek heeft coordinaat (2,1), hij is vier pixels breed en drie pixels hoog.

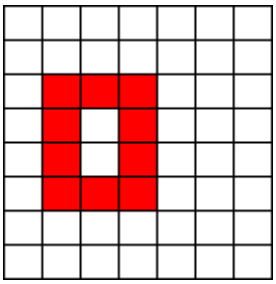

Figure 38: Rechthoek 1

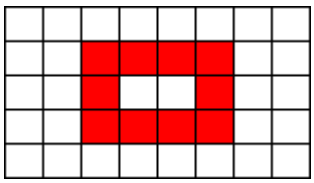

Figure 39: Rechthoek 2

## Vragen

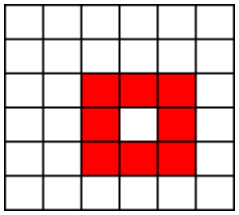

Figure 40: Rechthoek 3

1. Hierboven staat een rechthoek. Wat is de coordinaat van de linkerbovenhoek? Hoe breed is de rechthoek? Hoe hoog is de rechthoek?
2. Hierboven staat een rechthoek. Wat is de coordinaat van de linkerbovenhoek? Hoe breed is de rechthoek? Hoe hoog is de rechthoek?
3. Hierboven staat een rechthoek. Wat is de coordinaat van de linkerbovenhoek? Hoe breed is de rechthoek? Hoe hoog is de rechthoek?
4. Een rechthoek heeft als coordinaat (0,0), is twee pixels breed en drie pixels hoog. Wat is het Processing commando?
5. Een rechthoek heeft als coordinaat (1,2), is drie pixels breed en vier pixels hoog. Wat is het Processing commando?
6. Een rechthoek heeft als coordinaat (10,20), is dertig pixels breed en veertig pixels hoog. Wat is het Processing commando?

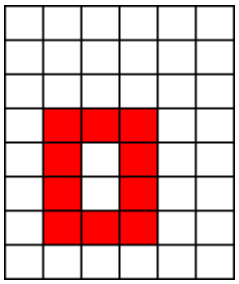

Figure 41: Rechthoek 4

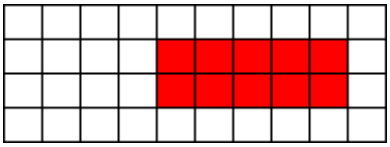

Figure 42: Rechthoek 5

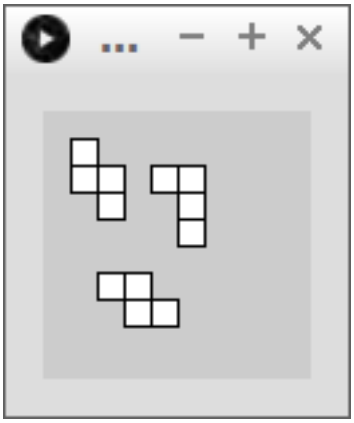

Figure 43: Rect

10.Hierboven staat een tekening. Maak deze tekening na in Processing

## Oplossing

10. Zie hieronder:

```
void setup()
{
  size(100,100);
}

void draw()
{
  rect(10,10,10,10);
  rect(10,20,10,10);
  rect(20,20,10,10);
  rect(20,30,10,10);

  rect(40,20,10,10);
  rect(50,20,10,10);
  rect(50,30,10,10);
  rect(50,40,10,10);

  rect(20,60,10,10);
  rect(30,60,10,10);
  rect(30,70,10,10);
  rect(40,70,10,10);
}
```

## Eindopdracht

Maak een cirkel van vierkanten. Dit *mag* zoals op het plaatje, maar je mag ook zelf iets verzinnen.

# `ellipse`

Cirkels en ovalen worden veel gebruikt in games.

Hier zie je een beroemde game, Bubble Bobble, dat veel met cirkels werkt:

Je kunt een ovaal tekenen met heel veel puntjes, maar de `ellipse` functie werkt gemakkelijker.

In deze les gaan we leren hoe je ovalen tekent.

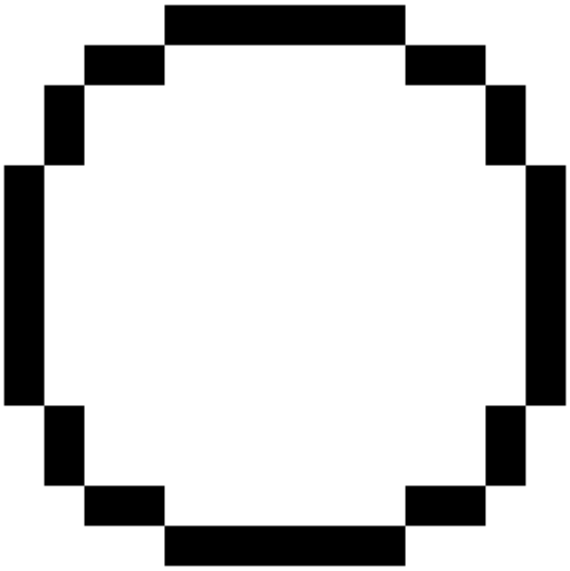

Figure 44: Een cirkel van vierkanten


Figure 45: Bubble Bobble

Zo gaat het eruit zien:

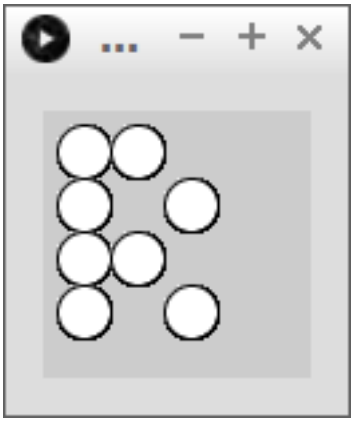

Figure 46: Ellipse

Kun je nog geen rechthoeken tekenen? Ga dan naar de les waarin je rechthokeen leert tekenen

## Ovalen

Een ovaal heeft een middelpunt, breedte en hoogte. Om een ovaal te tekenen, moet je een coordinaat, breedte en hoogte geven.

Om in Processing een ovaal te tekenen, gebruik je de functie `ellipse`. De functie `ellipse` heeft vier getallen nodig. De eerste twee getallen zijn de coordinaat van het midden van de ovaal. Het derde getal is de breedte van de ovaal. Het vierde getal is de hoogte van de ovaal.

Hier zie je een ovaal met middelpunt (3,2), een breedte van vijf pixels en hoogte van drie pixels:

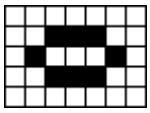

Figure 47: Ovaal 1

In Processing teken je deze ovaal met:

```
ellipse(3,2,5,3);
```

Hier is nog een ovaal:

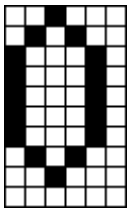

Figure 48: Ovaal 2

Het middelpunt heeft coordinaat (2,4), hij is vijf pixels breed en negen pixels hoog.

## Vragen

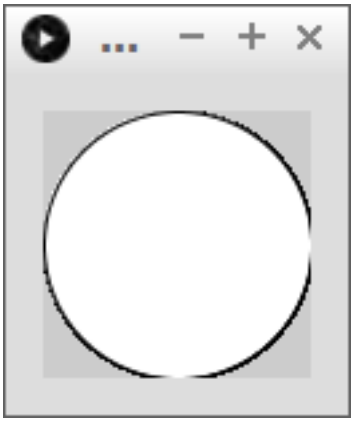

Figure 49: Ovaal 3

1. Je wilt bovenstaande plaatje namaken. Het venster is 100 pixels breed en 100 pixels hoog. Wat is het middelpunt van de cirkel? Hoe breed is de cirkel? En hoe hoog? Hoe maak je dit in Processing?
2. Je wilt bovenstaande plaatje namaken. Het venster is 200 pixels breed en 100 pixels hoog. Wat is het middelpunt van de cirkel? Hoe breed is de cirkel? En hoe hoog? Hoe maak je dit in Processing?
3. Je wilt bovenstaande plaatje namaken. Het venster is 200 pixels breed en 100 pixels hoog. Wat zijn de middelpunten van de cirkels? Hoe breed zijn de cirkels? En hoe hoog? Hoe maak je dit in Processing?
4. Hierboven staat een tekening. Maak deze tekening na in Processing


Figure 50: Ovaal 4

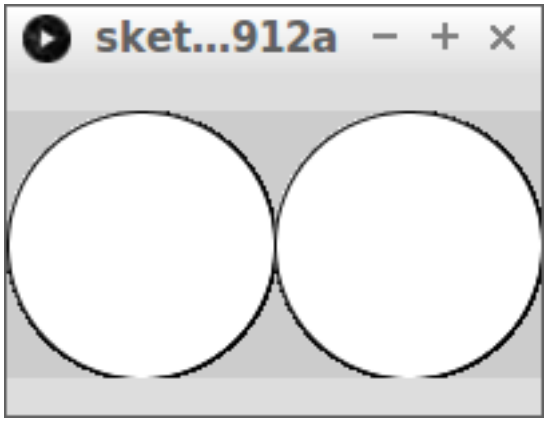


Figure 51: Ovaal 5

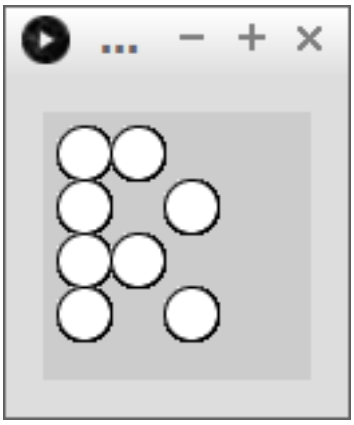

Figure 52: Ellipse

## Oplossing

1. Zie hieronder:

```
void setup()
{
  size(100, 100);
}

void draw()
{
  ellipse(50, 50, 100, 100);
}
```

2. Zie hieronder:

```
void setup()
{
  size(200, 100);
}

void draw()
{
  ellipse(100, 50, 200, 100);
}
```

3. Zie hieronder:

```
void setup()
{
  size(200, 100);
```

```
}

void draw() {
  ellipse( 50, 50, 100, 100);
  ellipse(150, 50, 100, 100);
}
```

4. Zie hieronder:

```
void setup()
{
  size(100, 100);
}

void draw()
{
  ellipse(15,15, 20, 20);
  ellipse(15,35, 20, 20);
  ellipse(15,55, 20, 20);
  ellipse(15,75, 20, 20);

  ellipse(35, 15, 20, 20);
  ellipse(55, 35, 20, 20);
  ellipse(35, 55, 20, 20);
  ellipse(55, 75, 20, 20);
}
```

### Eindopdracht

Maak een cirkel die door twaalf kleinere cirkels volledig omringd en afgesloten is.

## fill

Een rechthoek of een ovaal kan ook ingekleurd worden.

Een van de beroemdste games ooit heeft bijvoorbeeld veel ingekleurde vierkanten:

In deze les gaan we leren hoe we de invulkleur van vierkanten en ovalen instellen

Weet je nog niet hoe je een lijn een kleur kan geven? Ga dan naar stroke

### Een rood vierkant

Zo teken je een rood ingekleurd vierkant:

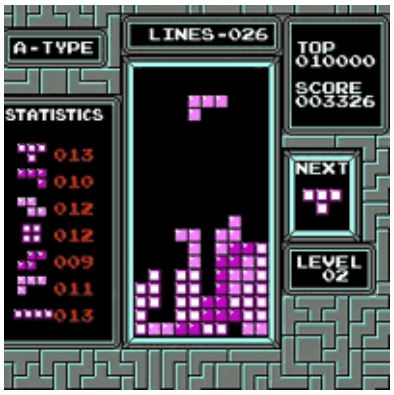


Figure 53: Tetris

```
void setup()
{
  size(200,100);
}

void draw()
{
  fill(255,0,0);
  rect(50,25,100,50);
}
```

## Een rood vierkant en een groene ovaal

Zo teken je een rood ingekleurd vierkant en een groen ingekleurde ovaal:

```
void setup()
{
  size(200,100);
}

void draw()
{
  fill(255,0,0);
  rect(50,25,100,50);
  fill(0,255,0);
  ellipse(50,75,40,40);
}
```

## Opdrachten

1. Wat gebeurt er als je de tweede `fill` weglaat?
2. Wat gebeurt er als je de eerste `fill` weglaat?
3. Maak een rood vierkant erbij, zonder een extra `fill` te gebruiken
4. Maak een groene ovaal erbij, zonder een extra `fill` te gebruiken
5. Maak een blauw vierkant erbij

## Oplossing

1. Dan wordt de ovaal ook rood
2. Dan wordt het vierkant ook groen
3. Hier een tweede rood vierkant:

```
void setup()
{
  size(200,100);
}

void draw()
{
  fill(255,0,0);
  rect(50,25,100,50);
  rect(75,50,100,50);
  fill(0,255,0);
  ellipse(50,75,40,40);
}
```

4. Hier een tweede groene ovaal:

```
void setup()
{
  size(200,100);
}

void draw()
{
  fill(255,0,0);
  rect(50,25,100,50);
  fill(0,255,0);
  ellipse(50,75,40,40);
  ellipse(75,50,40,40);
}
```

5. Hier een extra blauw vierkant:

```
void setup()
{
  size(200,100);
}

void draw()
{
  fill(255,0,0);
  rect(50,25,100,50);
  fill(0,255,0);
  ellipse(50,75,40,40);
  fill(0,0,255);
  rect(0,0,50,100);
}
```

## Eindopdracht

Hier zie je een schilderij van Mondriaan:

Figure 54: Composition II in Red, Blue, and Yellow door Mondriaan

Maak het schilderij ongeveer na, zoals bijvoorbeeld hier:

Je mag ook een eigen kunstwerk maken. Gebruik dan minstens vijf verschillende kleuren.

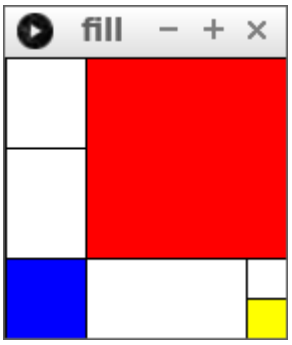


Figure 55: Composition II in Red, Blue, and Yellow door Richel

## `text`

Tekst wordt veel gebruikt, ook in games, voor bijvoorbeeld een score.

Hier zie je 'Zork, the underground empire', een van de beroemdste tekstavonturen ooit:


Figure 56: Zork

In deze les gaan we leren

- hoe je tekst op het scherm zet
- hoe je berekeningen op het scherm zet
- hoe je tekst vergroot
- hoe je tekst een kleur geeft

Kun je nog geen puntjes tekenen? Ga dan naar de les waarin je puntjes leert tekenen

Kun je nog geen vlakken inkleuren? Ga dan naar de les 'fill'

### Tekst

Hier zie je de tekst 'Hallo' laat zetten op coordinaat (10,20):

```
text("Hallo", 10, 20);
```

Let op dat de tekst tussen dubbele apostroffen (") moet.

`text` kan ook rekenen!

Hier plus en min:

```
text(128 + 64, 10, 20);
text(128 - 64, 10, 20);
```

Hier een keersom:

```
text(16 * 16, 10, 20);
```

Hier een deelsom:

```
text(256 / 16, 10, 20);
```

Tekstgrootte kun je aanpassen met

```
textSize(32);
```

Tekstkleur kun je aanpassen met `fill`:

```
fill(255, 0, 0);
```

## Opdracht

Zet de tekst `I love you 4 ever` op het scherm, waarbij:

- de coordinaat is (10, 30)
- de tekst heeft de normale kleur, dit is wit
- de tekst heeft de normale grootte, dit is 8 pixels

## Oplossing

```
void setup()
{
  size(300,300);
}

void draw()
{
  text("I love you 4 ever", 10, 30);
}
```

## Opdracht

Zet de tekst `I love you 4 ever` op het scherm, waarbij:

- de coordinaat is (10, 30)
- de tekst zwart is (tip: gebruik `fill`)
- de tekst heeft de normale grootte, dit is 8 pixels

## Oplossing

```
void setup()
{
  size(300,300);
}

void draw()
{
  fill(0, 0, 0);
  text("I love you 4 ever", 10, 30);
}
```

## Opdracht

Zet de tekst `I love you 4 ever` op het scherm, waarbij:

- de coordinaat is (10, 30)
- de tekst zwart is
- de tekst is 32 pixels groot (tip: gebruik `textSize`)

## Oplossing

```
void setup()
{
  size(300,300);
  textSize(32);
}

void draw()
{
  fill(0, 0, 0);
  text("I love you 4 ever", 10, 30);
}
```

## Opdracht

Zet de tekst `I love you 4 ever` op het scherm, waarbij:

- de tekst is 32 pixels groot (tip: gebruik `textSize`)
- alle woorden zwart zijn, behalve `love`, die rood is
- de `4` is de uitkomst van een berekening, bijvoorbeeld `2 + 2` (maar hoe moeilijker de berekening, hoe stoerder)

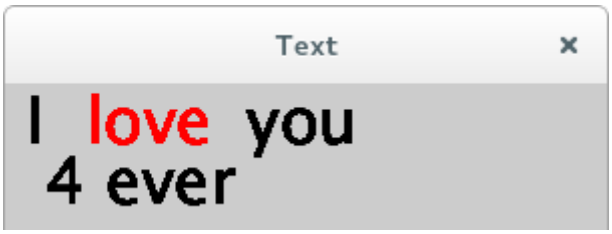


Figure 57: Text

## Oplossing

```
void setup()
{
  size(300,300);
  textSize(32);
}

void draw()
{
  fill(0, 0, 0);
  text("I", 10, 30);
  fill(255, 0, 0);
  text("love", 40, 30);
  fill(0, 0, 0);
  text("you", 120, 30);
  text(256 / 64, 20, 60);
  text("ever", 50, 60);
}
```

## Eindopdracht

Zet je eigen naam mooi op het scherm. Een voorbeeld:


Figure 58: Eindopdacht text