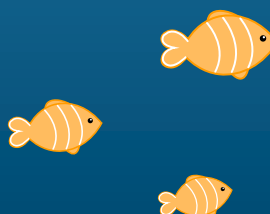
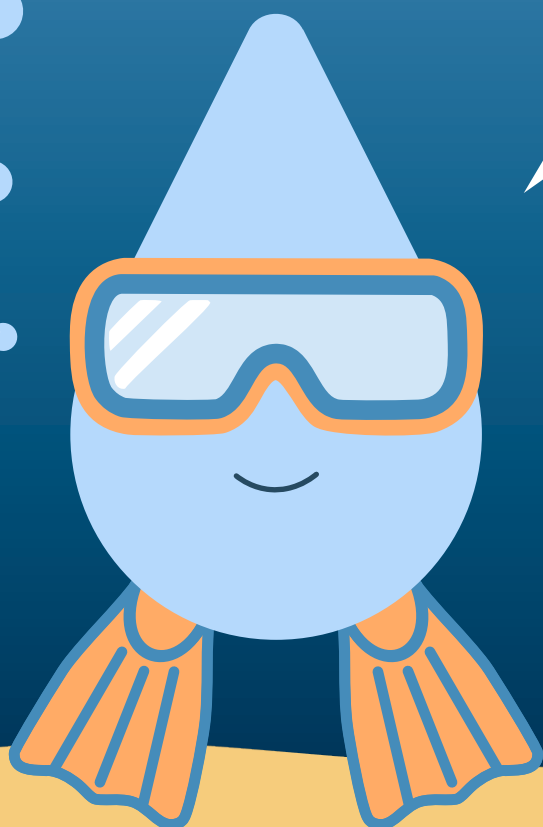


KUNST MET CODE

Werkboekje

DIGIDIVE

In deze workshop ontwikkel je leuke zeedieren in het programma p5.js. Kinderen zullen aan het einde van de les in staat zijn om hun eigen kunstwerk te maken!

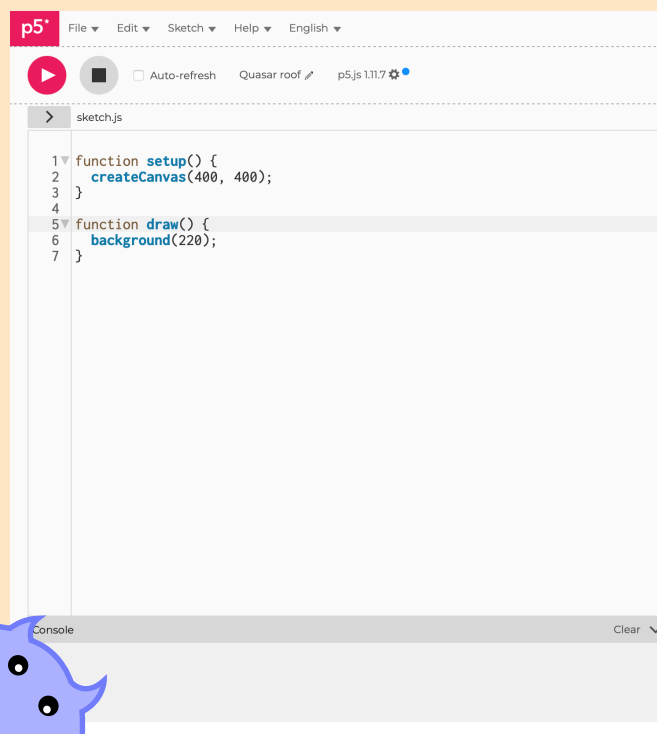




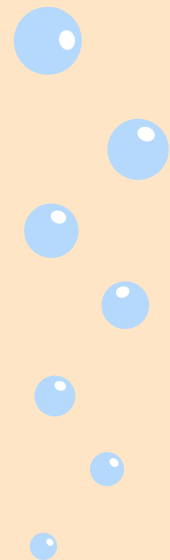
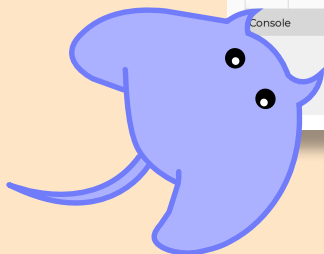
LEERKAART 1: INTRODUCTIE P5.JS!

Je leert stapsgewijs de code om je figuur te programmeren, aan te kleden en in te kleuren. Volg de stappen in de leerkaarten op. Als je hulp nodig hebt, vraag je Boris, Zoë of Lina om hulp!

Iedere keer als je code hebt toegevoegd en wil kijken wat het resultaat is klik je op de play knop. Zitten er geen fouten in je code? Dan krijg je je resultaten rechts te zien!



```
1 function setup() {
2   createCanvas(400, 400);
3 }
4
5 function draw() {
6   background(220);
7 }
```



LEERKAART 2: SET-UP & DRAW!



In p5.js begint alles met twee super belangrijke stukjes code: **setup()** en **draw()**. Ze zijn als het hart en de longen van je tekening, zonder die twee werkt er niks!

setup() - de voorbereiding: Dit stukje code wordt één keer uitgevoerd, helemaal aan het begin. Alles wat je hier neerzet, wordt klaargezet voor je tekening.

Bijvoorbeeld:

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

Met **createCanvas(400, 400);** zeg je: "maak een tekenblad van 400 bij 400 pixels groot."

draw() - de herhaling: Alles wat je in **draw()** zet, wordt steeds opnieuw gedaan, ongeveer 60 keer per seconde! Zo kun je dingen bewegen, veranderen of laten reageren.

Bijvoorbeeld:

```
function draw() {  
  background(220);  
}
```

Hiermee maak je het tekenblad steeds lichtgrijs, zodat je beeld elke keer opnieuw wordt getekend.

Belangrijk: Je hebt altijd een **setup()** en een **draw()** nodig in je sketch. Zonder die twee werkt je programma niet goed! Ook moet je je code na de haakjes **()** altijd afsluiten met een **;**. Anders werkt je code niet!



LEERKAART 3: PIXELS!

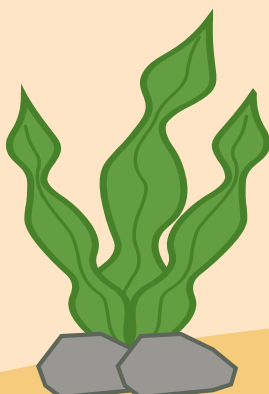
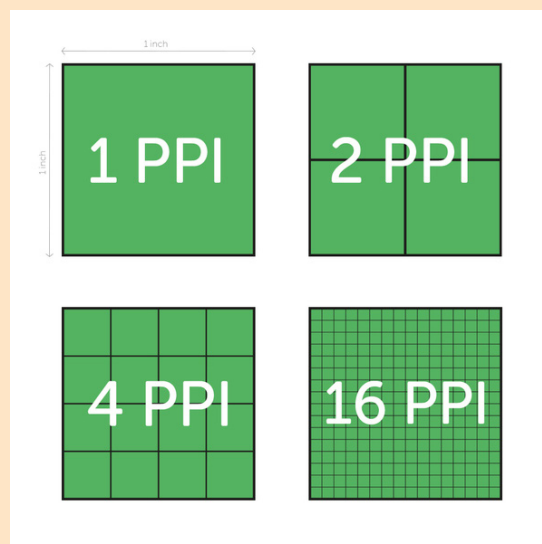
Alle schermen, je telefoon, je laptop, je tablet of zelfs een tv, bestaan uit hele kleine gekleurde vierkantjes. Die vierkantjes heten pixels. Je ziet ze meestal niet, omdat ze zó klein zijn en er zó veel van zijn dat ze samen gewoon een beeld vormen. Maar als je héél ver zou inzoomen op je scherm, dan zie je ineens duizenden of zelfs miljoenen kleine vakjes.

Bij coderen maken we ook gebruik van pixels. Als je iets maakt in p5.js, bijvoorbeeld een cirkel of een rechthoek, dan vertel je precies op welke pixel het vormpje moet beginnen.

Bijvoorbeeld:

```
createCanvas (400 , 400) ;
```

Dit betekent dat het canvas waarop je jouw ontwerp maakt 400 pixels bij pixels groot is.

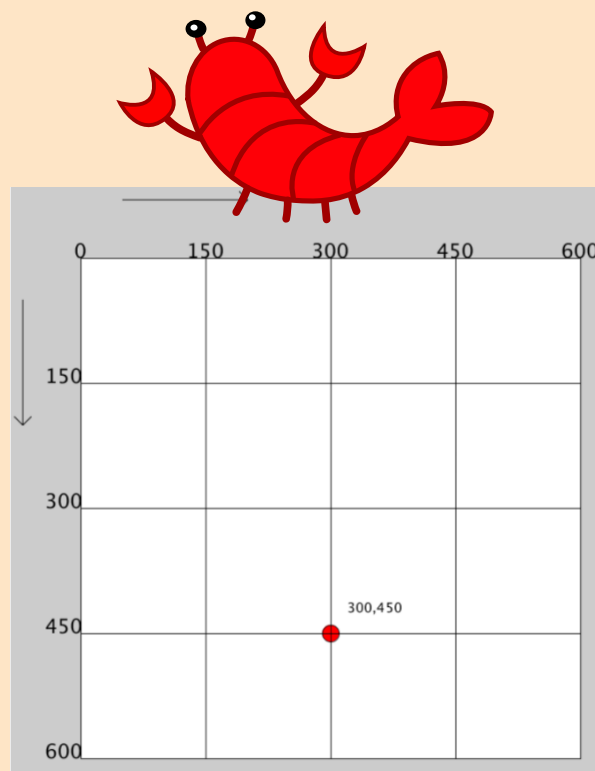


LEERKAART 4: COÖRDINEREN!



We zijn aan het programmeren in de programmeertaal JavaScript. In JavaScript kun je heel makkelijk vormen tekenen. Je kunt het venster in gedachten indelen in heel veel vakjes (pixels) waarbij je vormen kunt plaatsen op een specifieke plek door aan te geven hoeveel vakjes naar rechts en hoeveel vakjes naar beneden de vorm getekend moet worden. Dat gebeurt altijd vanaf de linkerbovenhoek. De coördinaten van de linkerbovenhoek zijn 0, 0.

In dit voorbeeld is de rode stip 300 pixels naar rechts en 450 pixels naar beneden geplaatst vanuit de linkerbovenhoek.





LEERKAART 5: VORMEN!

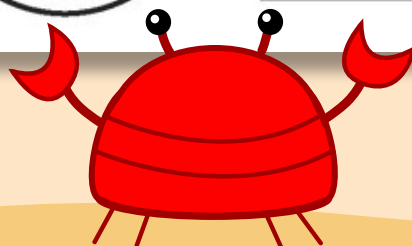
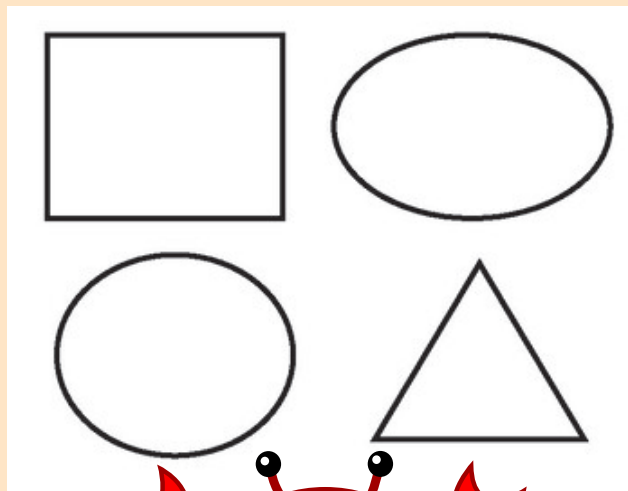
In p5.js kun je met een paar simpele regels code allerlei vormen tekenen op het scherm! Denk aan rechthoeken, cirkels en lijnen. Deze vormen helpen je om dingen te bouwen, zoals gezichten, dieren, gebouwen of wat je maar kunt verzinnen. Hier zijn een paar vormen die je kunt gebruiken:

```
rect(50, 50, 100, 50);    // rechthoek  
ellipse(200, 100, 80, 80); // cirkel  
line(300, 50, 350, 100); // lijn
```

`rect(50, 50, 100, 50);` betekent: begin op plek (50, 50), maak een rechthoek van 100 breed en 50 hoog.

`ellipse(200, 100, 80, 80);` betekent: teken een cirkel in het midden op (200, 100) met een breedte en hoogte van 80.

Speel zelf eens met de getallen om te zien wat er gebeurt! Wil je een grotere vorm? Maak de getallen groter. Wil je hem naar links of rechts verplaatsen? Verander de eerste twee cijfers.



LEERKAART 6: KLEUR!










Rechthoeken, ellipsen en andere vormen kun je vullen met een kleur. Als je een bepaalde vorm een kleur wil geven moet je bóven die vorm een nieuwe regel toevoegen en de volgende code typen:

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

Let op! Tussen de haakjes staan alleen maar cijfers. Je zet tussen de haakjes de RGB code van een kleur. In dit geval het cijfer 255, het cijfer 0 en het cijfer 0.

Gebruik deze kleurentabel om zelf een mooie kleur toe te voegen aan je vormen (als je een andere kleur wil, kan je ook naar de site [rapidtables.com > RGB color codes](https://rapidtables.com/RGB-color-codes/) gaan):

fill(255,0,0); = ROOD	
fill(0,255,0); = GROEN	
fill(0,0,255); = BLAUW	
fill(255,255,0); = GEEL	
fill(247,160,17); = GEMENGD	
fill(0); = ZWART	
fill(255); = WIT	

Als je een vorm juist géén kleur wil geven kun je die vorm bijvoorbeeld vullen met wit of zwart.

Voor wit geldt de code: **fill (255) ;**

Voor zwart geldt de code: **fill (0) ;**



LEERKAART 7: STROKE!








Elke vorm die je codeert heeft een rand eromheen.

Dit doe je met `stroke(0)` ;

Tussen de haakjes komt ook de RGB code van de kleur die je wilt dat de lijn heeft.
Als je de dikte van de lijn wilt aanpassen gebruik je `strokeWeight()` ;

Hoe groter het getal hoe dikker de lijn.

Soms wil je een vorm zonder omlijning. Dat doe je met `noStroke()` ;

<code>stroke(255,0,0);</code> = ROOD	
<code>stroke(0,255,0);</code> = GROEN	
<code>stroke(0,0,255);</code> = BLAUW	
<code>stroke(255,255,0);</code> = GEEL	
<code>stroke(71,185,85);</code> = GEMENGD	
<code>stroke(0);</code> = ZWART	
<code>stroke(255);</code> = WIT	

(Als je een andere kleur wil, kan je ook naar de site [rapidtables.com > RGB color codes](https://rapidtables.com/rgb-color-codes/) gaan)

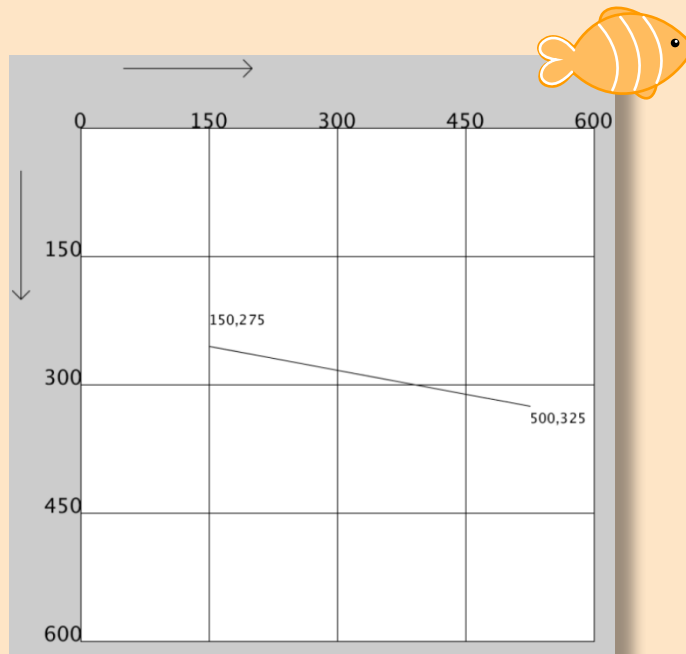
LEERKAART 8: LIJNEN!



We kunnen niet alleen rechthoeken en ellipsen tekenen, maar ook lijnen. Een lijn heeft altijd een beginpunt en een eindpunt. Lijnen hebben geen vulling, het zijn alleen streepjes. En die streepjes hebben een rand.

Bijvoorbeeld:

```
line(150, 275, 500, 325);
```





EXTRA'S!

Bewegen met je muis

Wist je dat je computer precies weet waar je muis is op het scherm? En wist je dat je die plek kunt gebruiken in je code? Dat doe je met **mouseX** en **mouseY**!

mouseX betekent: hoe ver is je muis van de linkerkant van het scherm?

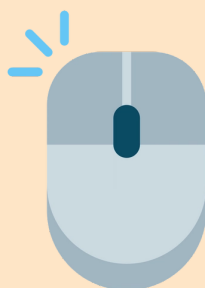
mouseY betekent: hoe ver is je muis van de bovenkant van het scherm?

Deze functies zijn dus ook net als coördinaten. Je zegt: "ik wil weten waar mijn muis is", en de computer zegt bijvoorbeeld: **mouseX** = 150 en **mouseY** = 90.

Je kunt deze waarden gebruiken om vormen mee te laten bewegen met je muis!

```
ellipse(mouseX, mouseY, 50, 50);
```

Nu tekent de computer een cirkel precies waar je muis is. Beweeg met je muis en kijk wat er gebeurt.



EXTRA'S!



Toetsen indrukken

Je kunt je vormen niet alleen met de muis, maar ook met je toetsenbord besturen! Daar gebruiken we `keyPressed()` voor.

`keyPressed()` betekent letterlijk: “er is op een toets gedrukt”. De computer let dus goed op: zodra jij een toets indrukt, weet hij het!

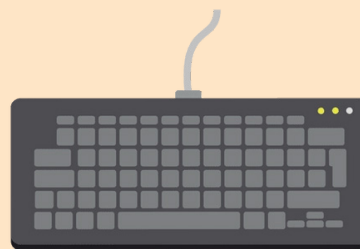
In dit voorbeeld laten we een vis omhoog en omlaag zwemmen met de pijltjestoetsen. Probeer het maar eens!

```
let y = 200;

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

function draw() {
  background(220);
}

function keyPressed() {
  if (keyCode === UP_ARROW) {
    y = y - 10; // vis gaat omhoog
  }
  if (keyCode === DOWN_ARROW) {
    y = y + 10; // vis gaat omlaag
  }
}
```



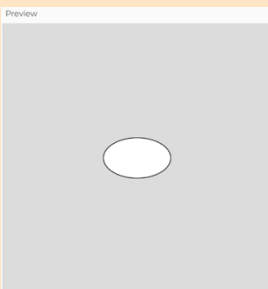
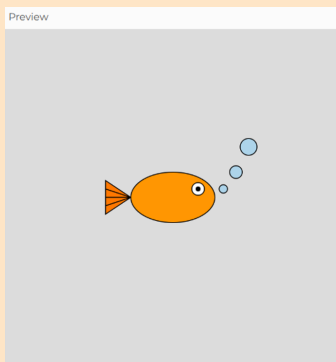


DE OPDRACHT!

De opdracht!

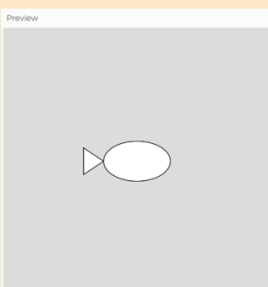
Nu gaan we zelf aan de slag! Je gaat je eigen vis maken. Als je hulp nodig hebt kan je ons om hulp vragen!

Voorbeeld:



Ellipse

Stap 1: Maak een ovaal in het midden van het canvas.

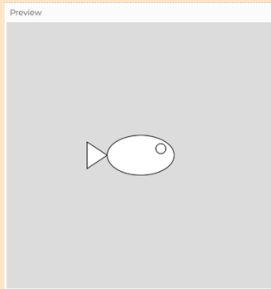


Triangle

Stap 2: Maak een triangle voor de staart. Deze geven we jullie anders is het te lastig.

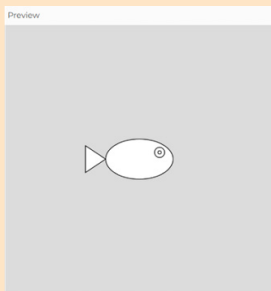
```
triangle(150, 200, 120, 180, 120, 220);
```

DE OPDRACHT!



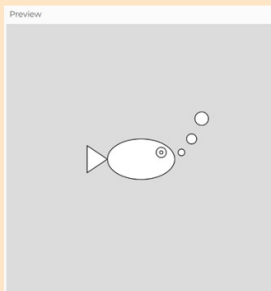
Oogwit

Stap 3: Maak een ellipse in het lichaam voor het oogwit.



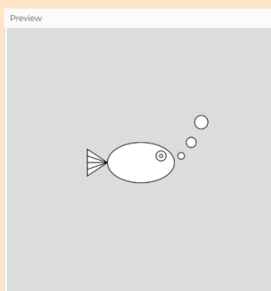
Pupil

Stap 4: Maak een rondje in het oogwit voor het pupil.



Bubbels

Stap 5: Teken 3 ellipsen boven de vis voor wat bubbels. Maak elke bubbel een andere grootte.



Details

Stap 6: Teken wat lijnen in de staart.

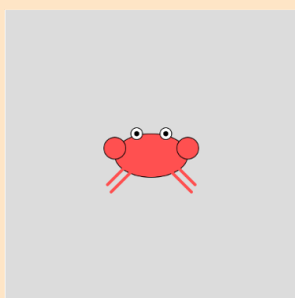


VOOR DE HARDLOPERS!

Als jullie eerder klaar zijn, kunnen jullie nu zelf kiezen welk zeedier jullie gaan na-maken. Dit doen jullie zelfstandig. Als jullie er niet uitkomen kunnen jullie het aan je buur vragen en anders aan ons!



Makkelijk: octopus



Medium: krab



Moeilijk: rog

TEKENRUIMTE!



Gemaakt door Lina, Zoë en Boris

