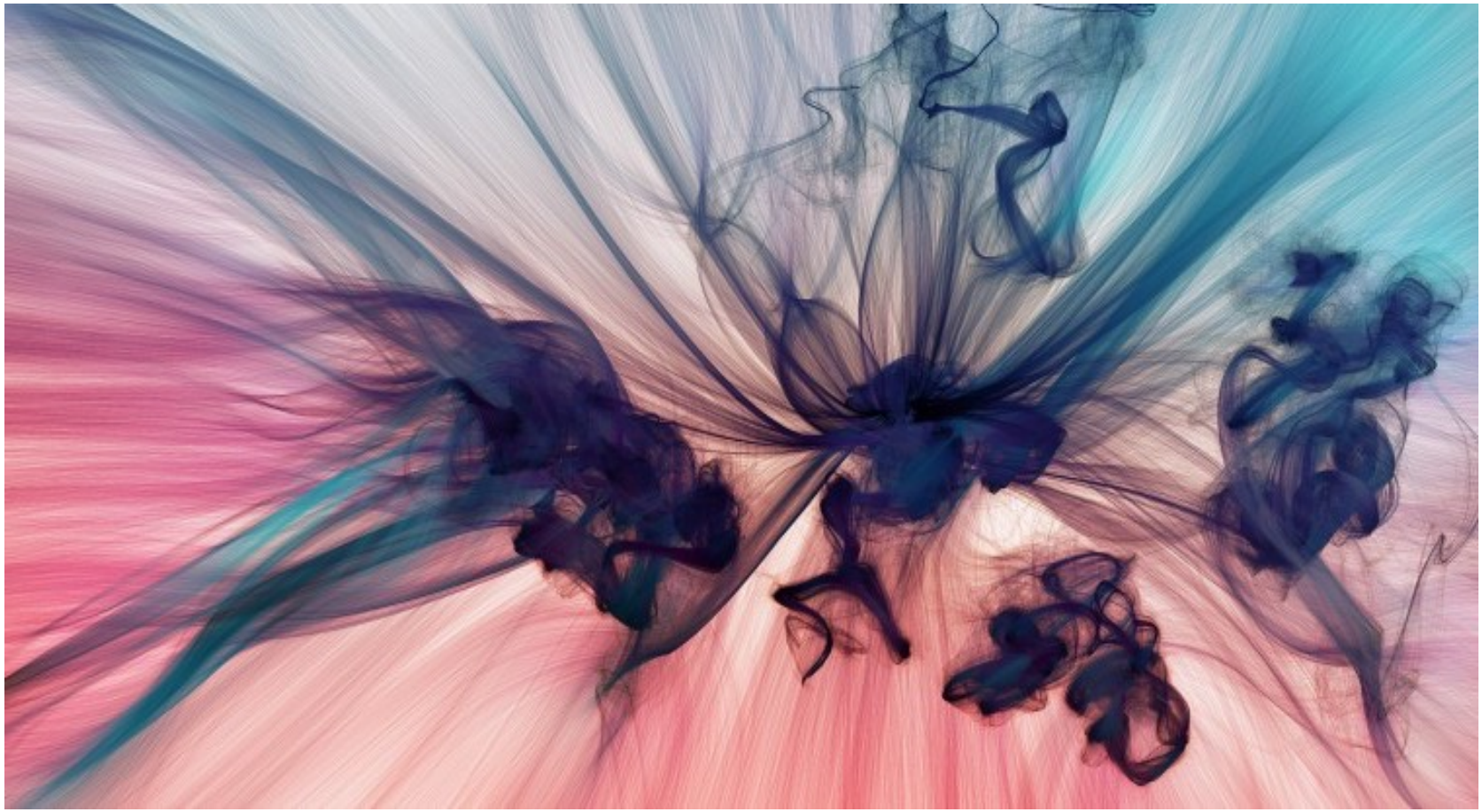
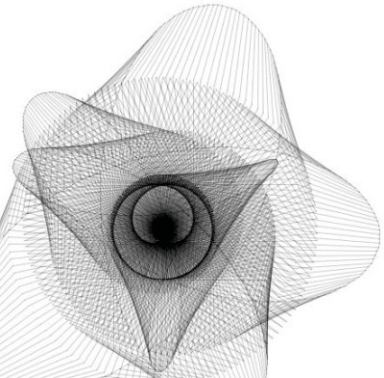
Met Processing kun je hele mooie tekeningen maken –
niet met stiften of potloden, maar door tegen de computer
te zeggen wat je wil zien.



J.R. Smith, Processing Posters



Start het programma Processing op.

In Processing zeg je eerst hoe groot het veld moet zijn waar je tekening in komt. Dat doe je zo:

```
size( 700, 500 );
```

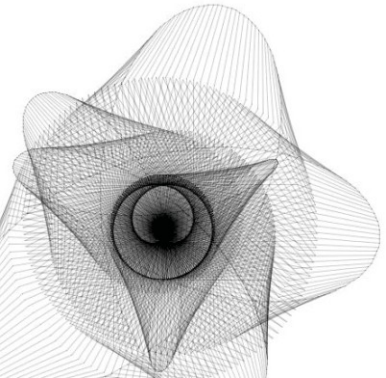
Als je nu linksboven op de -knop drukt, dan zie je een vierkant van 700 pixels breed en 500 pixels hoog.
Dat vierkant noemen we het canvas.

```
size( 700, 500 );
```

'size' is een woord dat Processing herkent om de grootte van het canvas (tekenblad) in te stellen.

Hoe breed en hoe hoog het canvas moet worden, geef je op met twee getallen met een komma ertussen.

Instructies sluit je altijd af met een puntkomma.

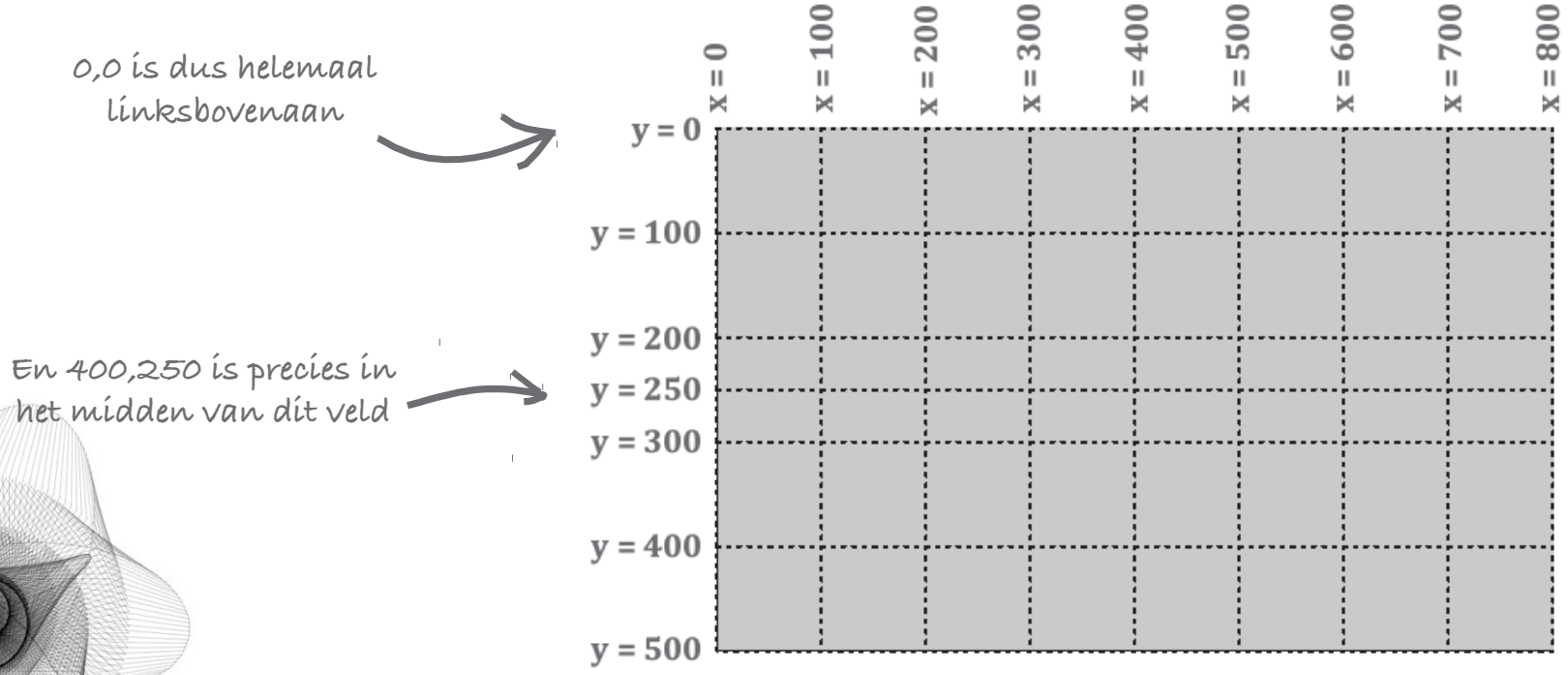


Coördinaten: plaats op het canvas

Het computerscherm waar je nu op kijkt, bestaat uit heel veel lichtpuntjes. Die puntjes noemen we pixels.

Het tekenblad van Processing bestaat ook uit pixels. Met twee getallen kun je aangeven waar je wil gaan tekenen:

Eén getal betekent 'hoe ver verwijderd van de linkerkant' (x)
Eén getal betekent 'hoe ver verwijderd van de bovenkant' (y)



Zo kun je bijvoorbeeld een lijntje tekenen:

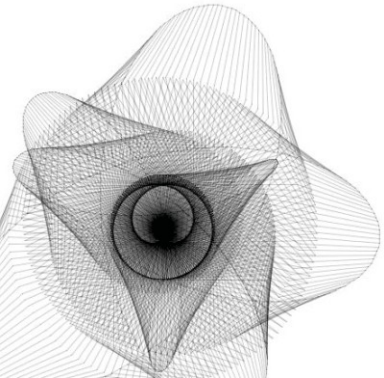
```
size( 700, 500 );
line( 200, 100, 500, 100 );
```

En druk op de  -knop.

```
line( 200, 100, 500, 100 );
```

Zo maak je dus een lijn die begint op 200 pixels van de linkerkant en 100 pixels van de bovenkant...

....en die doorloopt tot 500 pixels van de linkerkant en 100 pixels van de bovenkant.



Meer lijntjes tekenen

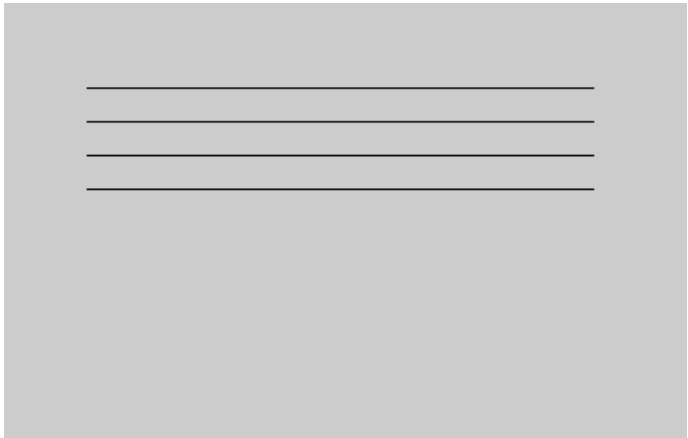
Als je meer lijntjes onder elkaar wil tekenen, dan kun je dat op twee manieren doen.

De eerste manier is de lijntjes stuk voor stuk toevoegen, dus zo:

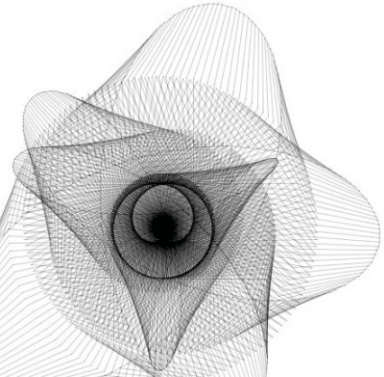
Code:

```
size(700, 500);


line( 200, 100, 500, 100 );
line( 200, 120, 500, 120 );
line( 200, 140, 500, 140 );
line( 200, 160, 500, 160 );
```

Resultaat:



Zie je dat de afstand van het lijntje ten opzichte van de bovenkant steeds 100 pixels groter wordt?

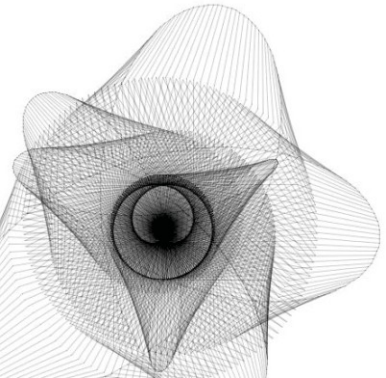


Automatisch tekenen

De andere manier is handiger als je heel veel lijntjes wil tekenen.

We gebruiken daarvoor variabelen. Variabelen zijn een manier om iets te onthouden. We kunnen bijvoorbeeld onthouden hoeveel lijntjes we al getekend hebben. Probeer dit maar eens:

```
size(700, 500);

int lijntjes = 0;                // de variabele 'lijntjes' onthoudt hoeveel lijntjes we hebben getekend.
int hoogte = 100;                // de variabele 'hoogte' onthoudt op welke hoogte het lijntje moet komen.

while (lijntjes < 21) {          // dit betekent dat we doorgaan met tekenen totdat we er 20 gedaan hebben.
  line(200, hoogte, 500, hoogte);  // teken een lijn.
  lijntjes = lijntjes + 1;       // onthouden dat we weer een lijntje gedaan hebben.
  hoogte = hoogte + 10;          // onthouden dat het volgende lijntje 10 pixels lager moet komen.
}
```



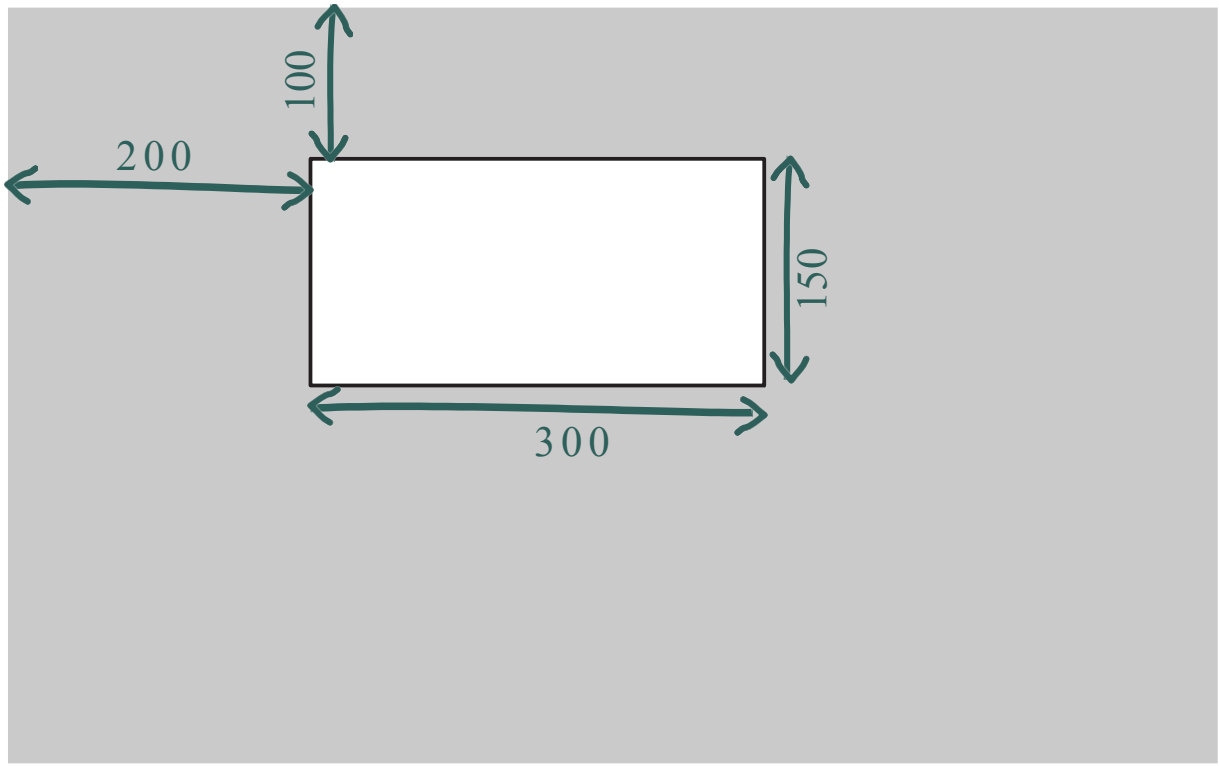
Verander de aantallen maar een paar keer om te zien wat er verandert!

Rechthoek tekenen (rect)

Instructies

```
size( 800, 500 );
rect( 200, 100, 300, 150 );
```

Resultaat



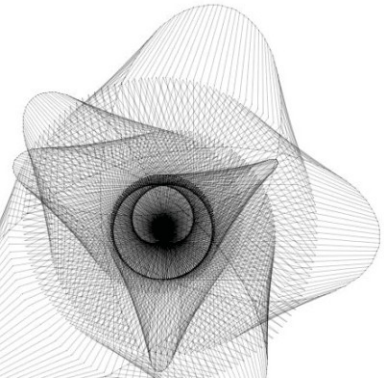
x-coördinaat van de linker-bovenhoek

y-coördinaat van de linker-bovenhoek

```
rect( 200, 100, 300, 150 );
```

Breedte van de rechthoek

Hoogte van de rechthoek

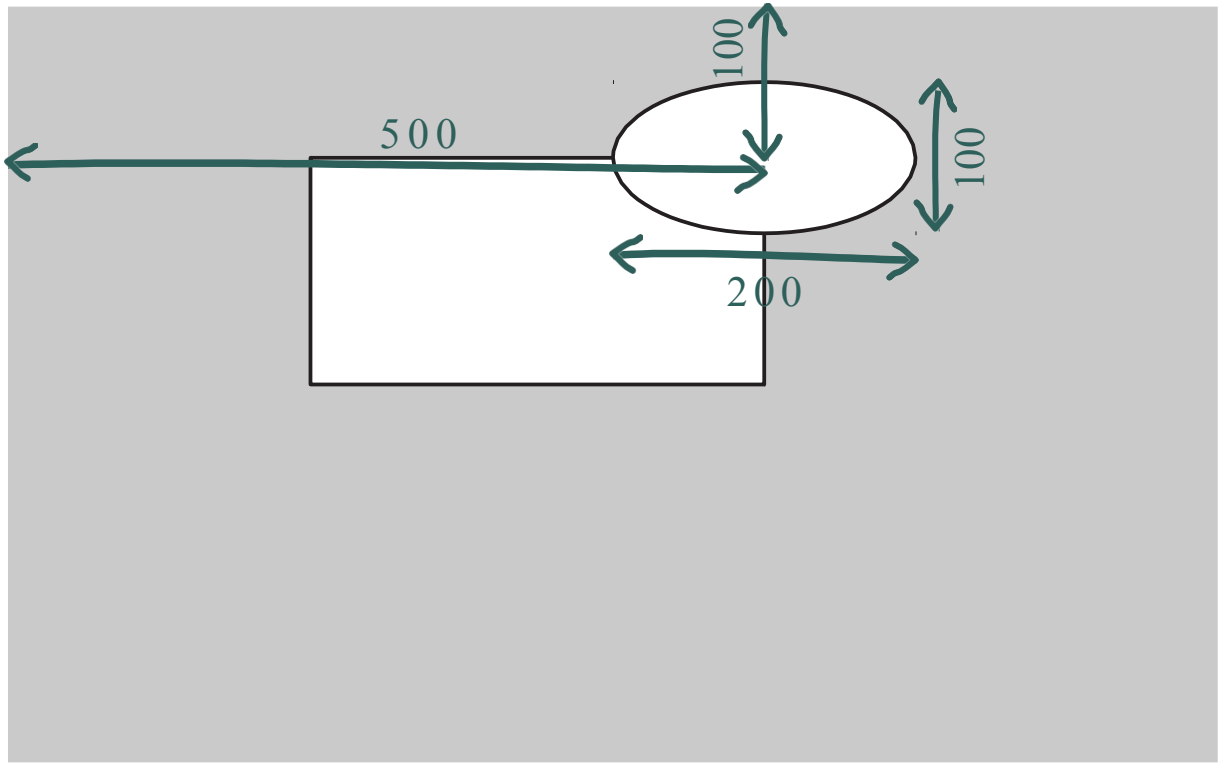


Ellips tekenen (ellipse)

Instructies

```
size( 800, 500 );
rect( 200, 100, 300, 150 );
ellipse( 500, 100, 200, 100 );
```

Resultaat



x-coördinaat
van het middelpunt

y-coördinaat
van het middelpunt

ellipse(500, 100, 200, 100);

Breedte van de ellips

Hoogte van de ellips

Kleur: achtergrond

Instructies

```
size( 800,  50 );
background(  0 );
```

```
size( 800, 50 );
background( 85 );
```

```
size( 800, 50 );
background( 170 );
```

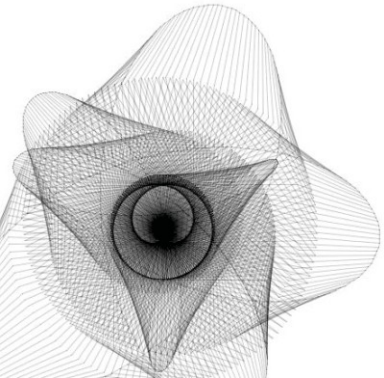
```
size( 800, 50 );
background( 255 );
```

Resultaat

Met de instructie 'background'
verander je de kleur van de achtergrond

background(100);

Een kleur aangeven met 1getal:
0 = zwart, 255 = wit



Kleur: Rood + Groen + Blauw = Alles

Instructies

```
size( 800, 50 );
background( 255, 0, 0 );
```

```
size( 800, 50 );
background( 0, 255, 0 );
```

```
size( 800, 50 );
background( 0, 0, 255 );
```

```
size( 800, 50 );
background( 242, 158, 41 );
```

```
size( 800, 50 );
background( 189, 47, 177 );
```

Resultaat

Een kleur aangeven met 3 getallen:

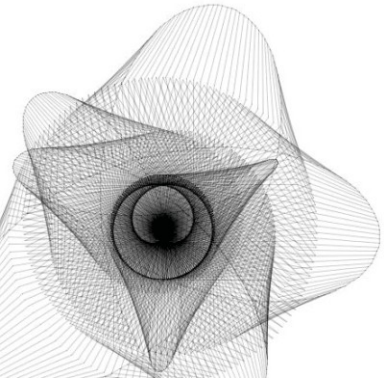
Hoeveel rood?
(0 = geen rood, 255 = maximaal rood)

Hoeveel blauw?
(0 = geen blauw, 255 = maximaal blauw)

```
background( 189, 47, 177 );
```

Hoeveel groen?
(0 = geen groen, 255 = maximaal groen)

RG



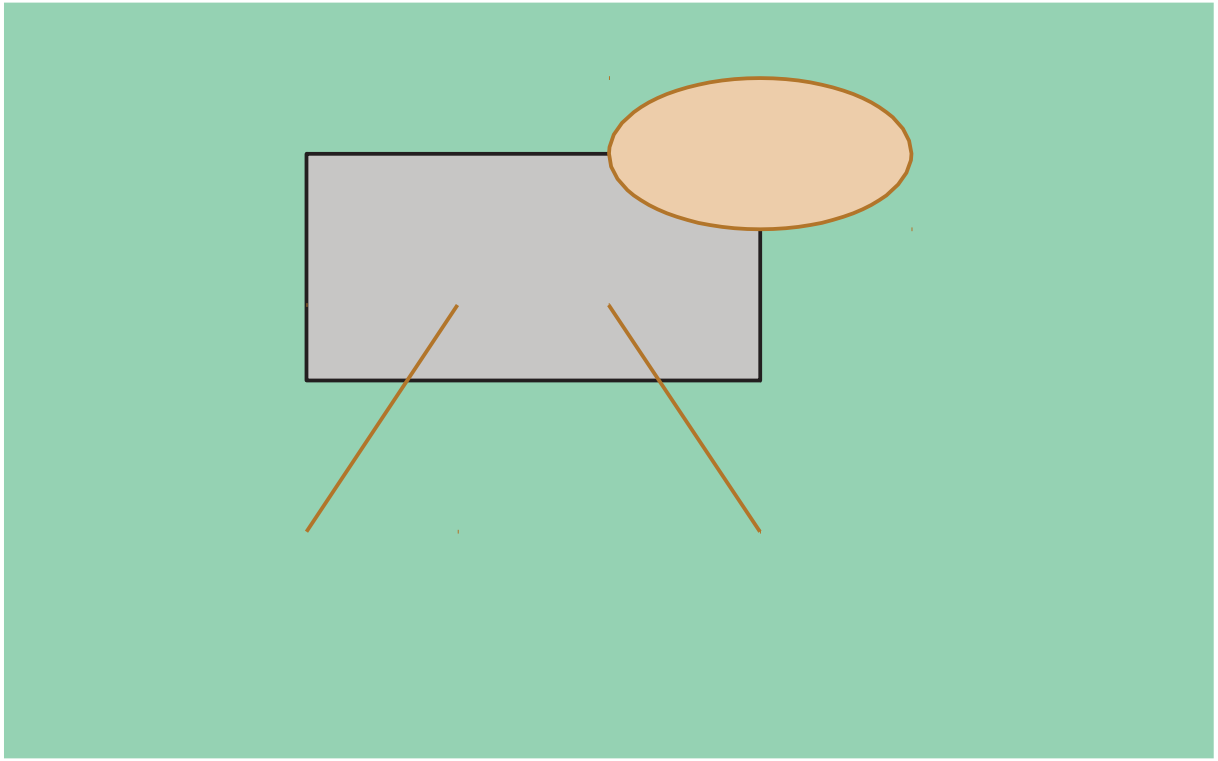
author: @kurtdevocht

Kleur - Achtergrond / Vulling / Omlijning

Instructies

```
size( 800, 500 );
background( 129, 255, 199 );

fill( 200 );
rect( 200, 100, 300, 150 );

stroke( 180, 122, 60 );
fill( 237, 206, 172 );
ellipse( 500, 100, 200, 100 );


  line( 300, 200, 200, 350 );
  line( 400, 200, 500, 350 );
```

Resultaat

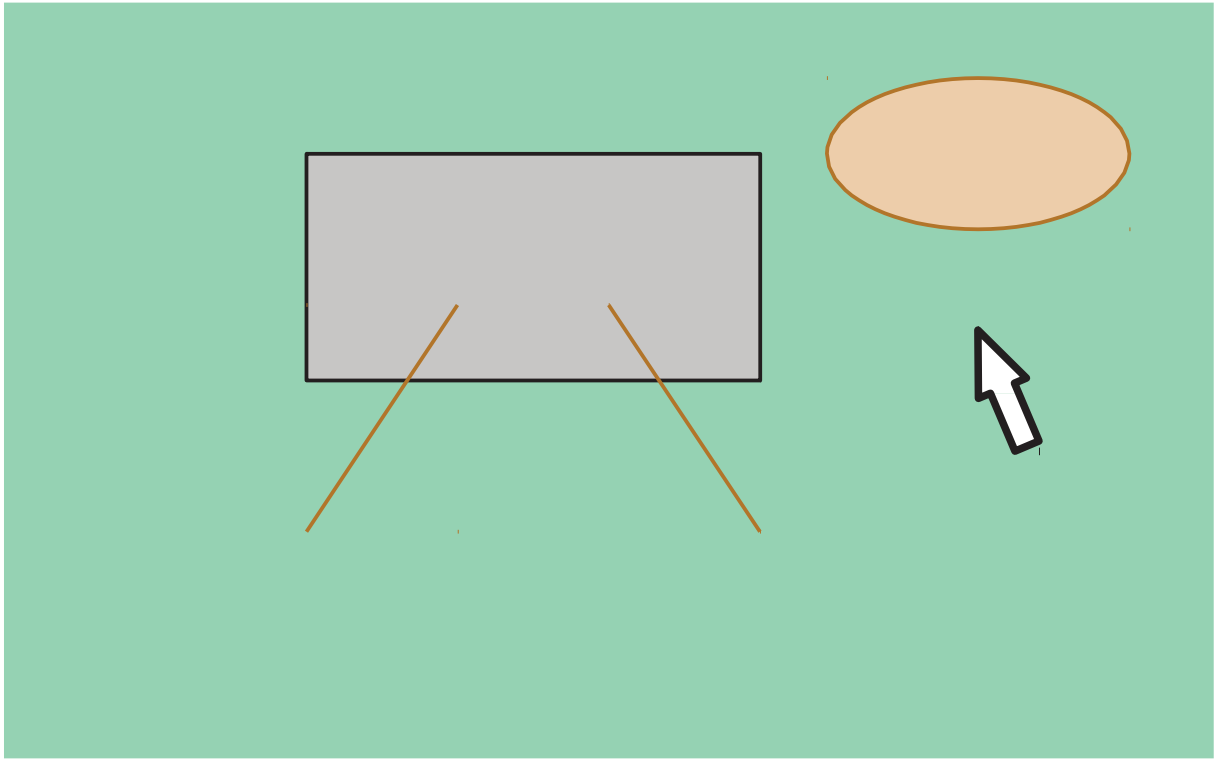


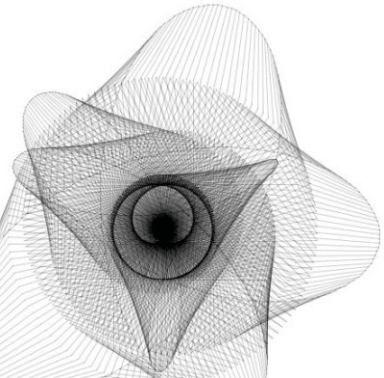
```
background( 129, 255, 199 );
fill( 200 );
stroke( 180, 122, 60 );
```

Instructies

```
void setup(){
  size( 800, 500 );
}

void draw(){
  background( 129, 255, 199 );

  fill( 200 );
  rect( 200, 100, 300, 150 );

  stroke( 180, 122, 60 );
  fill( 237, 206, 172 );
  ellipse( mouseX, 100, 200, 100 );


  line( 300, 200, 200, 350 );
  line( 400, 200, 500, 350 );
}
```

Resultaat





void setup()
{
}

Instructies die hier staan worden 1keer uitgevoerd als het programma start

void draw()
{
}

Instructies die hier staan worden steeds herhaald

mouse
x

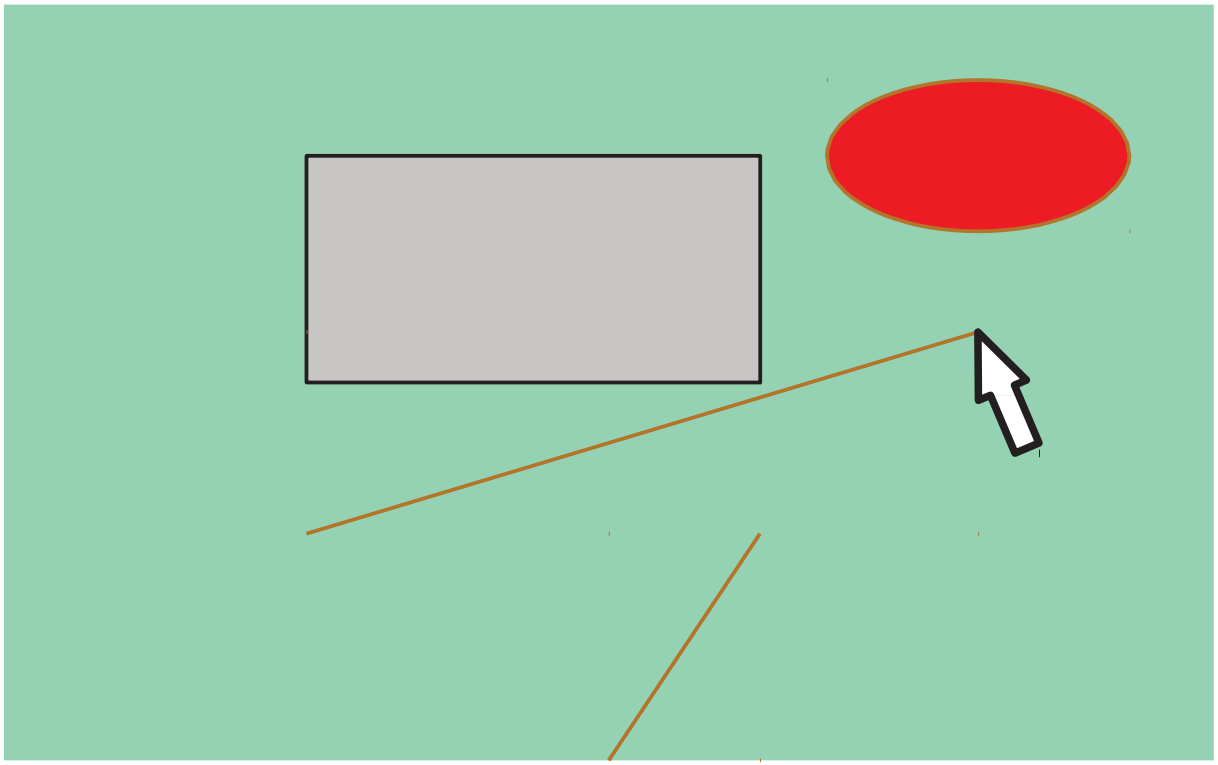
Een variabel getal dat steeds de x-coördinaat van de muisaanwijzer bevat

Meer animatie

Instructies

```
void setup(){
  size( 800, 500 );
}

void draw(){
  background( 129, 255, 199 );

  fill( random( 255 ) );
  rect( 200, 100, 300, 150 );

  stroke( 180, 122, 60 );
  fill( mouseX/ 2, 0, 0 );
  ellipse( mouseX, 100, 200,
  100 );

    line( mouseX, mouseY, 200, 350 );
    line( mouseY, mouseX, 500, 350 );
}
```

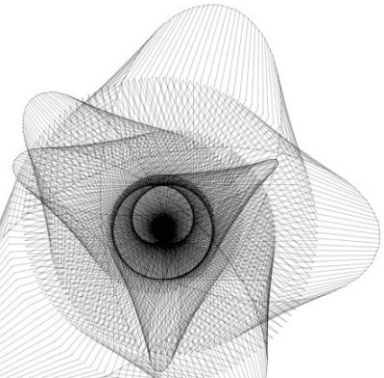
Resultaat



mouseX - Een variabel getal dat steeds de x-coördinaat van de muisaanwijzer bevat

mouseY - Een variabel getal dat steeds de y-coördinaat van de muisaanwijzer bevat

random(255) - Een willekeurig getal van 0 tot en met 255



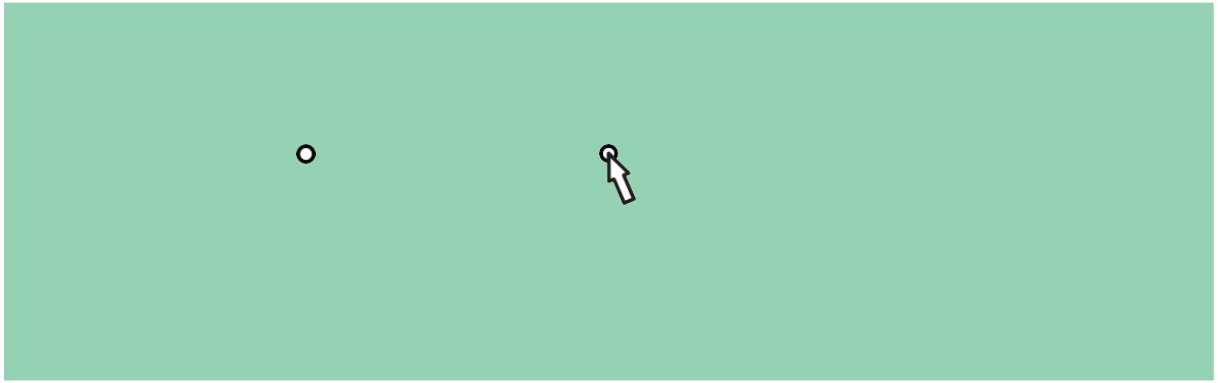
author: @kurtdevocht

Steeds een nieuwe achtergrond?

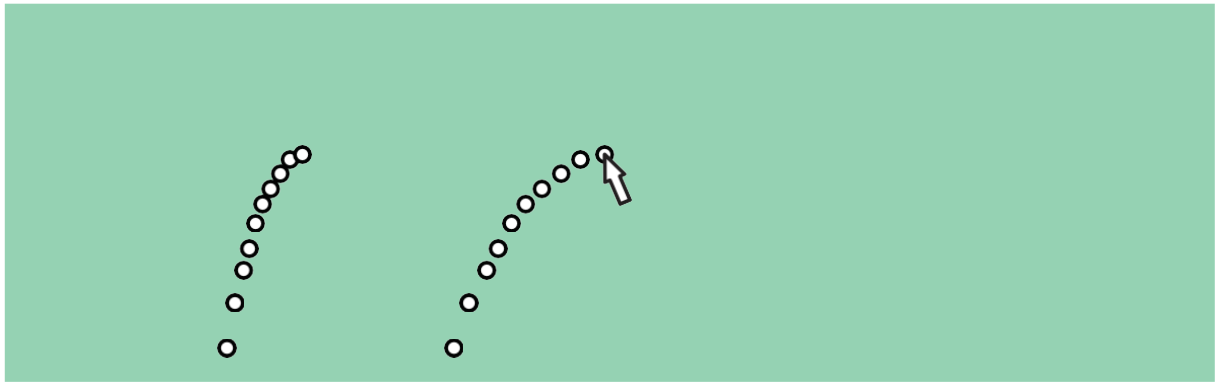
Instructies

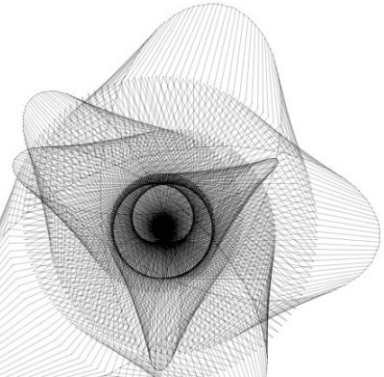
```
void setup(){
  size( 800, 250 );
}

void draw(){
  background( 129, 255, 199 );
  ellipse( mouseX, mouseY, 10, 10 );
  ellipse( mouseX / 2, mouseY, 10, 10 );
}
```

Resultaat



```
void setup(){
  size( 800, 250 );
  background( 129, 255, 199 );
}

void draw(){
  ellipse( mouseX, mouseY, 10, 10 );
  ellipse( mouseX / 2, mouseY, 10, 10 );
}
```





author: @kurtdevocht

Er kan nog heel veel meer met Processing. Je kunt geluiden laten horen, 3D-animaties maken, bewegende tekst laten zien en foto's bewerken.

Mooie voorbeelden vind je op deze websites:

http://www.aaronkoblin.com/work/flightpatterns/

http://formandcode.com/code-examples/simulate-particles

https://processing.org/exhibition/

Op de site van Processing zelf staan handleidingen en voorbeelden (wel alleen in het Engels):
https://processing.org/tutorials/
https://processing.org/examples/