



Figure 1: Open Dag

Contents

Voorwoord	1
Een Mooi Programma	1
Bal naar rechts	6
Bal die eeuwig naar rechts gaat	12
Bal die horizontaal stuitert	17

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

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



Figure 3: De licentie van dit boek

(C) Dojo Groningen 2016

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 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.

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:

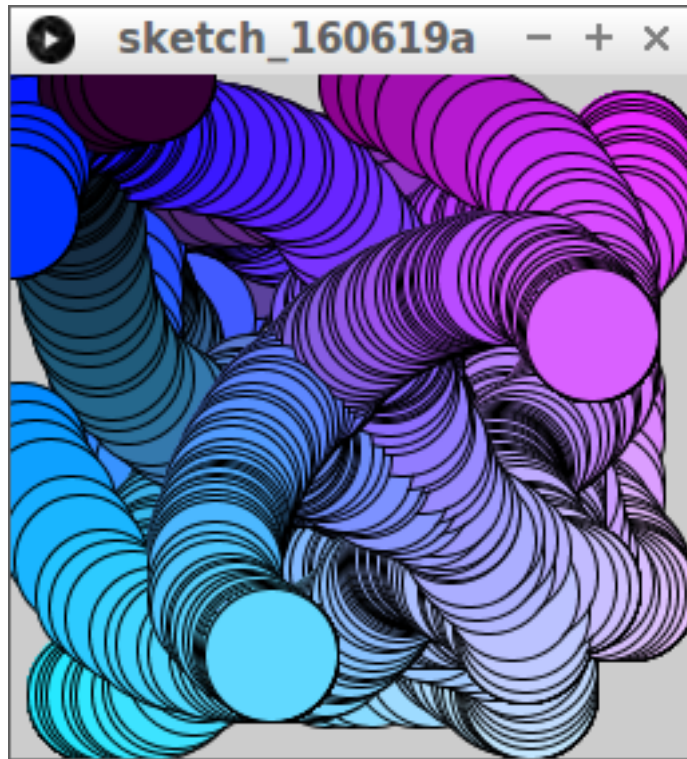


Figure 4: EenMooiProgramma

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 kopiëren

Processing begint met een leeg programma 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.

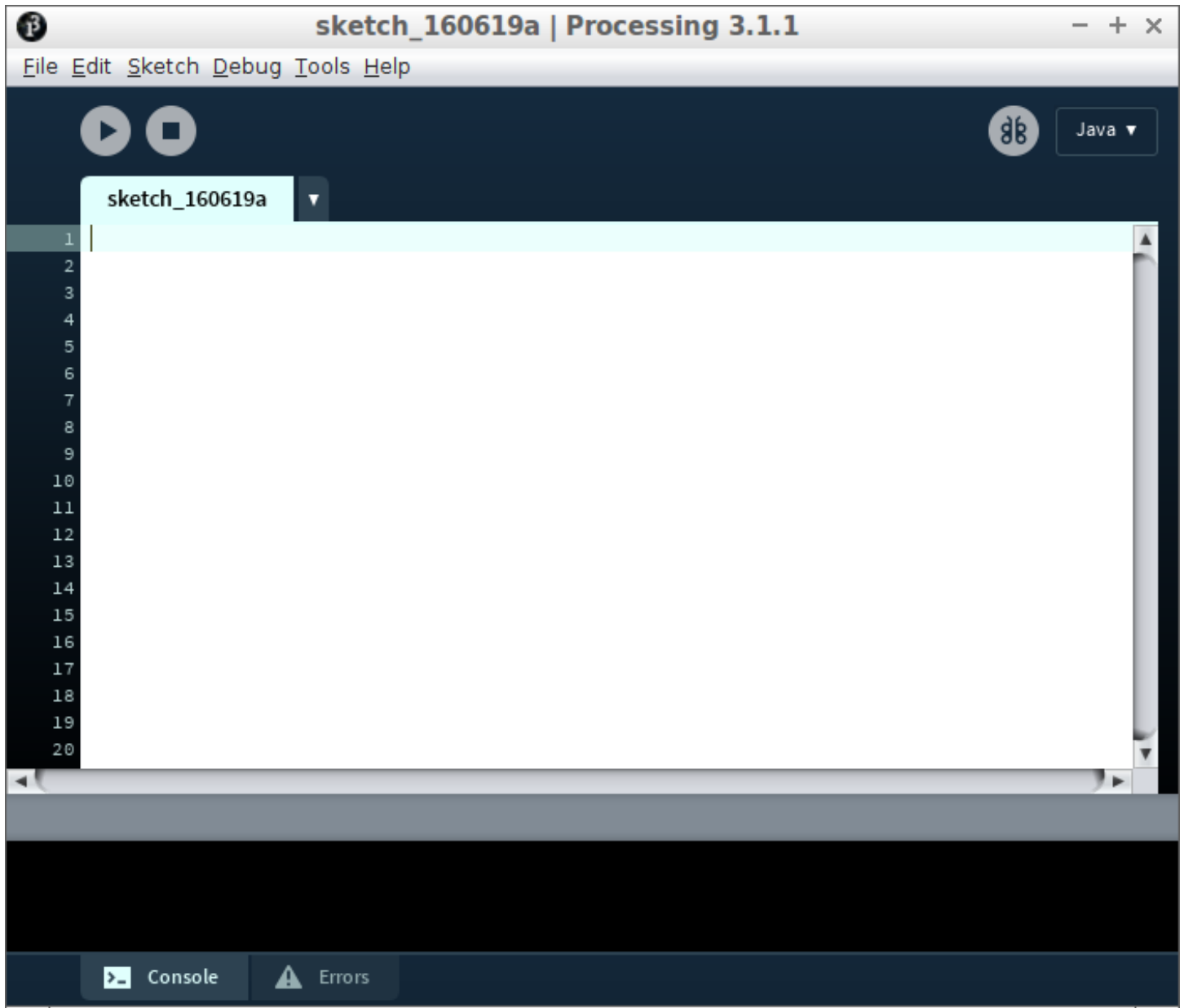


Figure 5: Processing zonder code

Om code te kopiëren gebruik je sneltoetsen:

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

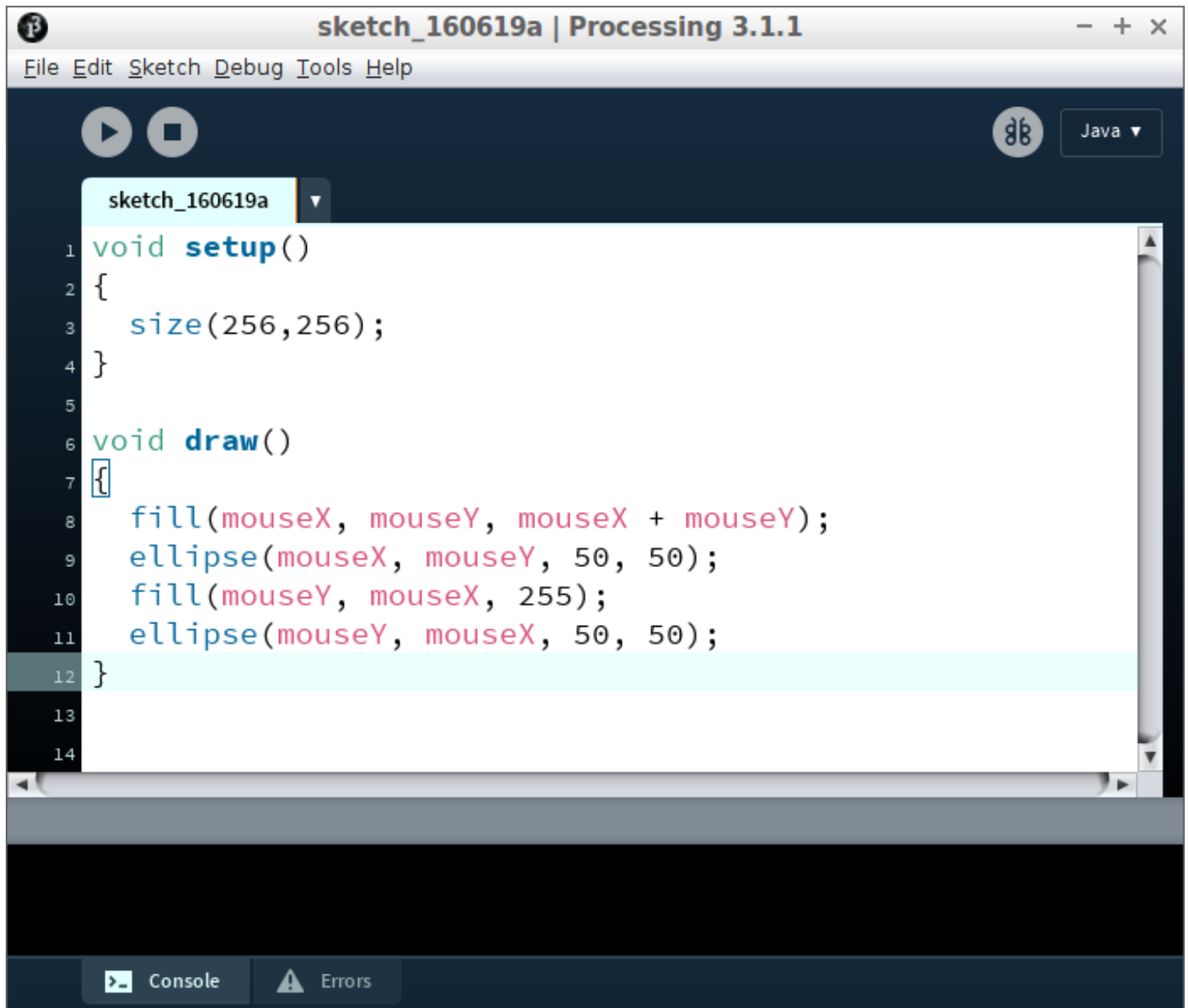


Figure 6: Processing met code

Programma uitvoeren

- Klik op de 'Run' knop

Als het goed is, zie je nu het programma!

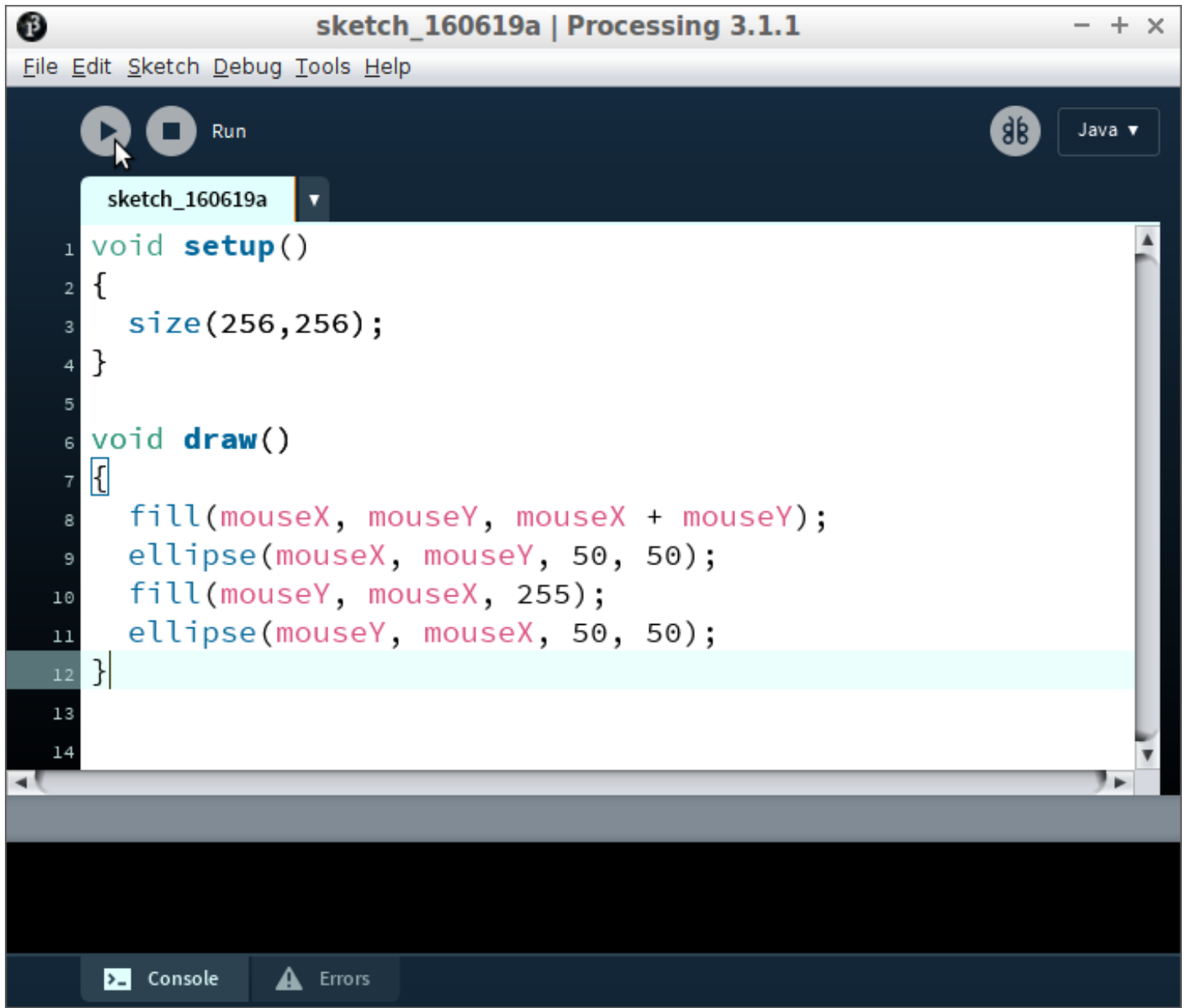


Figure 7: De Run knop

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

Bal naar rechts

In deze les gaan we een bal naar rechts laten bewegen.

Het ziet er zo uit:

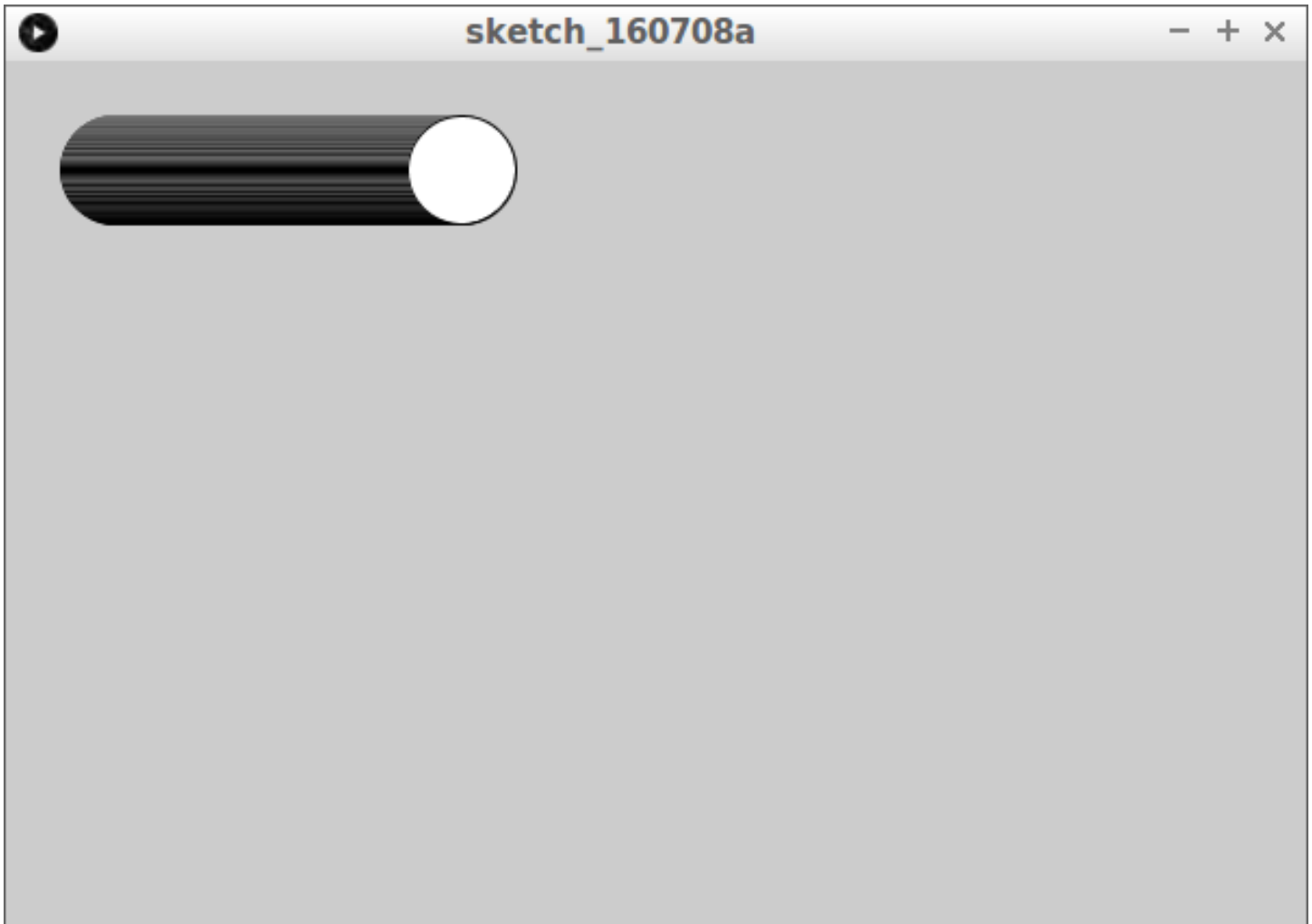


Figure 8: Bal naar rechts

We leren in deze les wat een variabele is. Je kunt (bijna) niet programmeren zonder variabelen.

Wat weten we al?

Als je de vorige lessen hebt gedaan, weet je wat deze code doet:

```
void setup()
{
  size(600, 400);
}

void draw()
```

```
{
  ellipse(50,50,50,50);
}
```

Vragen

- 1. Wat doet dit programma?
- 2. Waar wordt deze cirkel getekend?
- 3. Komt deze cirkel tegen de rand aan?

Oplossing

- 1. Het programma maakt een scherm van 600 pixels breed en 400 pixels hoog. Dan wordt er een cirkel getekent met een middelpunt op coördinaat (50,50) en vijftig pixels breed en hoog
- 2. In de linkerbovenhoek
- 3. Nee, de cirkel komt maar voor de helft van middelpunt tot de rand

Code met een variabele

We gaan de code aanpassen:

```
float x = 50;

void setup()
{
  size(600, 400);
}

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

Vragen

- 1. Wat doet dit programma?
- 2. Wat zijn de verschillen?

Oplossing

- 1. Het programma maakt een scherm van 600 pixels breed en 400 pixels hoog. Dan wordt er een cirkel getekent met een middelpunt op coördinaat (50,50) en vijftig pixels breed en hoog
- 2. Er zijn geen verschillen!

Variabelen

De eerste nieuwe regel is:

```
float x = 50;
```


In mensentaal is dit: ‘Lieve computer, onthoud het getal **x**. **x** heeft een beginwaarde van vijftig.’

Een variabele is iets dat onthouden moet worden. Een kassa onthoudt bijvoorbeeld de hoeveelheid geld die alle boodschappen bij elkaar zijn. Variabelen die jij weet, zijn: je naam, je leeftijd, je geboortedatum, je adres, je telefoonnummer, je emailadres, en nog veel meer. Als iemand je je leeftijd vraagt, dan weet je welk getal je moet zeggen.

Het woord **x** is de naam van een variable. In dit geval van hoe ver de cirkel naar rechts staat. Het woord **float** betekent dat ‘x’ een getal is. Het symbool **=** betekent ‘wordt vanaf nu’. Het getal 50 is de beginwaarde.

De tweede veranderde regel is:

```
ellipse(x,50,50,50);
```

In mensentaal is dit: ‘Lieve computer, teken een ovaal die:

- **x** naar rechts is. De computer weet nog wel wat **x** is: vijftig!
- 50 omlaag is
- 50 pixels breed is
- 50 pixels hoog is

Vragen

1. Wat als je `float x = 50;` weghaalt?
2. Wat als je `float x = 50;` verandert naar `float rechts = 50;`?
3. Wat als je `float x = 50;` verandert naar `float x = 100;`?
4. Wat als je `ellipse(x,50,50,50);` weghaalt?
5. Wat als je `ellipse(x,50,50,50);` verandert naar `ellipse(rechts,50,50,50);`?
6. Wat als je `ellipse(x,50,50,50);` verandert naar `ellipse(x,x,50,50);`?
7. Wat als je `ellipse(x,50,50,50);` verandert naar `ellipse(x,x,x,50);`?
8. Wat als je `ellipse(x,50,50,50);` verandert naar `ellipse(x,x,x,x);`?
9. Wat als je **x** vervangt door **dinosaurus**?

Oplossing

- 1. Dan komt er onderin ‘x cannot be resolved to variable’ te staan. Het programma doet het niet meer. Dit omdat de computer **x** niet meer kan vinden
- 2. Dan komt er onderin ‘x cannot be resolved to variable’ te staan. Het programma doet het niet meer. Dit omdat de computer **x** niet meer kan vinden
- 3. Dan komt het middelpunt van de cirkel honderd pixels naar rechts te liggen
- 4. Dan wordt er geen cirkel meer getekend
- 5. Dan komt onderin ‘The variable “rechts” does not exist’ te staan. De computer kent geen variabele met de naam **rechts**
- 6. Dan komt het middelpunt van de cirkel honderd pixels naar rechts en honderd pixels naar onder te liggen
- 7. Dan komt het middelpunt van de cirkel honderd pixels naar rechts en honderd pixels naar onder te liggen. De cirkel wordt honderd pixels breed
- 8. Dan komt het middelpunt van de cirkel honderd pixels naar rechts en honderd pixels naar onder te liggen. De cirkel wordt honderd pixels breed en hoog
- 9. Dan komt onderin ‘The variable “dinosaurus” does not exist’ te staan. De computer kent geen variabele met de naam **dinosaurus**

Bewegen

Nu gaan we de cirkel laten bewegen:

```
float x = 50;

void setup()
{
    size(600, 400);
}

void draw()
{
    ellipse(x,50,50,50);
    x = x + 1;
}
```

Vragen

- Wat doet dit programma?
- Wat zijn de verschillen?

De nieuwe regel is:

```
x = x + 1;
```

In mensentaal is dit: ‘Lieve computer, x is vanaf nu plus een’. Of: ‘Maak x een hoger’.

Vragen

- Als x vijftig is, wat is x dan na `x = x + 1`;
- Als x 51 is, wat is x dan na `x = x + 1`;
- Als x 52 is, wat is x dan na `x = x + 1`;
- Als x 53 is, wat is x dan na `x = x + 1`;
- Als x 54 is, wat is x dan na `x = x + 1`;

Nu kunnen we snappen wat het programma doet. Hier staat het programma weer:

```
float x = 50;

void setup()
{
    size(600, 400);
}

void draw()
{
    ellipse(x,50,50,50);
    x = x + 1;
}
```

De eerste keer dat de computer **draw** gaat doen, dan vult deze voor x een 50 in. Daarna wordt x een hoger. Dan is **draw** klaar.

De tweede keer dat de computer **draw** gaat doen, dan vult deze voor x een 51 in. Daarna wordt x een hoger. Dan is **draw** klaar.

De derde keer dat de computer **draw** gaat doen, dan vult deze voor x een 52 in. Daarna wordt x een hoger. Dan is **draw** klaar.

Vragen

- Wat als je `ellipse(x,50,50,50)`; vervangt door `ellipse(50,50,50,50)`;
- Wat als je `ellipse(x,50,50,50)`; vervangt door `ellipse(50,x,50,50)`;

- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(50,50,x,50);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(50,50,50,x);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(x,x,50,50);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(50,x,x,50);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(50,50,x,x);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(x,50,50,x);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(x,x,x,50);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(50,x,x,x);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(x,50,x,x);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(x,x,50,x);`?
- Wat als je `ellipse(x,50,50,50);` vervangt door `ellipse(x,x,x,x);`?
- Wat als je `x = x + 1;` vervangt door `x = x + 2;`?
- Wat als je `x = x + 1;` vervangt door `x = x + 10;`?
- Wat als je `x = x + 1;` vervangt door `x = x + 0;`?
- Wat als je `x = x + 1;` vervangt door `x = x - 1;`?
- Wat als je `x = x + 1;` vervangt door `x = x - 0;`?
- Wat als je `x = x + 1;` vervangt door `x = x * 2;`?
- Wat als je `x = x + 1;` vervangt door `x = x * 1;`?
- Wat als je `x = x + 1;` vervangt door `x = x * 0;`?
- Wat als je `x = x + 1;` vervangt door `x = x / 2;`?
- Wat als je `x = x + 1;` vervangt door `x = x / -2;`?
- Wat als je `x = x + 1;` vervangt door `x = x / 1;`?
- Wat als je `x = x + 1;` vervangt door `x = x / 0;`?

Bal naar links

Haha, deze les heet ‘Bal naar rechts’, toch gaan we ook een bal naar links laten bewegen!

Opdracht

- Laat een bal aan de rechterkant van het scherm beginnen
- De bal moet in een rechte lijn naar links gaan

Oplossing

```
float x = 500;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(x,50,50,50);
  x = x - 1;
}
```

Bal naar onder

Haha, deze les heet ‘Bal naar rechts’, toch gaan we ook een bal naar onder laten bewegen!

Opdracht

- Verander de naam van de variabele `x` in `y`

- Laat een bal aan de bovenkant van het scherm beginnen
- De bal moet in een rechte lijn naar onder gaan

Oplossing

```
float y = 50;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(50,y,50,50);
  y = y + 1;
}
```

Bal snel omhoog

Nu gaan we de bal sneller laten bewegen

Opdracht

- Laat een bal aan de onderkant van het scherm beginnen
- De bal moet in een rechte lijn naar boven gaan
- De bal moet twee keer zo snel gaan

Oplossing

```
float y = 50;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(50,y,50,50);
  y = y - 2;
}
```

Bal groter

Nu gaan we de bal sneller bewegen en groter maken

Opdracht

- Gebruik als variabele naam **x**
- Laat een bal aan de linker van het scherm beginnen
- De bal moet in een rechte lijn naar rechts gaan
- De bal moet even groot blijven

Oplossing

```
float x = 50;

void setup()
{
    size(600, 400);
}

void draw()
{
    fill(x,x,x);
    ellipse(x,50,50,50);
    x = x + 1;
}
```

Bal veranderd van kleur

Nu gaan we de bal een kleur geven met `fill`

Opdracht

- Gebruik als variabele naam `t` (van tijd) in plaats van `x` of `y`
- Laat een bal aan de linker van het scherm beginnen
- De bal moet in een rechte lijn naar rechts gaan
- De kleur moet van zwart naar wit veranderen

Oplossing

```
float t = 50;

void setup()
{
    size(600, 400);
}

void draw()
{
    fill(t,t,t);
    ellipse(50,t,50,t);
    t = t + 1;
}
```

Eindopdracht

- gebruik een variabele `t` (van tijd)
- de bal moet snel omlaag en naar rechts gaan
- de bal moet groter worden in de breedte en hoogte
- de kleur moet van zwart naar wit veranderen

Bal die eeuwig naar rechts gaat

In deze les gaan we een bal eeuwig naar rechts laten gaan.



Figure 9: Bal die eeuwig naar rechts gaat 1

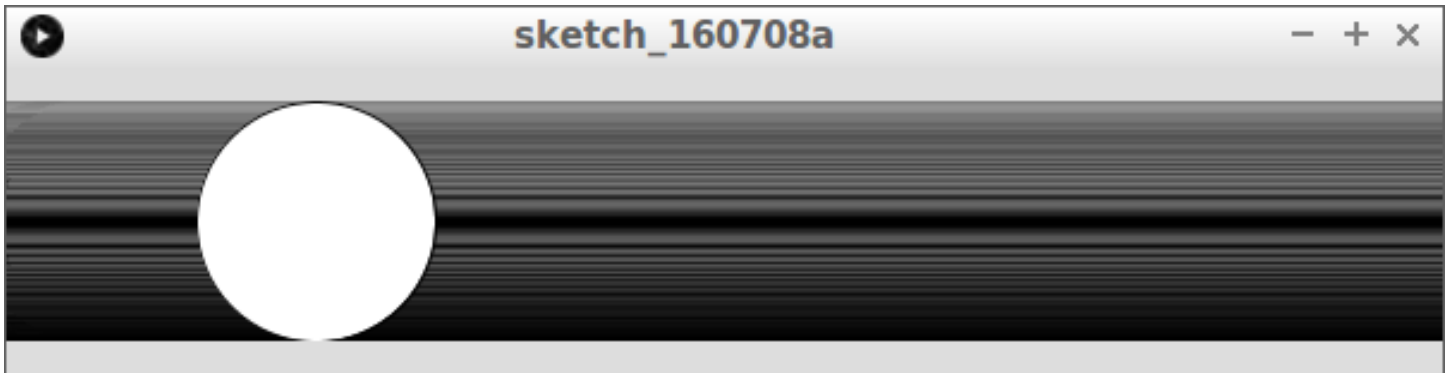


Figure 10: Bal die eeuwig naar rechts gaat 2

Het ziet er zo uit:

We leren in deze les wat `if`-statement is. Je kunt (bijna) niet programmeren zonder `if`-statements.

Wat weten we al?

Als je de vorige lessen hebt gedaan, weet je wat deze code doet:

```
float x = 50;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(x, 50, 50, 50);
  x = x + 1;
}
```

Vragen

- Wat doet dit programma?
- In welke richting beweegt de ovaal
- Blijft de ovaal zichtbaar op het scherm?
- Kopieer de code en bekijk het programma. Klopt wat je dacht?

Een if-statement

We willen kunnen zeggen: ‘Lieve computer, *als* de bal te ver naar rechts is, dan teleporteer je de bal naar rechts’. **if** is Engels voor ‘als’.

Zo zou dit kunnen:

```
if (x > 200)
{
    x = 100;
}
```

Dit betekent:

- **if**: begin van een if statement. Een if-statement heeft dan twee gedeeltes:
- **()**: tussen de ronde haken staat een test; iets wat waar of niet waar is
- **{}**: tussen de accolades staat wat de computer moet doen als de test waar is
- **x > 200**: dit staat tussen de ronde haken. Dit is de test ‘x is groter dan 200’. Het **>** tekenje betekent ‘groter dan’
- **x = 100**: dit staat tussen de accolades. Als ‘x is groter dan 200’ waar is, dan krijgt x de waarde 100

Preciezer zeg je: ‘Lieve computer, *als* x meer is dan 200, zet x dat op 100’. **if** is Engels voor ‘als’.

Vragen

- Kopieer het **if**-statement tussen de accolades van de **draw** functie
- Wat doet het programma?

Als het kopiëren niet is gelukt, gebruik dan deze code:

```
float x = 50;

void setup()
{
    size(600, 400);
}

void draw()
{
    ellipse(x,100,100,100);
    x = x + 1;
    if (x > 200)
    {
        x = 100;
    }
}
```

- Kun je ervoor zorgen dat de ovaal helemaal naar de linkerkant van het scherm springt?
- Kun je ervoor zorgen dat de ovaal helemaal naar rechts beweegt, voordat deze naar de linkerkant van het scherm springt?

Antwoord

Dit is een eeuwig naar rechts gaande bal:

```
float x = -50;

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

void draw()
{
```

```

ellipse(x,50,100,100);
x = x + 1;
if (x > 650)
{
    x = -50;
}
}

```

Het lijkt al een beetje op Lonelier Pong. Dit is geen toeval :-)

Bal die eeuwig naar links gaat

Gefopt! Ook al het de les ‘Bal die eeuwig naar rechts gaat’, we gaan toch de bal ook andere kanten op laten gaan.

We gaan nu een bal programmeren die eeuwig naar links gaat.

Wat je nu moet weten is het `if` statement om te zeggen wanneer de `x` te klein is:

```

if (x < 100)
{
    x = 500;
}

```

Hiermee zeg je: ‘Lieve computer, als `x` kleiner (`<`, hier kun je een `k` van maken) is dan honderd, zet dan `x` op vijfhonderd.

Opdracht

Maak een bal die eeuwig naar links gaat:

- De bal begint buiten het beeld
- De bal gaat helemaal het beeld uit
- Als de bal net uit het beeld uit, komt ‘ie meteen de andere kant weer binnen

Antwoord

Dit is een eeuwig naar links gaande bal:

```

float x = 650;

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

void draw()
{
    ellipse(x,50,100,100);
    x = x - 1;
    if (x < -50)
    {
        x = 650;
    }
}

```

Bal die eeuwig omlaag gaat

We hebben een bal naar rechts en naar links laten bewegen door de `x` coördinaat te veranderen. De bal kan ook naar omlaag en omhoog gaan door de `y` coördinaat te veranderen.

Opdracht

Schrijf een programma waarin een bal eeuwig omlaag gaat:

- maak het scherm 300 pixels breed en 200 pixels hoog
- gebruik een variable met naam `y`
- vervang de code `ellipse(x,50,100,00)` door `ellipse(50,y,100,100)`
- als de bal omlaag uit het scherm gaat, moet de bal weer bovenin komen

Oplossing

```
float y = -50;

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

void draw()
{
  ellipse(50,y,100,100);
  y = y + 1;
  if (y > 250)
  {
    y = -50;
  }
}
```

Bal die schuin gaat

Hoppa, nu we een variabele `x` of een `y` hebben gemaakt, gaan we beiden doen!

Als we code samenvoegen, gelden de volgende regels:

- alles wat boven de `setup` functie staat, moet daar blijven
- alles wat binnen de `setup` functie staat, moet binnen de `setup` functie blijven
- alles wat binnen de `draw` functie staat, moet binnen de `draw` functie blijven

Opdracht

- Voeg de code van ‘Bal die eeuwig naar rechts gaat’ samen met ‘Bal die eeuwig omlaag gaat’
- Verander de code zo dat de bal schuin gaat

Oplossing

```
float x = -50;
float y = -50;

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

void draw()
{
  ellipse(x,y,100,100);
  x = x + 1;
```

```

y = y + 1;
if (x > 350)
{
    x = -50;
}
if (y > 250)
{
    y = -50;
}
}

```

Eindopdracht

Laat de bal nu eeuwig schuin naar linksonder gaan.

Bal die horizontaal stuitert

In deze les gaan we een bal horizontaal laten stuiten.

Het ziet er zo uit:

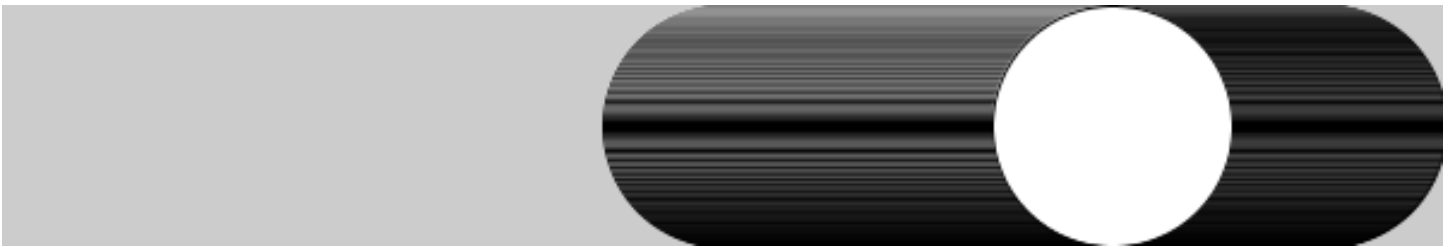


Figure 11: Bal die horizontaal stuitert (zie 'dojo/Images/BalDieHorizontaalStuitertGif')

We gaan in deze les twee variabelen en twee if-statements gebruiken.

Wat weten we al?

Dit is een eeuwig naar rechts gaande bal:

```

float x = 300;

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

void draw()
{
    ellipse(x,50,100,100);
    x = x + 1;
    if (x > 650)
    {
        x = -50;
    }
}

```

Vragen

- Wat doet dit programma?
- In welke richting beweegt de ovaal?
- Blijft de ovaal zichtbaar op het scherm?
- Kopieer de code en bekijk het programma. Klopt wat je dacht?
- Verander het programma: laat de bal nu eeuwig naar links gaan

Twee variabelen

Nu onthoudt de computer een variabele: de x-coördinaat van de ovaal. Om een bal te laten stuiteren, moet er ook een richting onthouden worden: de bal gaat immers of naar links of naar rechts.

In deze code wordt de richting van de bal **dx** genoemd. Dit is een afkorting van ‘delta x’ en dat is weer wiskundetaal voor ‘de verandering van x’.

Vragen

```
float x = 300;
float dx = 2;

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

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

- In welke richting beweegt de bal?
- Hoeveel pixels beweegt de bal per keer?
- Zet de waarde van dx op 1. Wat zie je?
- Zet de waarde van dx op 0. Wat zie je?
- Zet de waarde van dx op -1. Wat zie je?
- Bij sommige waarden van dx gaat de bal links het beeld uit. Maak een tweede if-statement, die ervoor zorgt dat de bal eeuwig links kan gaan. Test dit bij een dx van 2, 0 en -2.
- Wat moet er met de dx gebeuren om de bal te laten stuiteren? Probeer dit!

Stuiteren

Als een bal met een snelheid van drie pixels naar rechts gaat en stuitert, dan gaat deze vanaf dan drie pixels naar links. Andersom is dat ook zo.

Hier is een manier om de bal te laten stuiteren:

```
float x = 300;
float dx = 2;

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

```

void draw()
{
    ellipse(x,50,100,100);
    x = x + dx;
    if (x > 650)
    {
        dx = -2;
    }
    if (x < 50)
    {
        dx = 2;
    }
}

```

Vragen

- Kopieer en run de code. Stuitert de bal?
- Verander de snelheid van de bal naar een pixel per keer. Op hoeveel plekken moet je de code aanpassen?
- Verander de snelheid van de bal naar drie pixels per keer. Op hoeveel plekken moet je de code aanpassen?
- Verander de snelheid van de bal terug naar een pixel per keer. Op hoeveel plekken moet je de code aanpassen? Laat nu de bal in het begin naar links gaan. Op hoeveel plekken moet je de code aanpassen?

De richting omklappen

Er is een slimmere manier om `dx` te veranderen. We hebben gezien dat als `dx` gelijk was aan 2, deze -2 wordt bij bij een stuiter. We hebben gezien dat als `dx` gelijk was aan -2, deze 2 wordt bij bij een stuiter. Er komt een minnetje voor, of er komt een minnetje bij. Dit is gemakkelijk op dezelfde manier te doen:

```
dx = -dx;
```

Hiermee zeg je 'Lieve computer, de nieuwe waarde van `dx` is min de oude waarde'. Als de oude waarde van `dx` 2 is, dan wordt deze nu -2. Als de oude waarde van `dx` -2 is, dan wordt deze nu --2 (jawel, min min twee) en dat mag je schrijven als 2 (omdat min keer min is plus).

Eindopdracht

- Gebruik nu de slimme manier om een bal te laten stuiten.

Het lijkt al een beetje op Lonelier Pong. Dit is geen toeval :-)