

Figure 1: Book 3

## Contents

Voorwoord	1
Bal die horizontaal stuitert	2
ellipse en background	12
Bal die schuin stuitert	19
text	25

## Voorwoord



Figure 1: Het logo van De Jonge Onderzoekers

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

### Over dit boek

Dit boek heeft een CC-BY-NC-SA licentie.



Figure 2: De licentie van dit boek

(C) Dojo Groningen 2016-2018

Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is **niet altijd even mooi**.

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.

### text: oplossing 7

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float botsingen = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  fill(0, 255, 0);
  text(y, 10, 40);
  fill(0, 0, 255);
  text(botsingen, 10, 60);
  fill(random(255), random(255), random(255));
  text("PONG", width / 2, height / 2);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
    botsingen = botsingen + 1;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### text: eindopdracht

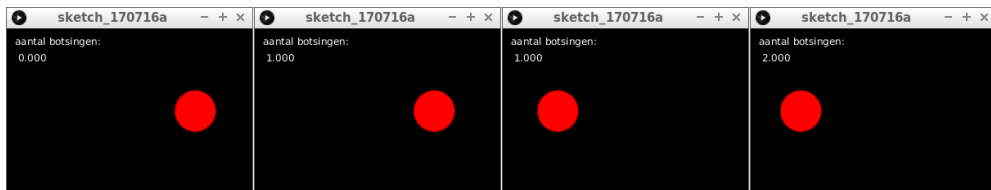


Figure 33: text: eindopdracht

Zet linksboven de tekst: `aantal botsingen:` in het wit. Zet daaronder de waarde van `botsingen` in het wit. Haal de rest van de tekst weg. Het aantal botsingen moet ook hoger worden als je links botst.

### Bal die horizontaal stuitert

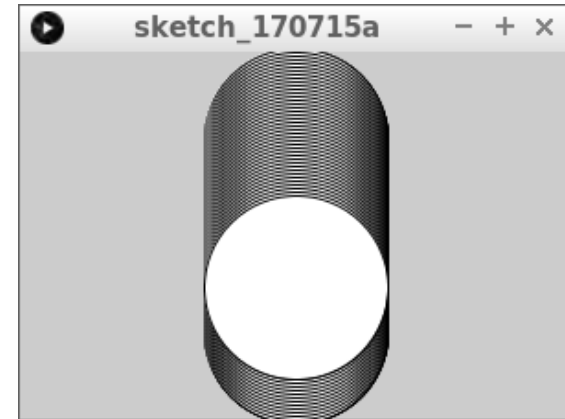


Figure 3: Eindopdracht Bal Die Horizontaal Stuitert

In deze les gaan we een bal horizontaal laten stuiteren.

### Bal die horizontaal stuitert: opdracht 1

```
float x = 300;
float snelheid = 2;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
  if (x > 650)
  {
    x = -50;
  }
}
```

Type deze code over. Zet de waarde van `snelheid` op 1. Wat zie je?

### Bal die horizontaal stuitert: oplossing 1

Je ziet dat de bal rustig naar rechts beweegt. Als de bal rechts het beeld uit gaat, komt deze vanaf links weer in het beeldscherm.

```
float x = 300;
float snelheid = 1;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
  if (x > 650)
  {
    x = -50;
  }
}
```



Figure 4: Bal die horizontaal stuitert: oplossing 1

### Bal die horizontaal stuitert: opdracht 2

Zet de waarde van `snelheid` op 0. Wat zie je?

### text: oplossing 6

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float botsingen = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  fill(0, 255, 0);
  text(y, 10, 40);
  fill(0, 0, 255);
  text(botsingen, 10, 60);
  fill(random(255), random(255), random(255));
  text("PONG", width / 2, height / 2);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### text: opdracht 7

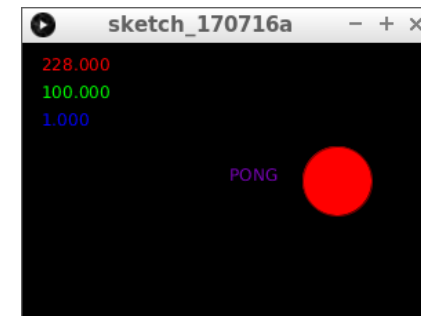


Figure 32: text: opdracht 7

Maak `botsingen` hoger als de bal teken de rechter muur komt.

### text: oplossing 5

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  fill(0, 255, 0);
  text(y, 10, 40);
  fill(random(255), random(255), random(255));
  text("PONG", width / 2, height / 2);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### text: opdracht 6

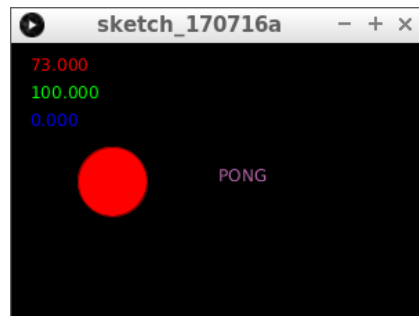


Figure 31: text: opdracht 6

Maak een variabele erbij: `aantal_botsingen`. `aantal_botsingen` heeft als beginwaarde nul. Zet deze waarde op het scherm, in het wit.ex

### Bal die horizontaal stuitert: oplossing 2

De bal staat nu stil

```
float x = 300;
float snelheid = 0;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
  if (x > 650)
  {
    x = -50;
  }
}
```

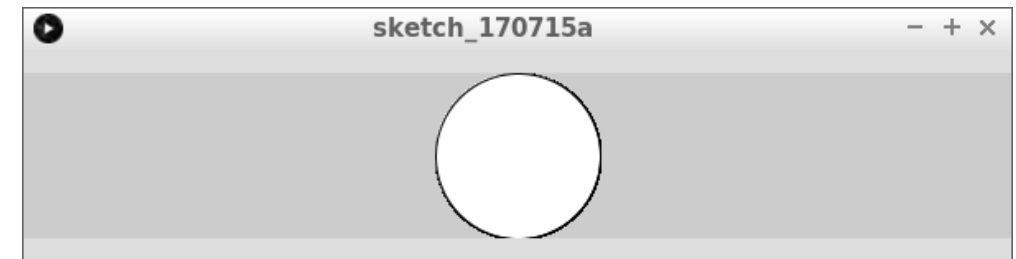


Figure 5: Bal die horizontaal stuitert: oplossing 2

### Bal die horizontaal stuitert: opdracht 3

Zet de waarde van `snelheid` op -1. Wat zie je?



Het getal -1 spreek je uit als 'min een'. Het is het getal een lager dan nul.

### Bal die horizontaal stuitert: oplossing 3

De bal gaat nu naar links en komt niet meer in beeld.

```
float x = 300;
float snelheid = -1;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
  if (x > 650)
  {
    x = -50;
  }
}
```



Figure 6: Bal die horizontaal stuitert: oplossing 3

### Bal die horizontaal stuitert: opdracht 4



Figure 7: Bal die horizontaal stuitert: opdracht 4

Zorg dat als de bal links uit beeld gaat, deze weer rechts verschijnt. Doe dit door een tweede `if` te maken

### text: oplossing 4

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  fill(0, 255, 0);
  text(y, 10, 40);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### text: opdracht 5

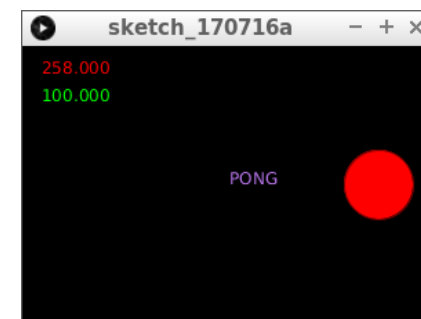


Figure 30: text: opdracht 5

Zet de tekst `PONG` bovenin en midden op het scherm. De tekst moet een willekeurige kleur krijgen. Gebruik `width` om het midden te bepalen.

### text: oplossing 3

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  text(y, 10, 40);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### text: opdracht 4

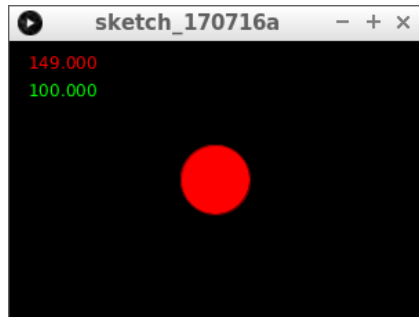


Figure 29: text: opdracht 4

Met fill kun je de kleur van je tekst bepalen. Na de eerste text, zet de fill op groen.

### Bal die horizontaal stuitert: oplossing 4

```
float x = 300;
float snelheid = -1;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
  if (x > 650)
  {
    x = -50;
  }
  if (x < -50)
  {
    x = 650;
  }
}
```


### Bal die horizontaal stuitert: opdracht 5


Denk na: wat moet er met de snelheid gebeuren om de bal te laten stuiten (van links naar rechts)?


Als je durft: probeer dit zonder verder te lezen!

Bal die horizontaal stuitert: oplossing 5



Als tegen de linkerkant komt, moet de bal de bal naar rechts gaan. Als tegen de rechterkant komt, moet de bal de bal naar links gaan.



	Als de bal ...	dan moet deze ...
	aan de linkerkant komt	naar rechts gaan
	aan de rechkant komt	naar links gaan



```
if (...) { }
```

	x < 50	snelheid = 1;
	x > 550	snelheid = -1;

Bal die horizontaal stuitert: opdracht 6



Figure 8: Opdracht 6

Laat de bal nu van links naar rechts stuiten

text: opdracht 3

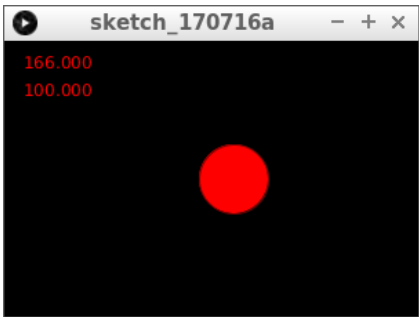


Figure 28: text: opdracht 3

Het programma laat nu de x zien. Laat eronder nu ook de y coördinaat op het scherm zien.



text: oplossing 2

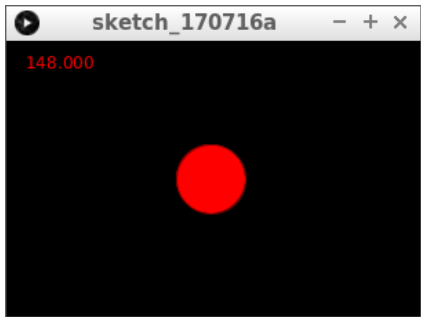


Figure 27: text: oplossing 2

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Bal die horizontaal stuitert: oplossing 6

```
float x = 300;
float snelheid = 1;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x,50,100,100);
  x = x + snelheid;
  if (x > 550)
  {
    snelheid = -1;
  }
  if (x < 50)
  {
    snelheid = 1;
  }
}
```

Bal die horizontaal stuitert: opdracht 7

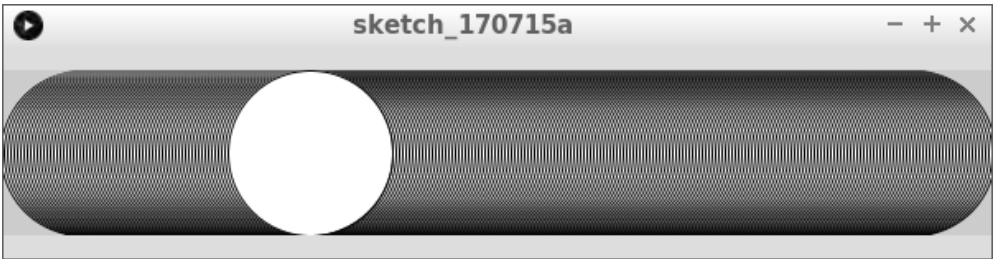




Figure 9: Opdracht 7

Laat de bal nu met een snelheid van 2 pixels per keer bewegen.

	
text(100, 200, 300);	'Lieve computer, zet de tekst 100 op het scherm met als linkerbovenhoek (200, 300)
text("Hallo", 100, 200);	'Lieve computer, zet de tekst Hallo op het scherm met als linkerbovenhoek (100, 200)

Bal die horizontaal stuitert: oplossing 7

```
float x = 300;
float snelheid = 2;



void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
  if (x > 550)
  {
    snelheid = -2;
  }
  if (x < 50)
  {
    snelheid = 2;
  }
}
```


Bal die horizontaal stuitert: opdracht 8

Verander in de code:

van	naar
snelheid = 2;	snelheid = -snelheid;
snelheid = -2;	snelheid = -snelheid;



`snelheid = -snelheid;`      'Lieve computer, verander het (min of plus)teken van snelheid'



Het getal -1 heeft een minteken. Het getal 1 (ook goed: +1) heeft een plusteken.

text: oplossing 1

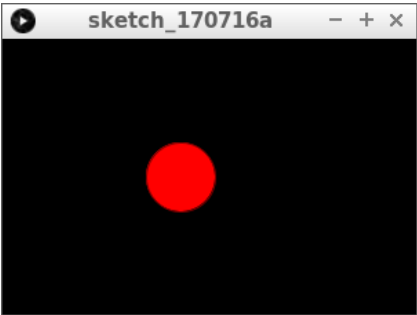


Figure 26: text: oplossing 1

Opdracht 2

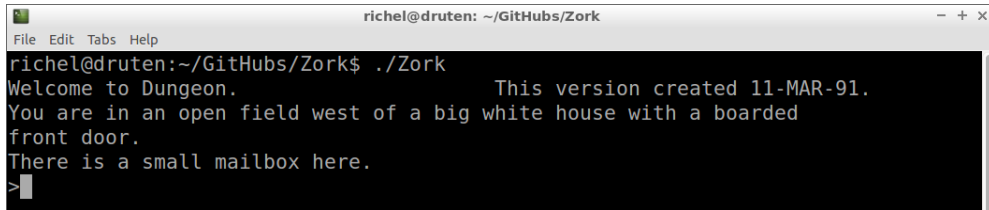
Voeg na ellipse de volgende regel toe:

```
text(x, 10, 20);
```

## 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:

A screenshot of a terminal window titled 'richel@druten: ~/GitHubs/Zork'. The terminal shows the command './Zork' being executed, which outputs a welcome message and game instructions. The text displayed is: 'Welcome to Dungeon. This version created 11-MAR-91. You are in an open field west of a big white house with a boarded front door. There is a small mailbox here.' followed by a prompt '>'.

```
richel@druten: ~/GitHubs/Zork
File Edit Tabs Help
richel@druten:~/GitHubs/Zork$ ./Zork
Welcome to Dungeon.          This version created 11-MAR-91.
You are in an open field west of a big white house with a boarded
front door.
There is a small mailbox here.
>
```

Figure 25: Zork

## text: opdracht 1

Run deze code:

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

## Bal die horizontaal stuitert: oplossing 8



Figure 10: Opdracht 8

```
float x = 300;
float snelheid = 2;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
  if (x > 550)
  {
    snelheid = -snelheid;
  }
  if (x < 50)
  {
    snelheid = -snelheid;
  }
}
```

Bal die horizontaal stuitert: eindopdracht

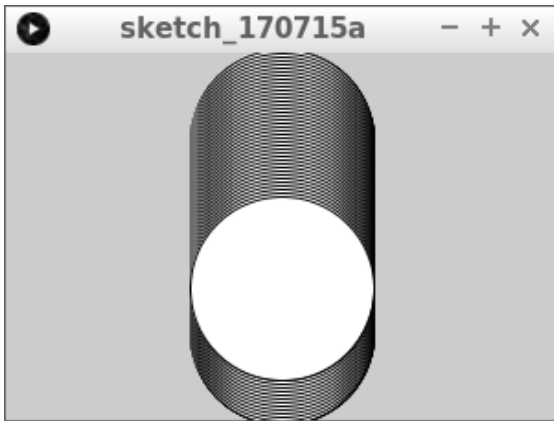


Figure 11: Bal die horizontaal stuitert: eindopdracht

Laat de bal nu omhoog en omlaag stuiten.

Bal die schuin stuitert: eindopdracht

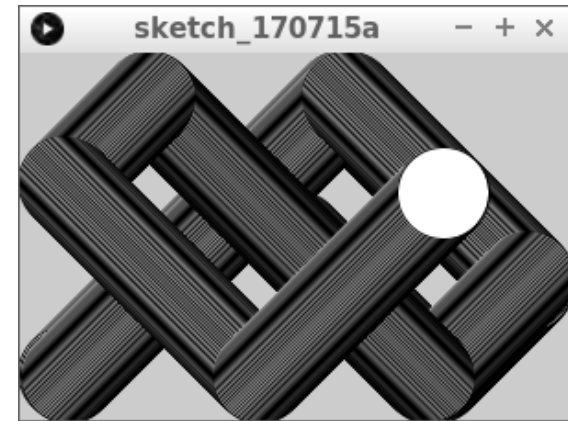


Figure 24: Bal die schuin stuitert: eindopdracht

Laat nu de bal ook stuiten als deze de bovenkant raakt.

### Bal die schuin stuitert: oplossing 4

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float snelheid_omlaag = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  y = y + snelheid_omlaag;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (y > 175)
  {
    snelheid_omlaag = -snelheid_omlaag;
  }
}
```

### ellipse en background

In deze les gaan we leren hoe je ovalen tekent.

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



Figure 12: Bubble Bobble

### ellipse en background: opdracht 1

Run deze code

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
}

void draw()
{
  stroke(x, x, x);
  fill(x, x, x);
  ellipse(x, height / 2, 50, 50);
  x = x + snelheid;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

ellipse en background: oplossing 1

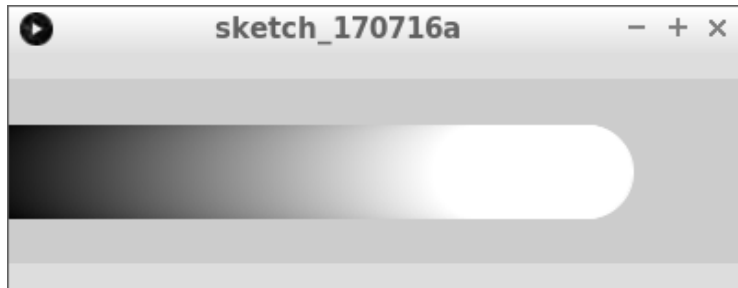


Figure 13: Oplossing 1

ellipse en background: opdracht 2



Figure 14: Opdracht 2

Verander het derde en vierde getal van `ellipse` (de 100s), zodat de ovaal twee keer zo hoog wordt.

Bal die schuin stuitert: oplossing 3

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float snelheid_omlaag = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  y = y + snelheid_omlaag;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Bal die schuin stuitert: opdracht 4

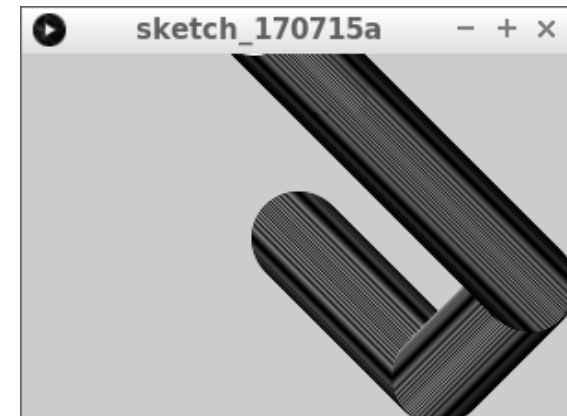


Figure 23: Bal die schuin stuitert: opdracht 4

Laat de bal nu omhoog stuiten als de bal de onderkant van het scherm raakt.

### Bal die schuin stuitert: oplossing 2

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float snelheid_omlaag = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### Bal die schuin stuitert: opdracht 3



Figure 22: Bal die schuin stuitert: opdracht 3

Laat de bal nu ook naar onder bewegen. Hij hoeft nog niet te stuiten.

### ellipse en background: oplossing 2

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
}

void draw()
{
  stroke(x, x, x);
  fill(x, x, x);
  ellipse(x, height / 2, 50, 100);
  x = x + snelheid;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

### ellipse en background: opdracht 3



Figure 15: Opdracht 3

Voeg als eerste regel van draw toe:

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

### ellipse en background: oplossing 3

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
}

void draw()
{
  background(255, 0, 0);
  stroke(x, x, x);
  fill(x, x, x);
  ellipse(x, height / 2, 50, 100);
  x = x + snelheid;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

### ellipse en background: opdracht 4

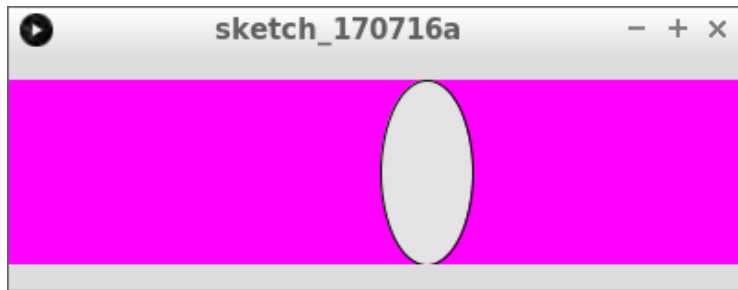


Figure 16: Oplossing 4

Maak de achtergrond nu magenta. Maak de rand van de cirkel zwart

### Bal die schuin stuitert: oplossing 1

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### Bal die schuin stuitert: opdracht 2

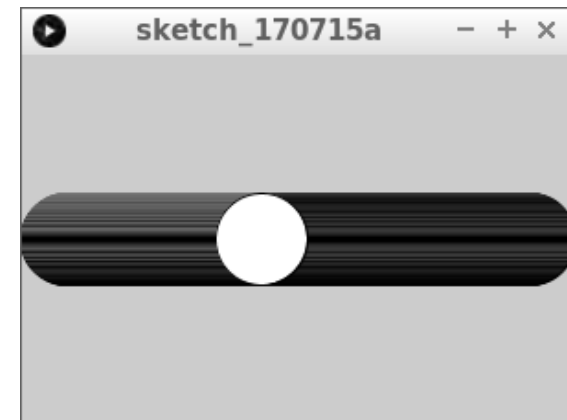


Figure 21: Bal die schuin stuitert: opdracht 2

Maak een nieuwe variable aan met de naam `snelheidomlaag`. Geef deze de beginwaarde van 1. De bal hoeft nog niet te bewegen.



## Bal die schuin stuitert

In deze les gaan we een bal schuin laten stuiten.

### Bal die schuin stuitert: intro

```
float x = 150;
float snelheid_naar_rechts = 1;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, 100, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

### Bal die schuin stuitert: opdracht 1

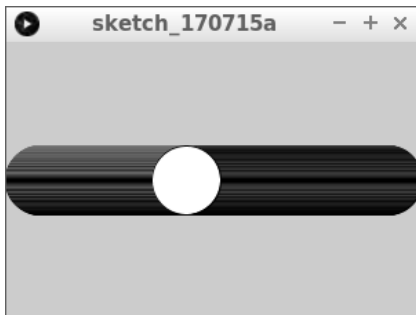


Figure 20: Bal die schuin stuitert: opdracht 1

Maak een nieuwe variable aan met de naam y. Teken de ovaal op coördinaat (x, y). Zorg dat de code hetzelfde blijft doen



### ellipse en background: oplossing 4

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
}

void draw()
{
  background(255, 0, 255);
  stroke(0, 0, 0);
  fill(x, x, x);
  ellipse(x, height / 2, 50, 100);
  x = x + snelheid;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

### ellipse en background: opdracht 5

Zet de regel met background nu na size (binnen setup).

#### ellipse en background: oplossing 5



Figure 17: Oplossing 5

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
  background(255, 0, 255);
}
```

```
void draw()
{
  stroke(0, 0, 0);
  fill(x, x, x);
  ellipse(x, height / 2, 50, 100);
  x = x + snelheid;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

#### ellipse en background: opdracht 6

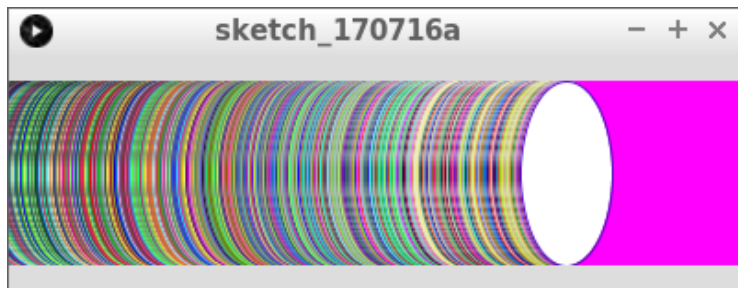


Figure 18: Opdracht 6

Maak de lijnkleur van de ovaal een willekeurige kleur: de rood, groen en blauw waarde moeten alledrie willekeurig zijn.

#### ellipse en background: oplossing 6

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
  background(255, 0, 255);
}

void draw()
{
  stroke(random(256), random(256), random(256));
  fill(x, x, x);
  ellipse(x, height / 2, 50, 100);
  x = x + snelheid;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

#### ellipse en background: eindopdracht

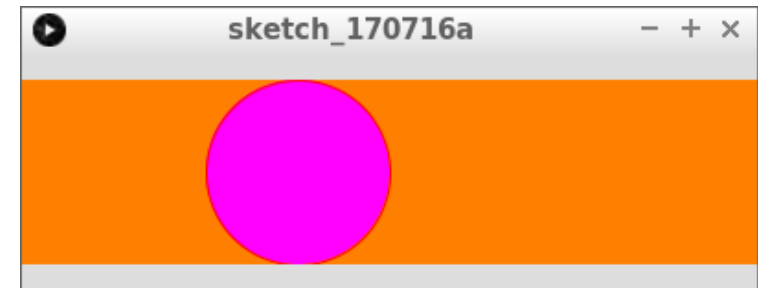


Figure 19: Eindopdracht ellipse en background

Maak de ovaal rond, even hoog als het scherm, magenta, en met een rode rand. De achtergrond moet oranje zijn (oranje is vol rood, halfvol groen en geen blauw). Er mag geen sliert komen van oude ovalen.