





Cleve



Barcode

# Het TEAM

Hoi! Wij zijn Gemma, Max, Milène en Zoë  
van de opleiding ‘Communicatie en Multi-  
media-design’ in Maastricht.  
En je gaat met ons een wereldreis maken!  
Een wereldreis in code.

Gemma      Max      Milène      Zoë



# Op reis!

Geleen

Londen

New York

Rio de Janeiro

Hongkong

Parijs

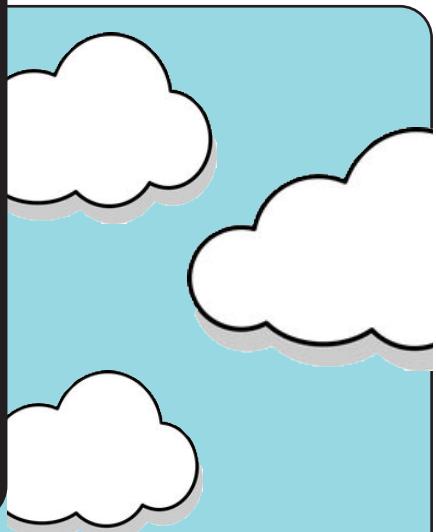
## de basis van coderen

Wij gaan in deze workshop aan de slag met de codetaal JavaScript. Een codetaal lijkt eigenlijk heel erg op een ‘normale’ vreemde taal, zoals Frans of Spaans of Engels.

Als je naar Frankrijk op vakantie gaat, zul je daar Frans moeten spreken zodat mensen je begrijpen. En wij moeten nu dus JavaScript tegen onze computer gaan ‘praten’ zodat onze computer ons begrijpt.



Hieronder zie je hoe p5.js eruitziet. Je ziet dat het bestaat uit cijfers, woorden en leestekens. Voor jou is deze taal nu nog moeilijk te lezen, maar de computer begrijpt het prima. Wat hieronder in code is geschreven, zijn instructies om een panda te tekenen – de panda die je net hebt gezien.



```
1▼ function setup() {
2    createCanvas(600, 600);
3}
4
5▼ function draw() {
6    background("lightblue");
7
8    //oren
9    fill(0)
10   ellipse(150,150,100,120)
11   ellipse(450,150,100,120)
12
13   //hoofd en lichaam
14   noStroke()
15   fill(255)
16   ellipse(300,300,400,400)
17   ellipse(300,700, 500,500)
18
19   //ogen en mond
20   fill(0)
21   ellipse(250, 250, 70, 90)
22   ellipse(350, 250, 70, 90)
23   ellipse(300, 400, 150, 100)
24
25   //mond vulling en pupillen
26   fill(255)
27   ellipse(300,380,150,100)
28   ellipse(260,250, 10,10)
29   ellipse(340,250,10,10)
30
31   //neus
32   fill(0)
33   ellipse(300, 350, 60, 30)
34 }
```

## volgorde

In p5.js is volgorde heel belangrijk. De computer 'leest' de code van boven naar beneden; wat er bovenaan je code staat, wordt als eerste getekend. Dat klinkt misschien lastig, maar er zijn veel dingen waarbij volgorde belangrijk is. Als je op reis gaat, bijvoorbeeld, is volgorde ook belangrijk:

- Kiezen waar je heen gaat  
reizen
- Vliegtickets bestellen
  - Koffer inpakken
  - Naar het vliegveld rijden
  - In het vliegtuig stappen
  - Uitstappen op je bestemming
  - Aan het strand gaan liggen

Je kunt dit lijstje niet van beneden naar boven lezen. Je kunt niet aan het strand gaan liggen als je nog niet bij het strand bent, en je kunt niet in het vliegtuig stappen als je nog geen vliegtickets hebt (niet legaal, in elk geval), en het wordt moeilijk om vliegtickets te bestellen als je nog niet weet waar je heen gaat.

P5.js werkt ook zo, alleen ziet een stappenplan er dan bijvoorbeeld zo uit:

Maak een canvas (een canvas kun je zien als een vel papier waarop je straks je tekeningen gaat maken)

- Geef aan hoe groot het canvas moet zijn
- Kies een kleur voor het canvas
- Kies een kleur die je voor je vormen gaat gebruiken
- Teken een vorm
- Geef aan waar je je vorm gaat tekenen en hoe groot je vorm gaat zijn

Ook dit lijstje kun je niet van beneden naar boven lezen. Je kunt de positie en grootte van je vorm niet aangeven als je vorm nog niet bestaat, en je kunt je vorm überhaupt niet tekenen als je geen canvas hebt om het op te tekenen.

Nu dat je dat weet, kunnen we aan de slag om zelf code te gaan schrijven!

# Een eerste p5.js-bestand

Voor het aanmaken van een p5.js-bestand ga je naar <https://editor.p5js.org/>. Daar staat een bestand al voor je klaar, maar voordat we er iets mee gaan doen, willen we dat je inlogt – zo kun je je werk straks opslaan. We hebben een account aangemaakt waarin de hele klas gaat werken. Je vindt de ‘log in’-knop rechtsboven in het scherm.

English ▾ Log in or Sign up

Je logt in met,  
e-mail: workshopCMD12@gmail.com  
wachtwoord: workshopcmd

Log In

Email or Username

Password

or

 Login with GitHub

 Login with Google

Don't have an account? Sign Up  
Forgot your password? Reset Password

# Hoe ziet het bestand eruit?

Zoals je ziet, is de code verdeeld in twee zogenoemde 'functions': setup en draw.

Alle code in setup wordt één keer uitgevoerd.

Alle code in draw wordt zestig keer per seconde uitgevoerd.

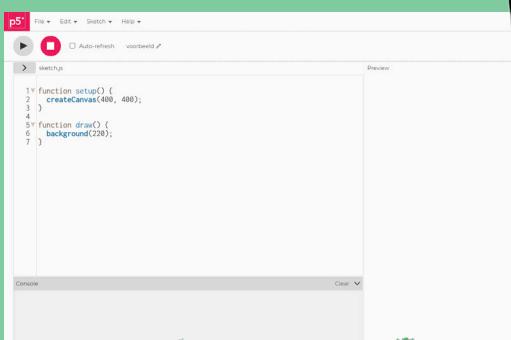
Het is voornamelijk belangrijk om draw te gebruiken als je iets maakt wat beweegt, maar het is geen probleem als je het gebruikt voor stilstaande vormen.

Aangezien wij vooral tekeningen gaan maken die stilstaan, maakt het eigenlijk niet uit of je je code in setup of draw zet, zolang je je code maar in één van de twee zet.

Links van de regels code zie je de regelnummers staan. Op regel 2 staat createCanvas. Tussen de haakjes staat '400, 400'. Dit is de grootte van het canvas. Het canvas is automatisch 400 pixels breed en 400 pixels hoog. Voor onze tekeningen werken we met een canvas van 600 bij 600 pixels. Kun je de code zo aanpassen dat het canvas 600 pixels breed en 600 pixels hoog is?

De setup en draw beginnen bij '{' en eindigen bij '}'. Die haakjes heten accolades. Als je iets in de setup of de draw wilt typen, typ je het dus ergens na de '{' en voor de '}'. Alles wat buiten de accolades getypt wordt, werkt niet.

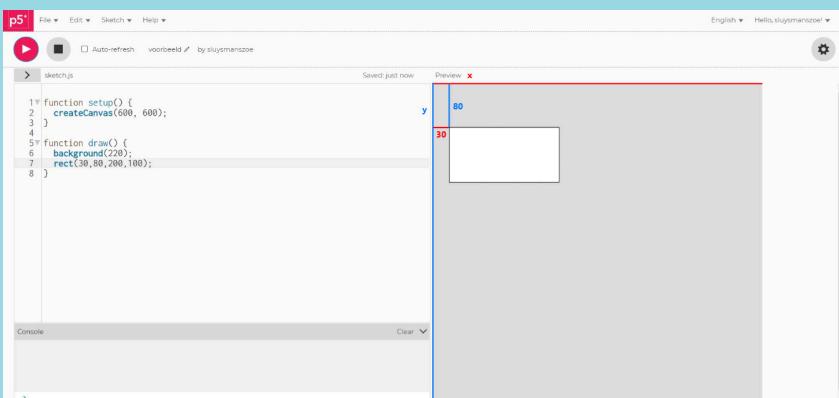
Door op de play-knop linksboven op het scherm te klikken, voer je je code uit.



```
p5.js
File Edit Sketch Help
sketch.pde
1// function setup() {
2  createCanvas(400, 400);
3 }
4
5// function draw() {
6  background(220);
7 }
```

# coordinaten

Gefeliciteerd – je canvas is klaar voor gebruik! Nu kunnen we onze eerste vorm gaan tekenen. Voor onze eerste vorm gaan we een rechthoek ('rectangle' in het Engels, of rect in code) tekenen.

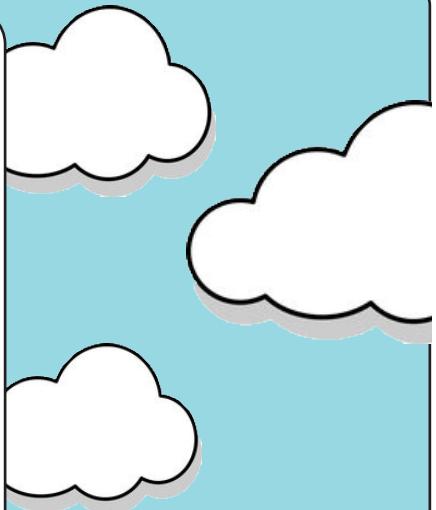


The screenshot shows the p5.js code editor interface. The code in the script pane is:

```
1 // function setup() {
2   createCanvas(600, 600);
3 }
4
5 // function draw() {
6   background(220);
7   rect(30,80,200,100);
8 }
```

The preview window shows a red border around the canvas area. A coordinate system is overlaid with a vertical blue line labeled 'y' and a horizontal red line labeled 'x'. A white rectangle is drawn on the canvas at the coordinates (30, 80, 200, 100). The top-left corner of the rectangle is at (30, 80), and its width is 200 and height is 100. The console pane at the bottom is empty.

Zoals je ziet, staan er bij 'rect' vier getallen tussen de haakjes. De eerste twee getallen bepalen de locatie van de rechthoek. Het eerste getal bepaalt hoeveel pixels de rechthoek van de linkerkant van het canvas af is. Dit noemen we 'x'. Deze rechthoek staat dus 30 pixels van de linkerkant van het canvas af.



Het tweede getal bepaalt hoeveel pixels de rechthoek van de bovenkant van het canvas af is. Dit noemen we 'y'. Deze rechthoek staat 80 pixels van de bovenkant van het canvas af. De laatste twee getallen bepalen hoe groot de rechthoek is: deze rechthoek is 200 pixels breed en 100 pixels hoog. Kort samengevat werkt een 'rect' dus zo: `rect(x, y, breedte, hoogte)`.

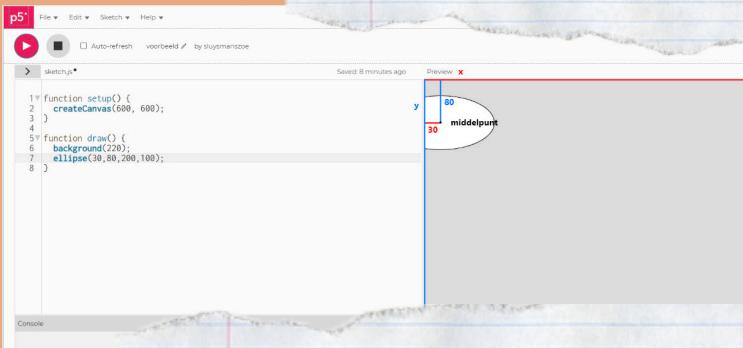


# Vormen

## cirkels

Rechthoeken zijn niet de enige vormen die je kunt tekenen. In onze workshop komen ook lijnen (line) en cirkels (ellipse) aan bod.

Een cirkel werkt hetzelfde als een rechthoek op het gebied van getallen: de getallen van een cirkel zijn ook ellipse(x, y, breedte, hoogte). Maar er is wel een belangrijk verschil.



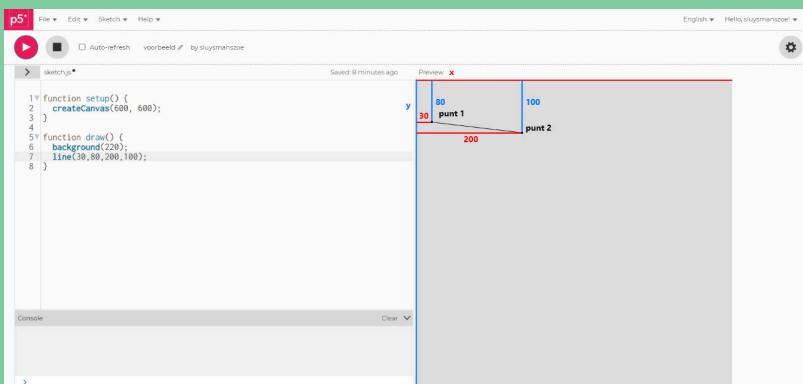
Deze cirkel heeft exact dezelfde getallen als de rechthoek, maar hij staat op een andere plek. De eerste twee getallen van 'ellipse' geven namelijk aan waar het middelpunt van de cirkel zich bevindt. Terwijl de eerste twee getallen van 'rect' aangeven waar de linkerbovenhoek van de rechthoek zich bevindt.

Deze cirkel heeft exact dezelfde getallen als de rechthoek, maar hij staat op een andere plek. De eerste twee getallen van 'ellipse' geven namelijk aan waar het middelpunt van de cirkel zich bevindt. Terwijl de eerste twee getallen van 'rect' aangeven waar de linkerbovenhoek van de rechthoek zich bevindt.

Het middelpunt van deze cirkel is dus 30 pixels van de linkerzij-kant en 80 pixels van de bovenkant van het canvas af. Dit is belang-rijk om te weten als je straks met cirkels gaat werken.

## Vormen lijnen

Een 'line' werkt anders: deze bestaat eigenlijk uit twee verschillende punten die door een lijn zijn verbonden. Jij tekent dus de twee uiteinden van de lijn; de computer tekent de lijn zelf.



De eerste twee getallen bepalen waar het eerste punt zich bevindt; in dit geval dus op 30 pixels afstand van de linkerkant van het canvas en op 80 pixels afstand van de bovenkant van het canvas. En de laatste twee getallen bepalen waar het tweede punt zich bevindt. In dit geval dus op 200 pixels afstand van de linkerkant van het canvas en op 100 pixels afstand van de bovenkant.



# kleuren

## woorden



Tot nu toe hebben we alleen maar saaie witte vormen getekend; het is tijd voor wat kleur!

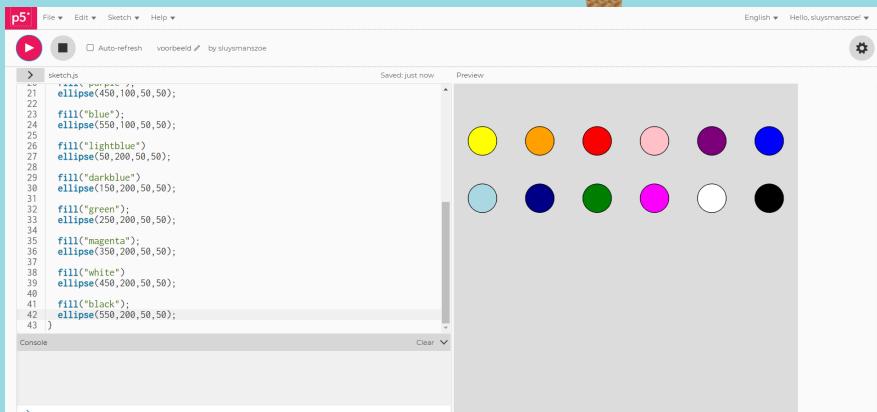
Bij een vorm breng je een kleur aan met `fill`. Bij de rand van een vorm of bij een line breng je een kleur aan met `stroke`. Er zijn drie manieren om kleur te gebruiken in p5.js: met woorden, met grijstinten en met rgb-kleurcodes.

The screenshot shows the p5.js code editor interface. The code in the sketch.js file is as follows:

```
1 // Function setup()
2 // createCanvas(600, 600);
3 //
4 // Function draw()
5 // background(220);
6 //
7 // fill("yellow");
8 // ellipse(50,100,50,50);
9 //
10 // fill("orange");
11 // ellipse(150,100,50,50);
12 //
13 // fill("red");
14 // ellipse(250,100,50,50);
15 //
16 // fill("pink");
17 // ellipse(350,100,50,50);
18 //
19 // fill("purple");
20 // ellipse(450,100,50,50);
21 //
22 // ... etc. ...
```

The preview window on the right displays a row of ten colored circles: yellow, orange, red, pink, purple, blue, light blue, dark blue, green, and black. The bottom right corner of the preview window features a small white cloud icon.

Woorden zijn het gemakkelijkst te gebruiken, maar de mogelijkheden zijn beperkt. Je kunt alleen de kleuren gebruiken die al voorgeprogrammeerd zijn in p5.js. Hieronder kun je zien hoe je de kleuren kunt gebruiken.



The screenshot shows the p5.js code editor interface. The code in the editor is as follows:

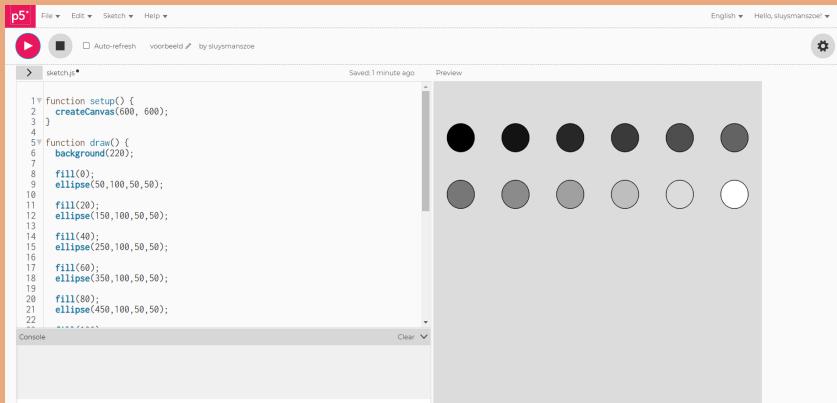
```
> sketch.js
21 ellipse(450, 100, 50, 50);
22 fill("blue");
23 ellipse(550, 100, 50, 50);
24 fill("lightblue")
25 ellipse(50, 200, 50, 50);
26 fill("darkblue")
27 ellipse(150, 200, 50, 50);
28 fill("green");
29 ellipse(250, 200, 50, 50);
30 fill("magenta");
31 ellipse(350, 200, 50, 50);
32 fill("white")
33 ellipse(450, 200, 50, 50);
34 fill("black");
35 ellipse(550, 200, 50, 50);
36
37
38
39
40
41
42
43 }
```

The preview window on the right displays a grid of colored circles corresponding to the fill() statements in the code. The colors include yellow, orange, red, pink, purple, blue, light blue, dark blue, green, magenta, white, and black. The preview shows a hot air balloon with a globe-like basket containing a small figure, set against a background of clouds.

# kleuren

## grijstinten

Grijstinten zijn wat lastiger; om een grijstint te gebruiken, typ je een getal in plaats van een kleur. 0 betekent zwart en 255 betekent wit – hoe hoger het getal, hoe lichter de kleur grijs. Hieronder staan wat voorbeelden.



The screenshot shows the p5.js code editor interface. The top bar includes tabs for File, Edit, Sketch, and Help, along with a preview button and user information. The main area displays a sketch titled "sketch.js" with the following code:

```
1 // function setup() {
2 //   createCanvas(600, 600);
3 // }
4
5 // function draw() {
6 //   background(220);
7 //
8 //   fill(0);
9 //   ellipse(50,100,50,50);
10
11  fill(20);
12  ellipse(150,100,50,50);
13
14  fill(40);
15  ellipse(250,100,50,50);
16
17  fill(60);
18  ellipse(350,100,50,50);
19
20  fill(80);
21  ellipse(450,100,50,50);
22
23 }
```

To the right of the code, a preview window shows a horizontal row of seven grayscale circles, each corresponding to one of the ellipses in the code. The circles range from black (representing 0) to white (representing 255). Below the preview is a gray panel, and at the bottom is a console window labeled "Console" with a "Clear" button.



p5.js

File ▾ Edit ▾ Sketch ▾ Help ▾

sketch.js

Auto-refresh voorbeeld by sluymanszoe

Saved: just now

Preview

```
1 sketch();
2
3   ellipse(450,100,50,50);
4   fill(100);
5   ellipse(550,100,50,50);
6   fill(100);
7   ellipse(50,200,50,50);
8
9   fill(140);
10  ellipse(150,200,50,50);
11
12  fill(160);
13  ellipse(250,200,50,50);
14
15  fill(180);
16  ellipse(350,200,50,50);
17
18  fill(220);
19  ellipse(450,200,50,50);
20
21  fill(255);
22  ellipse(550,200,50,50);
23
24 }
```

Console

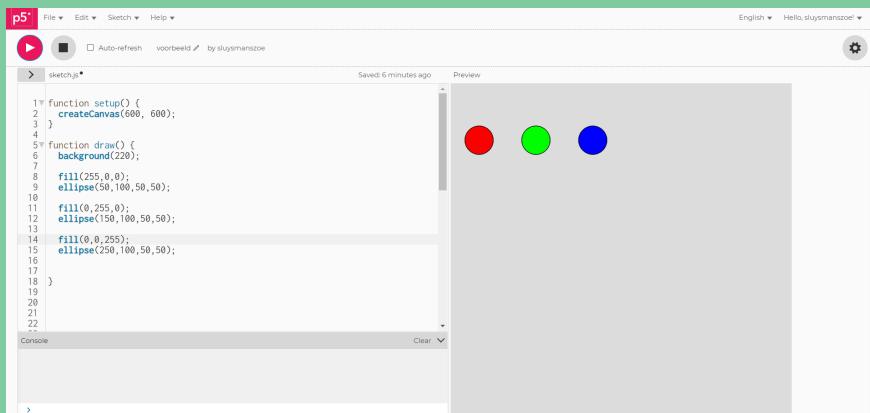
Clear

# kleuren

## RGB

Met rgb gaan we in onze workshop niet per sé aan de slag – je kunt de workshop prima maken zonder rgb-codes te gebruiken – maar als je specifieke kleuren wilt gebruiken, is het misschien handig om te gebruiken. De afkorting ‘rgb’ staat voor ‘red, green, blue’. Een rgb-code bestaat uit drie getallen: het eerste getal staat voor de hoeveelheid rood, het tweede getal voor de hoeveelheid groen, en het derde getal voor de hoeveelheid blauw.

Een hoeveelheid kan ergens tussen 0 en 255 zijn. 0 betekent dat je niets van die kleur gebruikt. Hieronder zie je hoe je rood, groen en blauw maakt.

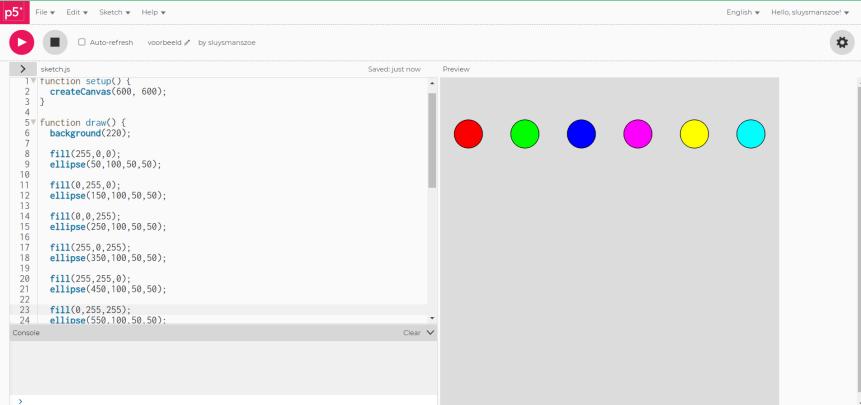


The screenshot shows the p5.js code editor interface. The code in the script pane is:

```
1 // function setup() {
2   // createCanvas(600, 600);
3 }
4
5 // function draw() {
6   background(220);
7
8   fill(255, 0, 0);
9   ellipse(50, 100, 50, 50);
10
11   fill(0, 255, 0);
12   ellipse(150, 100, 50, 50);
13
14   fill(0, 0, 255);
15   ellipse(250, 100, 50, 50);
16
17 }
18
19
20
21
22 }
```

The preview window shows three colored circles: red, green, and blue.

Het grote voordeel van rgb-codes is dat je kleuren kunt mengen. Blauw en rood maken samen paars. Blauw en groen maken samen turquoise. En rood en groen maken samen geel (dat klinkt onlogisch, maar helaas is het zo).

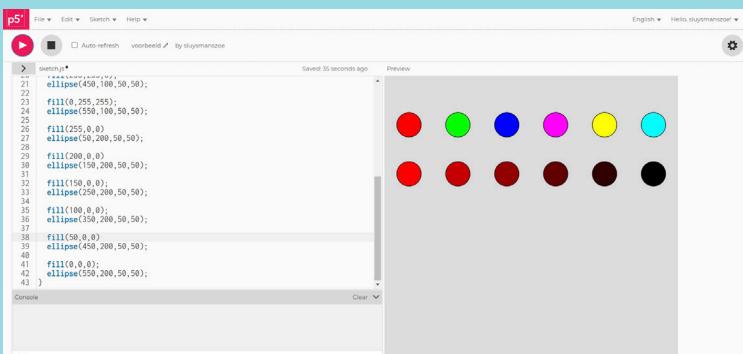


The screenshot shows the p5.js code editor interface. The code in the editor is as follows:

```
> sketch.js
1 function setup() {
2   createCanvas(600, 600);
3 }
4
5 function draw() {
6   background(220);
7
8   fill(255, 0, 0);
9   ellipse(50, 100, 50, 50);
10
11  fill(0, 255, 0);
12  ellipse(150, 100, 50, 50);
13
14  fill(0, 0, 255);
15  ellipse(250, 100, 50, 50);
16
17  fill(255, 0, 255);
18  ellipse(350, 100, 50, 50);
19
20  fill(255, 255, 0);
21  ellipse(450, 100, 50, 50);
22
23  fill(0, 255, 255);
24  ellipse(550, 100, 50, 50);
```

The preview window on the right displays six colored circles arranged horizontally: red, green, blue, magenta, yellow, and cyan.

Nog een groot voordeel van rgb is dat je met de felheid van kleuren kunt spelen. 255 is het felste dat een kleur kan worden, maar hoe lager het getal, hoe donkerder de kleur. Hieronder is dat gedemonstreerd met rood, maar het kan met alle kleuren.



The screenshot shows the p5.js code editor interface. On the left, the code editor displays the following sketch.js code:

```
function setup() {
  createCanvas(400, 200);
}

function draw() {
  fill(255, 0, 0);
  ellipse(450, 100, 50, 50);
  fill(255, 255, 255);
  ellipse(550, 100, 50, 50);
  fill(255, 0, 0);
  ellipse(50, 200, 50, 50);
  fill(200, 0, 0);
  ellipse(200, 200, 50, 50);
  fill(150, 0, 0);
  ellipse(150, 200, 50, 50);
  fill(100, 0, 0);
  ellipse(350, 200, 50, 50);
  fill(50, 0, 0);
  ellipse(450, 200, 50, 50);
  fill(0, 0, 0);
  ellipse(550, 200, 50, 50);
}
```

The right side of the interface features a preview window showing a grid of colored circles. Below the preview is a small hot air balloon illustration. A white cloud icon is positioned at the bottom right of the preview area.

Hieronder nog wat voorbeelden van mooie kleuren die je misschien leuk vindt om te gebruiken. Door de getallen lager of hoger te maken, kun je zelf de kleuren aanpassen aan je eigen smaak!

The screenshot shows the p5.js code editor with a sketch titled "voorbeeld". The code defines a setup function that creates a canvas and a draw function that repeatedly draws colored ellipses at various coordinates. The preview window shows a grid of colored circles.

```
function setup() {
  createCanvas(600, 600);
}

function draw() {
  background(220);
  fill(241,195,209);
  ellipse(56,100,50,50);
  fill(209,141,179);
  ellipse(158,100,50,50);
  fill(201,147,212);
  ellipse(250,100,50,50);
  fill(159,163,227);
  ellipse(350,100,50,50);
  fill(112,149,225);
  ellipse(450,100,50,50);
}
```

The screenshot shows the p5.js code editor with a sketch titled "voorbeeld". The code defines a setup function that creates a canvas and a draw function that repeatedly draws colored ellipses at various coordinates. The preview window shows a grid of colored circles.

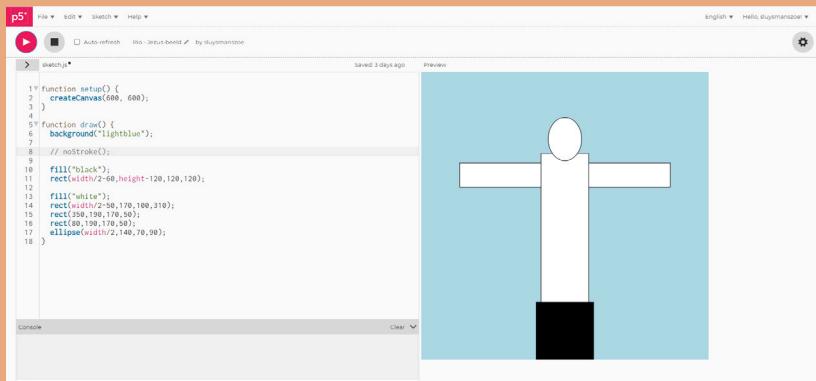
```
ellipse(450,100,50,50);
ellipse(169,172,208);
ellipse(558,100,50,50);
ellipse(81,192,191);
ellipse(50,200,50,50);
ellipse(178,226,137);
ellipse(150,200,50,50);
ellipse(255,179,70);
ellipse(250,200,50,50);
ellipse(249,205,151);
ellipse(250,200,50,50);
fill(255,176,145);
ellipse(450,200,50,50);
fill(255,109,98);
ellipse(559,200,50,50);
```

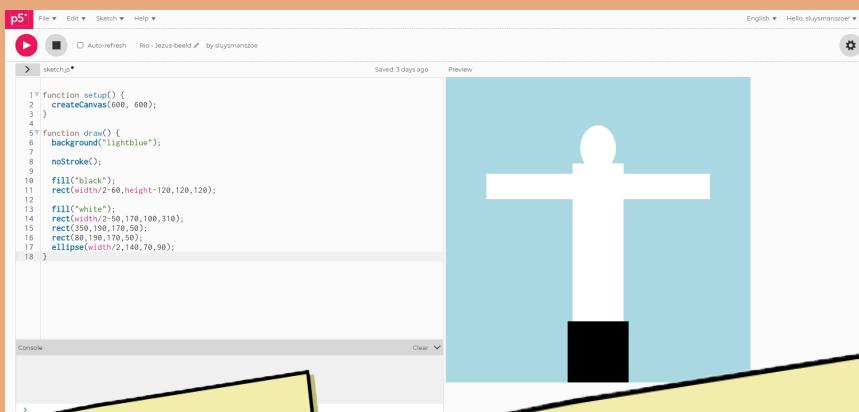
# finaltouches

## Strokes

Je kunt nu intussen mooie vormen tekenen en die vormen een leuke kleur geven, maar misschien ziet je tekening er nog steeds niet uit zoals je wilt dat hij eruitziet. Misschien vind je de randen om de vormen heen niet mooi, of misschien wil je je vorm doorzichtig maken. Ook daar bestaan gemakkelijke oplossingen voor!

In onze ervaring is **noStroke** de optie die je het vaakste zult gebruiken. noStroke haalt de randen om een vorm weg. Vooral als je meerdere vormen wilt gebruiken om één tekening te maken, is dit handig. Hieronder zie je twee versies van dezelfde tekening; eerst zonder noStroke, daarna met noStroke.

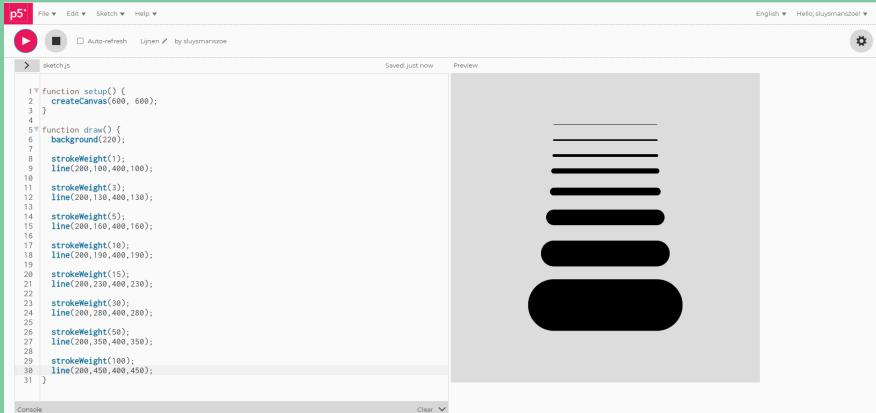




Met noStroke moet je wel oppassen dat het 'line's ook laat verdwijnen. Zorg er dus voor dat je 'line's in de code boven je noStroke komen te staan.

De '//'s die je in het eerste screenshot voor 'noStroke' ziet staan, betekenen dat die regel code commentaar is. Als er twee schuine strepen voor de code staan, wordt die regel code niet uitgevoerd. Zo kun je dus ook opmerkingen opschrijven zonder dat de computer er iets mee doet!

Met strokeWeight geef je de dikte van je rand of line aan, in pixels. Standaard staat de strokeWeight op 1 pixel. In onze workshop wordt de strokeWeight meestal niet hoger dan 5 pixels, maar je kunt de strokeWeight zo hoog maken als je wilt. Hieronder wat voorbeelden.



The screenshot shows the p5.js code editor interface. The top bar includes the p5 logo, File, Edit, Sketch, Help, Auto-refresh, Lijnen, and a user profile. The main area has tabs for sketch.js and Preview. The code in sketch.js is as follows:

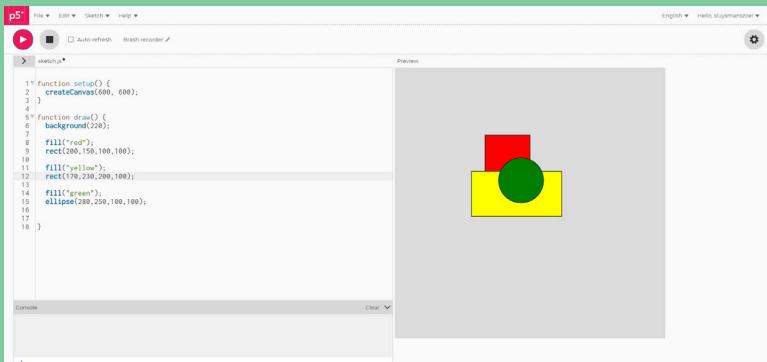
```
1 // Function setup() {
2   createCanvas(600, 600);
3 }
4
5 // Function draw() {
6   background(220);
7
8   strokeWeight(1);
9   line(200, 100, 400, 100);
10
11  strokeWeight(3);
12  line(200, 130, 400, 130);
13
14  strokeWeight(5);
15  line(200, 160, 400, 160);
16
17  strokeWeight(10);
18  line(200, 190, 400, 190);
19
20  strokeWeight(15);
21  line(200, 230, 400, 230);
22
23  strokeWeight(30);
24  line(200, 280, 400, 280);
25
26  strokeWeight(50);
27  line(200, 350, 400, 350);
28
29  strokeWeight(100);
30  line(200, 450, 400, 450);
31 }
```

The Preview window shows a vertical stack of horizontal lines of increasing thickness from top to bottom, starting at 1 pixel and ending at 100 pixels. Below the code editor is a green decorative footer featuring a row of stylized trees.

# final touches

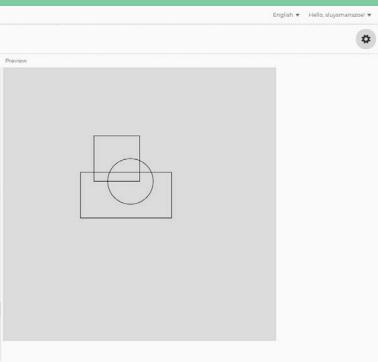
## fill

Met noFill kun je de vulling van een vorm weghalen. Dit is handig als je alleen de rand van een vorm over wilt houden.



```
p5.js Sketch: sketch.js
File Edit Sketch Help English Hello, sketchname! 
sketch.js
1// Function setup() {
2  // createCanvas(600, 600);
3 }
4
5// Function draw() {
6  background(220);
7
8  fill("red");
9  rect(150,150,100,100);
10
11  fill("yellow");
12  rect(179,239,209,100);
13
14  fill("green");
15  ellipse(289,259,100,100);
16
17
18 }
```

Hier zijn drie vormen met een fill...



```
p5.js Sketch: sketch.js
File Edit Sketch Help English Hello, sketchname! 
sketch.js
1//function setup() {
2  //createCanvas(600, 600);
3 }
4
5//function draw() {
6  //background(220);
7
8  rect(150,150,100,100);
9  noFill();
10
11  rect(179,239,209,100);
12  noFill();
13
14  noFill();
15  ellipse(289,259,100,100);
16
17
18 }
```

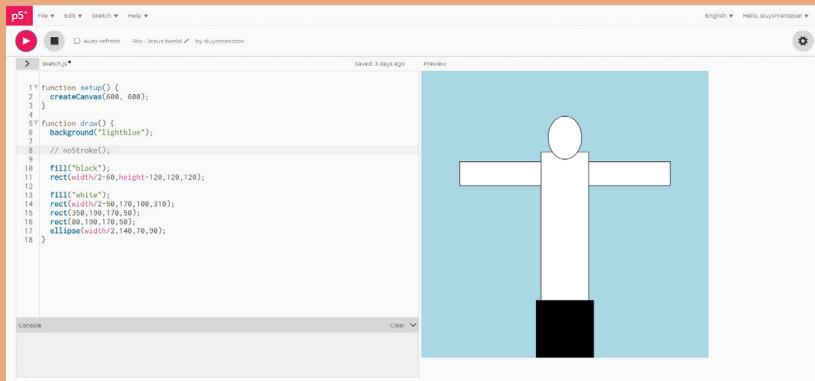
...en hier zijn diezelfde vormen zonder fill.

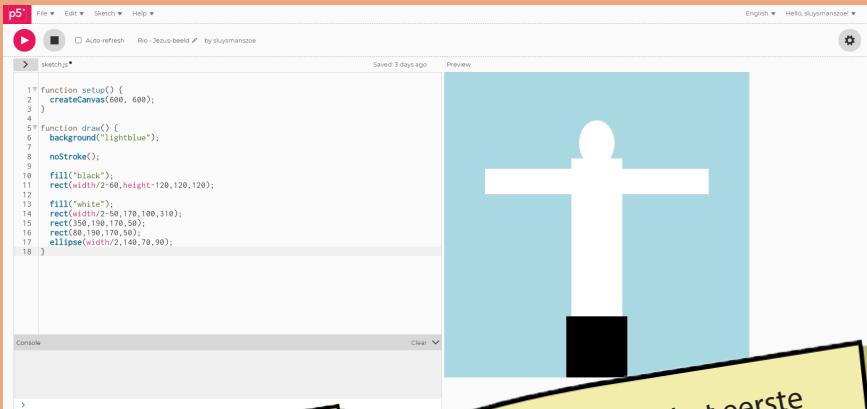
# finaltouches

## Strokes

Je kunt nu intussen mooie vormen tekenen en die vormen een leuke kleur geven, maar misschien ziet je tekening er nog steeds niet uit zoals je wilt dat hij eruitziet. Misschien vind je de randen om de vormen heen niet mooi, of misschien wil je je vorm doorzichtig maken. Ook daar bestaan gemakkelijke oplossingen voor!

In onze ervaring is **noStroke** de optie die je het vaakste zult gebruiken. noStroke haalt de randen om een vorm weg. Vooral als je meerdere vormen wilt gebruiken om één tekening te maken, is dit handig. Hieronder zie je twee versies van dezelfde tekening; eerst zonder noStroke, daarna met noStroke.



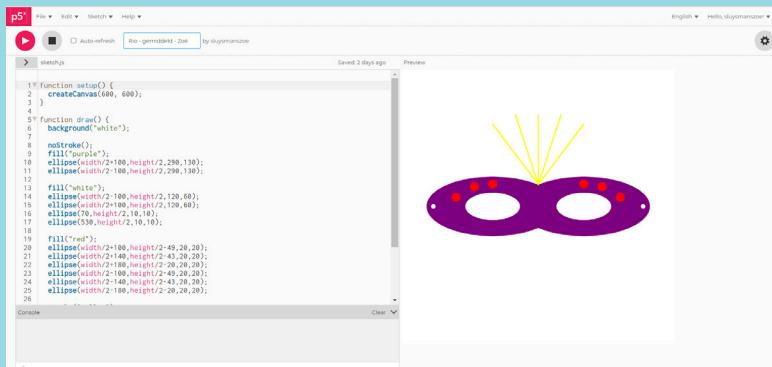


Met noStroke moet je wel oppassen dat het 'line's ook laat verdwijnen. Zorg er dus voor dat je 'line's in de code boven je noStroke komen te staan.

De '//'s die je in het eerste screenshot voor 'noStroke' ziet staan, betekenen dat die regel code commentaar is. Als er twee schuine strepen voor de code staan, wordt die regel code niet uitgevoerd. Zo kun je dus ook opmerkingen opschrijven zonder dat de computer er iets mee doet!

# Code opslaan

Als je je tekening af hebt, wil je 'm natuurlijk niet kwijtraken. (Natuurlijk wil je 'm niet kwijtraken! Hij is vast supermooi!) Om je tekening op te kunnen slaan, moet je hem eerst een titel geven. In deze workshop willen we graag dat je je tekening opslaat met de naam van de stad, de moeilijkheidsgraad en jouw naam. Dus als je de gemakkelijke tekening van Londen hebt gemaakt, is je titel: 'Londen – gemakkelijk – jouw naam'.

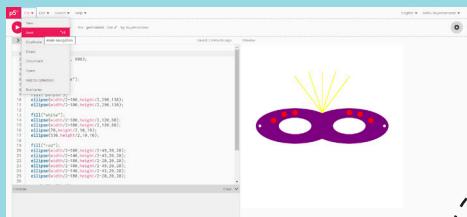


The screenshot shows the p5.js code editor with the following code:

```
1 //function setup() {
2 //    createCanvas(600, 600);
3 //}
4 function draw() {
5     background("white");
6     noStroke();
7     fill('purple');
8     ellipse(cwidth/2+100,height/2,290,130);
9     ellipse(cwidth/2+100,height/2,290,130);
10    fill('cyan');
11    ellipse(cwidth/2+100,height/2,120,60);
12    ellipse(cwidth/2+100,height/2,120,60);
13    ellipse(cwidth/2+100,height/2,120,60);
14    ellipse(cwidth/2+100,height/2,120,60);
15    fill('red');
16    ellipse(cwidth/2+100,height/2-49,29,20);
17    ellipse(cwidth/2+100,height/2-49,29,20);
18    ellipse(cwidth/2+100,height/2-49,29,20);
19    ellipse(cwidth/2+100,height/2-49,29,20);
20    ellipse(cwidth/2+100,height/2-49,29,20);
21    ellipse(cwidth/2+100,height/2-49,29,20);
22    ellipse(cwidth/2+100,height/2-49,29,20);
23    ellipse(cwidth/2+100,height/2-49,29,20);
24    ellipse(cwidth/2+100,height/2-49,29,20);
25    ellipse(cwidth/2+100,height/2-49,29,20);
26}
```

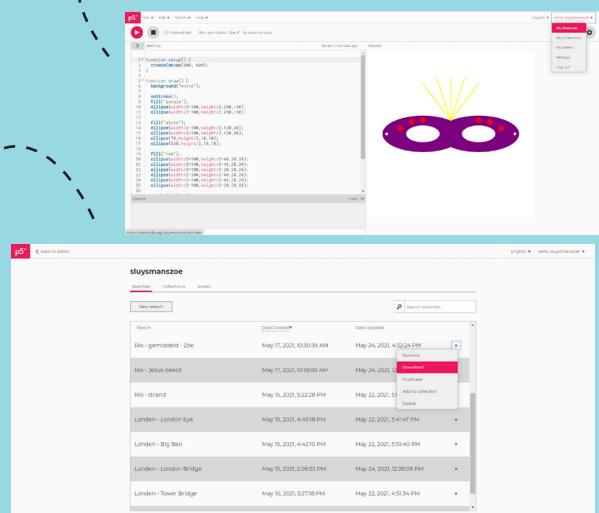
The preview window shows a purple mask with yellow eyes and red pupils. A small green hot air balloon is visible in the background.

Het is aan te raden om je tekening een titel te geven zodra je begint en tussendoor op te slaan. Linksboven kun je de knop 'file' uitklappen en kun je daaronder op 'save' klikken. Dan wordt je tekening opgeslagen.



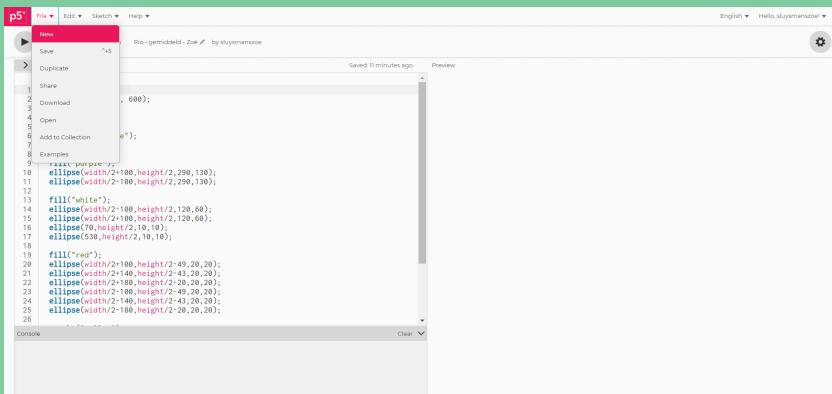
Als je tekening helemaal af is, kun je 'm downloaden. Om dat te doen, klap je de knop rechtsboven uit en klik je op 'My Sketches'.

Je komt dan uit bij de plek waar de bestanden van de klas staan opgeslagen. Je zoekt het bestand met de juiste stad en je eigen naam. (Het nieuwste bestand staat bovenaan, dus je bestand zou redelijk bovenaan moeten staan.) Als je rechts op het pijltje klikt bij dat bestand, klapt er een schermpje uit. Daar kun je op 'Download' klikken.



Je tekening wordt dan gedownload. Gefeliciteerd! Je kunt je tekening nu niet meer kwijtraken!

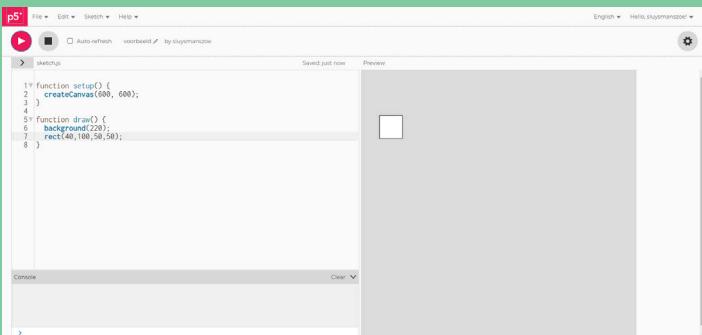
Om een nieuw bestand aan te maken, kun je het beste op 'Back to Editor' klikken (linksboven) en dan onder 'file' op 'new' klikken.



```
p5.js File Edit Sketch Help
New
Save
Duplicate
Share
Download
Open
Add to Collection
Examples
raxx - pompoen
ellipse(width/2-100,height/2,298,130);
ellipse(width/2-100,height/2,298,130);
ellipse(width/2-100,height/2,298,130);
fill("white");
ellipse(width/2-100,height/2,120,60);
ellipse(width/2-100,height/2,120,60);
ellipse(0,height/2,10,10);
ellipse(0,height/2,10,10);
fill("red");
ellipse(width/2+100,height/2-49,20,20);
ellipse(width/2+140,height/2-43,20,20);
ellipse(width/2+180,height/2-49,20,20);
ellipse(width/2+140,height/2-43,20,20);
ellipse(width/2+180,height/2-28,20,20);
```

# meer vromen

Als de breedte en hoogte van een rect gelijk zijn, heb je een vierkant.

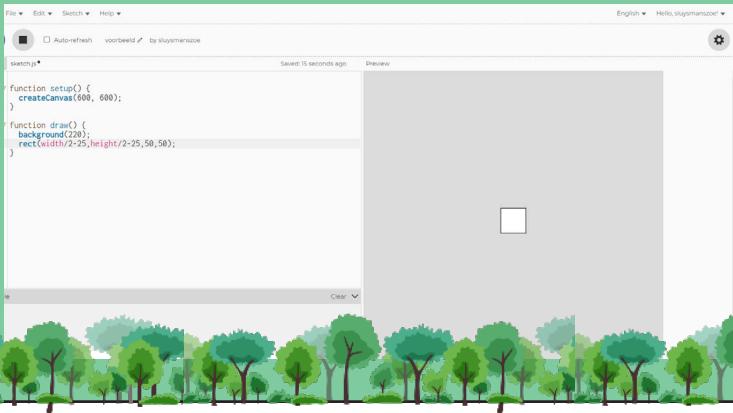


The screenshot shows the p5.js code editor interface. The code in the 'sketch.js' file is:

```
1 // Function setup()
2 // createCanvas(600, 600);
3 //
4 //
5 // Function draw()
6 // background(220);
7 // rect(40, 100, 50, 50);
8 //
```

The preview window shows a gray canvas with a single white square centered in the middle.

Je kunt x en y van een vorm vervangen door width of height – dit betekent de breedte en hoogte van het canvas. Een handig trucje om je rect in het midden van de pagina te krijgen, is 'width/2' (breedte van het canvas gedeeld door twee) of 'height/2', min de helft van de breedte van de rect. Je vervangt x door width en y door height.  
Deze code geeft hetzelfde resultaat als 'width/2-25' en 'height/2-25' (de uitkomst van die rekensommen is in dit geval 275) maar dat is meer rekenwerk. Je mag zelf kiezen wat je handiger vindt.



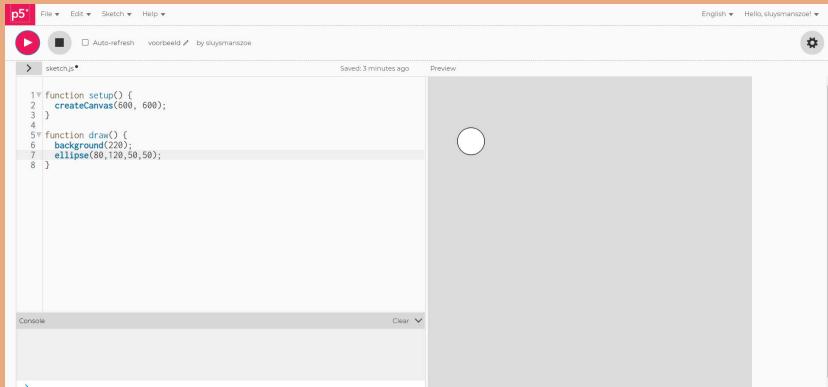
The screenshot shows the p5.js code editor interface. The code in the 'sketch.js' file is:

```
function setup() {
  createCanvas(600, 600);
}

function draw() {
  background(220);
  rect(width/2, height/2, 50, 50);
}
```

The preview window shows a gray canvas with a single white square centered in the middle.

Als de x en y van je ellipse gelijk zijn, heb je een cirkel.

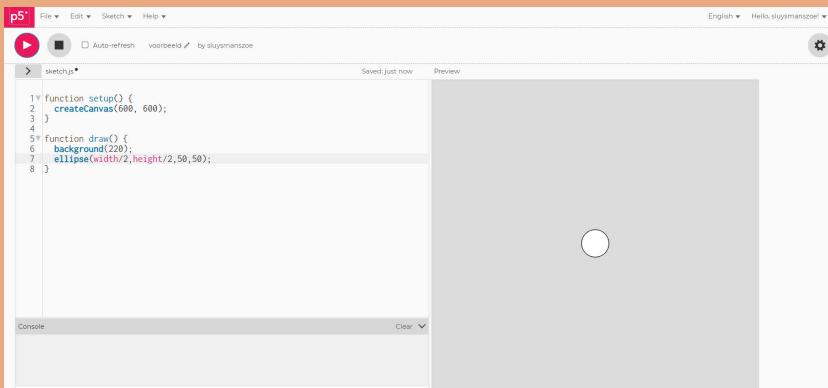


The screenshot shows the p5.js code editor interface. The code in the sketch.js file is:

```
1 function setup() {
2   createCanvas(600, 600);
3 }
4
5 function draw() {
6   background(220);
7   ellipse(300, 300, 50, 50);
8 }
```

The preview window shows a single white circle centered at the coordinates (300, 300) on a light gray background. The console window is empty.

Bij een ellipse, net als bij een rect, kun je x en y vervangen door width en height. Je hoeft hierbij niet de helft van de hoogte en breedte van width en height af te trekken.



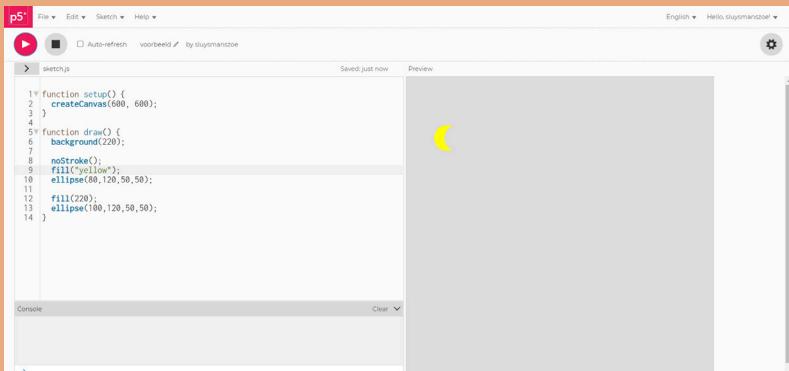
The screenshot shows the p5.js code editor interface. The code in the sketch.js file is:

```
1 function setup() {
2   createCanvas(600, 600);
3 }
4
5 function draw() {
6   background(220);
7   ellipse(width/2, height/2, 50, 50);
8 }
```

The preview window shows a single white circle centered at the coordinates (width/2, height/2) on a light gray background. The console window is empty.



Je kunt ellipses gebruiken om ingewikkeldere vormen te maken. Hier is bijvoorbeeld een maan gemaakt door een cirkel in de kleur van de achtergrond deels over een gele cirkel heen te plaatsen.

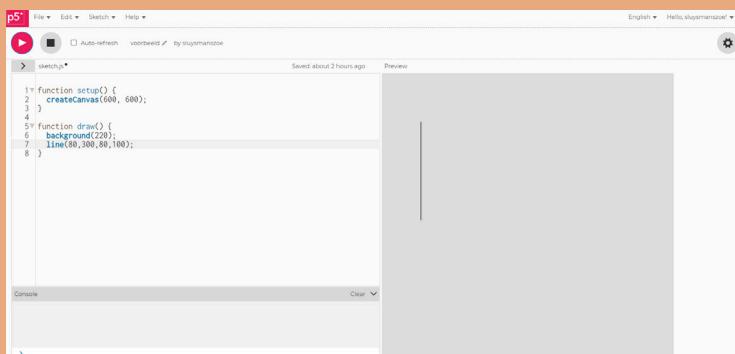


A screenshot of the p5.js code editor. The code in the sketch.js file is:

```
1 // function setup() {
2 //   createCanvas(600, 600);
3 // }
4
5 // function draw() {
6 //   background(220);
7 //
8 //   noStroke();
9 //   fill("yellow");
10 //   ellipse(0,120,50,50);
11 //
12 //   fill(220);
13 //   ellipse(100,120,50,50);
14 }
```

The preview window shows a yellow crescent moon shape on a grey background.

Als x bij allebei de punten gelijk is, is je line verticaal recht.



A screenshot of the p5.js code editor. The code in the sketch.js file is:

```
1 // function setup() {
2 //   createCanvas(600, 600);
3 // }
4
5 // function draw() {
6 //   background(220);
7 //   line(80,300,80,100);
8 // }
```

The preview window shows a single vertical black line at x=80 from y=300 to y=100.

A blue-themed digital workspace featuring a hot air balloon with a globe basket, a white cloud, and a p5.js code editor window.

The hot air balloon has a green and blue globe basket containing small flags. A white cloud is on the left, and another white cloud is at the bottom right.

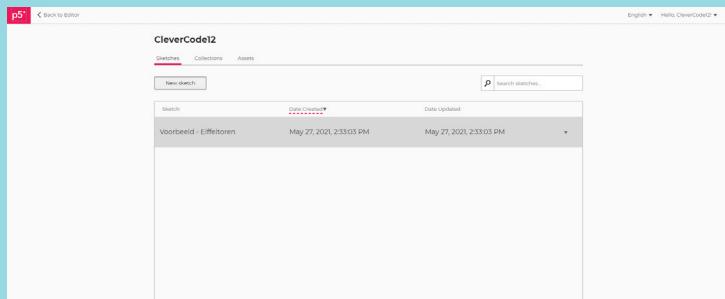
The p5.js code editor window shows the following code:

```
p5.js
1 // function setup() {
2 //   createCanvas(600, 600);
3 // }
4
5 // function draw() {
6 //   background(220);
7 //   line(0, 0, 200, 100);
8 // }
```

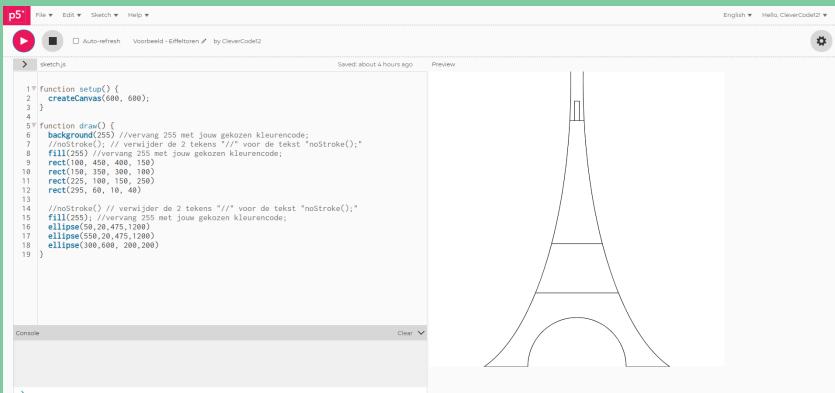
En dan zijn we eindelijk aangekomen bij de opdrachten van onze workshop! We reizen naar Londen, Rio de Janeiro, New York, Hongkong en Parijs. In elke stad kun je kiezen tussen een tekening van makkelijke, gemiddelde of moeilijke moeilijkheidsgraad. Die tekening ga je namaken! (Maar als je er een creatieve draai aan wilt geven, is dat natuurlijk helemaal geen probleem.)

## opdrachten uitleg

En dan zijn we eindelijk aangekomen bij de opdrachten van onze workshop! We reizen naar Londen, Rio de Janeiro, New York, Hongkong en Parijs. In elke stad kun je kiezen tussen een tekening van makkelijke, gemiddelde of moeilijke moeilijkheidsgraad. Die tekening ga je namaken! (Maar als je er een creatieve draai aan wilt geven, is dat natuurlijk helemaal geen probleem.)



De startercode zelf ziet er dan bijvoorbeeld zo uit:



The screenshot shows the p5.js code editor with a sketch titled "Voorbeeld - Effeltoren". The code is as follows:

```
function setup() {
  createCanvas(600, 600);
}

function draw() {
  background(255) //vervang 255 met jouw gekozen kleurencode;
  //fill(255); // verwijder de 2 tekens "//" voor de tekst "noStroke();"
  fill(255) //vervang 255 met jouw gekozen kleurencode;
  rect(100, 450, 400, 150)
  rect(100, 450, 400, 150)
  rect(225, 100, 150, 250)
  rect(295, 60, 10, 40)
  noStroke() // verwijder de 2 tekens "//" voor de tekst "noStroke();"
  fill(255); //vervang 255 met jouw gekozen kleurencode;
  ellipse(558, 20, 475, 1200)
  ellipse(558, 20, 475, 1200)
  ellipse(300, 600, 200, 200)
}
```

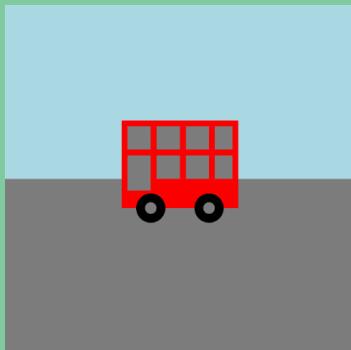
The preview window shows a wireframe drawing of the Eiffel Tower.

Hieronder vind je de opdrachten dus. Kies bij elke stad uit de makkelijke, gemiddelde of moeilijke optie. Probeer eerst zelf de tekening na te maken, en als het niet lukt, kun je altijd onze startercode nog gebruiken. Veel succes!

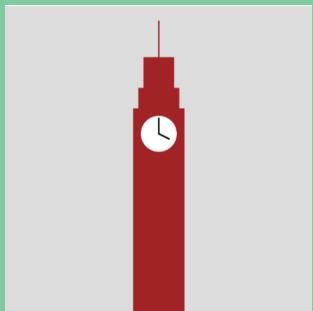
# opdrachten

Londen

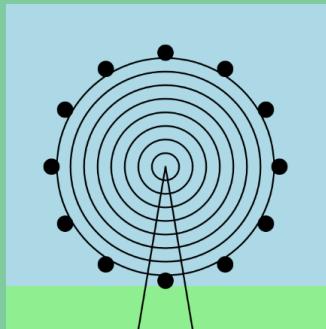
Gemakkelijk:  
een Londense  
dubbeldekker.



Gemiddeld:  
de Big Ben



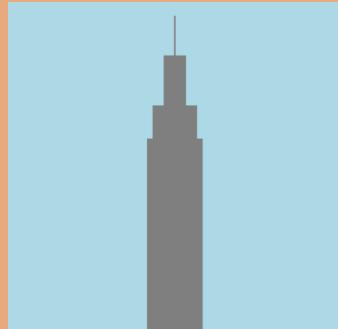
Moeilijk:  
de London Eye.



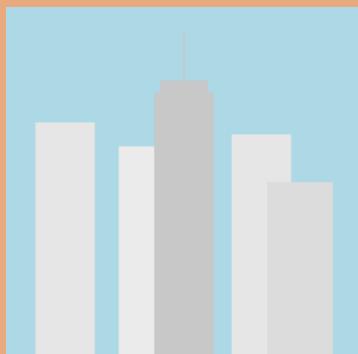
# opdrachten

New York

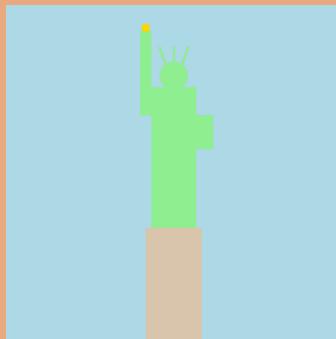
Gemakkelijk:  
Empire State Buil-  
ding.



Gemiddeld:  
landschap met wol-  
kenkrabbers.



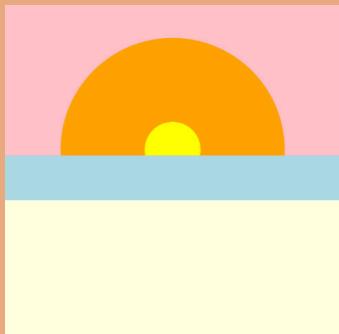
Moeilijk:  
het Vrijheidsbeeld.



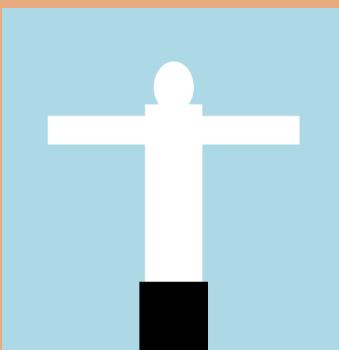
# opdrachten

Rio de Janeiro

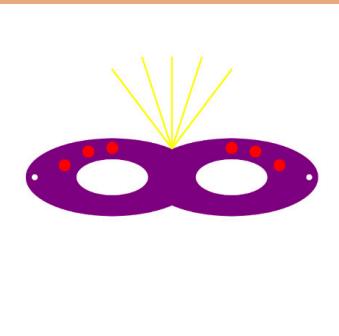
Gemakkelijk:  
strand.



Gemiddeld:  
Christus de Verlosser.



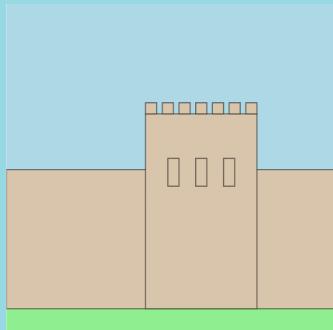
Moeilijk:  
Carnavalsmasker.



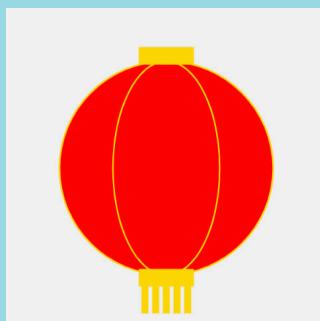
# opdrachten

## Hongkong

Gemakkelijk:  
De Chinese Muur



Gemiddeld:  
een lampion.



Moeilijk:  
een panda.



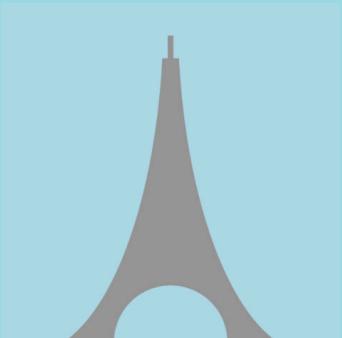
# opdrachten

Parijs

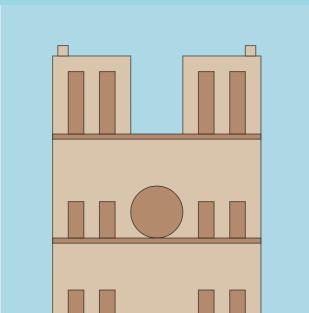
Gemakkelijk:  
De Arc de Triomphe.



Gemiddeld:  
de Eiffeltoren.



Moeilijk:  
de Notre Dame.



# **bedankt!**

Dankjewel dat je met ons op wereldreis bent gegaan! We vonden het erg leuk om met je op pad te gaan, en we hopen dat je iets nieuws hebt geleerd!

Groetjes,  
Gemma, Max, Milène en Zoë

## **waar vind je wat?**

### **Het team**

- Onze namen – pagina 1
- De steden die we gaan behandelen – pagina 1

### **De basis van coderen**

- Wat is JavaScript? – pagina 1
- Wat is p5.js? – pagina 1
- Hoe werkt volgorde in p5.js? – pagina 2

### **Een eerste p5.js-bestand**

- De website waar je op gaat werken – pagina 2
- Hoe log je in in p5.js? – pagina 2-3
- Wat zijn de inloggegevens? – pagina 3
- setup en draw (waar plaats je je code?) – pagina 4
- createCanvas (de grootte van je canvas instellen) – pagina 4
- Wat zijn pixels? – pagina 4
- Hoe voer je je code uit? – pagina 4



## **Coördinaten**

- rect (rechthoek) – pagina 4-5
- x en y (hoe plaats je je vorm op een bepaalde plek?) – pagina 5

## **Vormen**

- ellipse (cirkel) – pagina 5-6
- line (lijn) – pagina 6-7

## **Kleuren**

- fill (hoe maak je een vorm een bepaalde kleur?) – pagina 7
- stroke (hoe maak je een rand of lijn een bepaalde kleur?) – pagina 7
- Woorden gebruiken om iets in te kleuren – pagina 7-8
- Grijstinten gebruiken om iets in te kleuren – pagina 8-9
- rgb gebruiken om iets in te kleuren – pagina 9-10-11

## **Final touches**

- noStroke (hoe haal je de rand van een vorm weg?) – pagina 12
- strokeWeight (hoe maak je een lijn of rand dikker?) – pagina 13
- noFill (hoe haal je de vulling van een vorm weg?) – pagina 13-14



## **Code opslaan**

- Wat voor naam moet je je bestand geven?  
– pagina 14
- Hoe sla je je code op? – pagina 14-15-16
- Hoe download je je code? – pagina 15-16
- Hoe maak je een nieuw bestand aan nadat je je code hebt opgeslagen? – pagina 16

## **Meer vormen**

- Hoe maak je een vierkant? – pagina 17
- width en height – pagina 17-18
- Hoe maak je een cirkel? – pagina 18
- Hoe maak je rechte lijnen? – pagina 20

## **Opdrachten**

- Wat is startercode? – pagina 20
- Waar kun je de startercode vinden?  
– pagina 20-21
- Londen – pagina 21-22
- New York – pagina 22-23
- Rio de Janeiro – pagina 23-24
- Hongkong – pagina 24-25
- Parijs – pagina 25-26

## **Wat staat waar?**

- We hopen dat je onze workshop leuk vond :)  
– pagina 26

