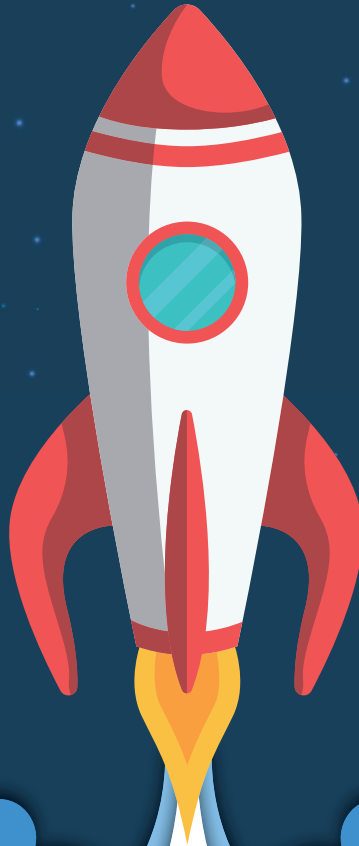


# SPACE CODE

## WERKBOEK



**Naam:** \_\_\_\_\_

**Groep:** \_\_\_\_\_

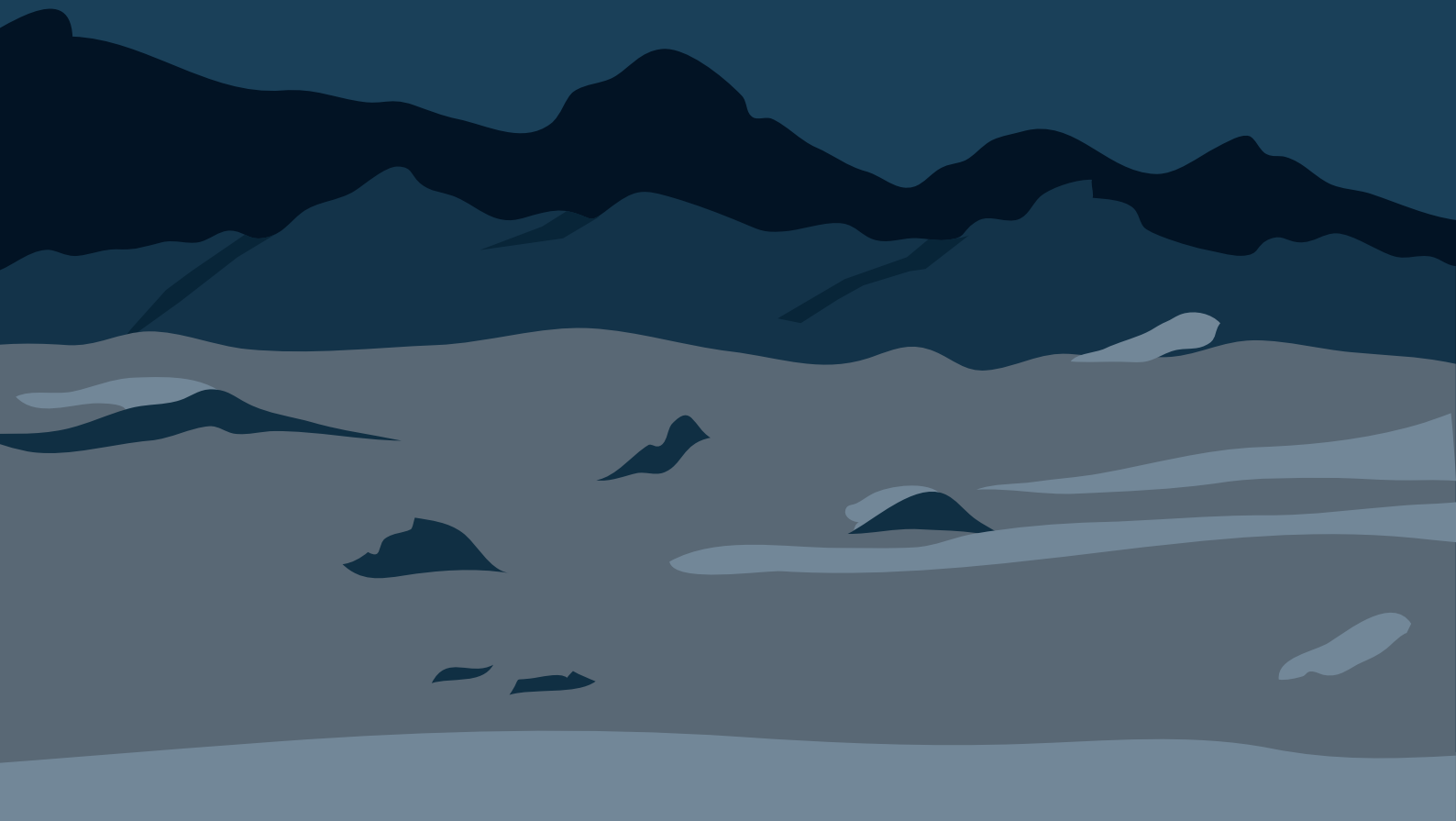


# WELKOM AAN BOORD!

Na 5 lange maanden op Mars ben je eindelijk klaar met onderzoek doen. Nu is het tijd om terug naar huis te gaan.

Maar als je in de raket stapt en wil vertrekken kom je erachter dat hij kapot is!

Aan jou is nu de taak om de raket te repareren, zodat je veilig terug naar huis kunt vliegen.



# WAT IS CODEREN?

Coderen is het stap voor stap uitleggen aan je computer wat hij moet doen. Je vertaalt je opdracht naar 'computertaal'.

Computertaal is geen taal zoals Nederlands of Engels. Talen die de computer wel kent zijn bijvoorbeeld Javascript, python en Processing. Dit zijn maar 3 voorbeelden, maar er zijn er nog veel meer! Misschien ken jij er toevallig een?

**Wij gaan vandaag aan de slag met P5, deze is eenvoudiger te leren en is speciaal gemaakt voor beginners!**

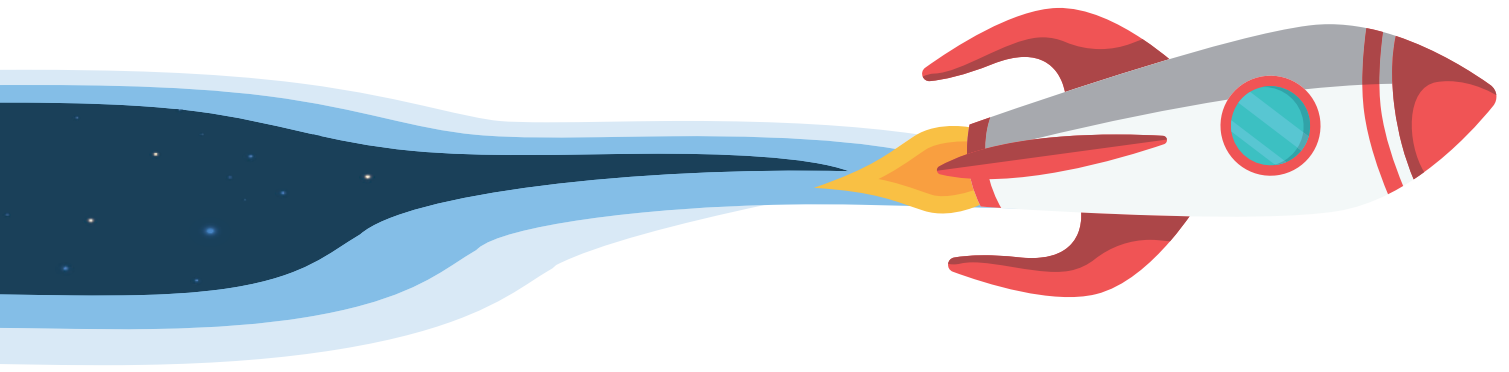
Als je code wil schrijven, heb je een **editor** nodig. Die kun je eenvoudig op internet vinden. Om te beginnen ga je naar de volgende website:

<https://editor.p5js.org>



## WIST JE DAT?

Er maar liefst 700 verschillende codeertalen bestaan? dat zijn er heeeel veel.



## LET'S GO!

De website die je nu voor je ziet, is een **editor**. Hierin kun je code schrijven. Zoals je ziet, staat er al 2 stukjes code, de setup en draw. Deze heb je **altijd** nodig als je met p5 werkt.

### Setup

De setup is de basis van je code. Alle code die in de setup staat voert de computer 1 keer uit. In de setup bepaal je ook de grootte van je tekenveld. Dit noemen we het **canvas**.

Zie je in de editor het volgende stukje code staan?

```
function setup(){}
```

Kijk nog eens naar de editor. Zie je op de regel eronder deze regel code staan?

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

Klik eens op de **play knop** links bovenin. Zie je dat er een grijs vak verschijnt aan de rechterkant van je scherm? Dit is je canvas!

Verander `createCanvas(400,400);` nu eens in `createCanvas(800,800);` en klik vervolgens weer op de play knop. Zie je het verschil?

**Denk je aan de puntkomma op het einde?**

## LET OP!

Zie je dat createCanvas uit 2 woorden bestaat?  
Als je een code schrijft die uit meerdere woorden bestaat, moet je de woorden aan elkaar schrijven.  
Om ervoor te zorgen dat de code leesbaar blijft, begint het eerste woord met een kleine letter, en elk woord dat daarna komt, begint met een hoofdletter!

Om even te oefenen hiermee, staan hieronder twee kleine opdrachten, aan jou de taak om het vakje aan te kruisen van de zin die goed is geschreven in codetaal!

Welke zin is goed geschreven?

- ☐ ditisGoedgeschreven
- ☐ DitIsGoedGeschreven
- ☐ ditIsGoedGeschreven
- ☐ Ditisgoedgeschreven

Welke zin is goed geschreven?

- ☐ DitIsCode
- ☐ ditiscode
- ☐ Ditiscode
- ☐ ditIsCode

## Draw

Weet je nog dat de setup maar 1 keer wordt uitgevoerd door de computer? Om te tekenen hebben we de **draw** nodig, deze ververs maar liefst 60 keer per seconde!

Een draw schrijf je zo:

```
function draw(){}  

```

tussen de {} van de draw schrijven we de rest van de code, bijvoorbeeld ook de kleur van ons **canvas**. De regel code die daarvoor zorgt, staat al in de draw. Zie je hem staan? Schrijf de code op de regel hieronder!

---

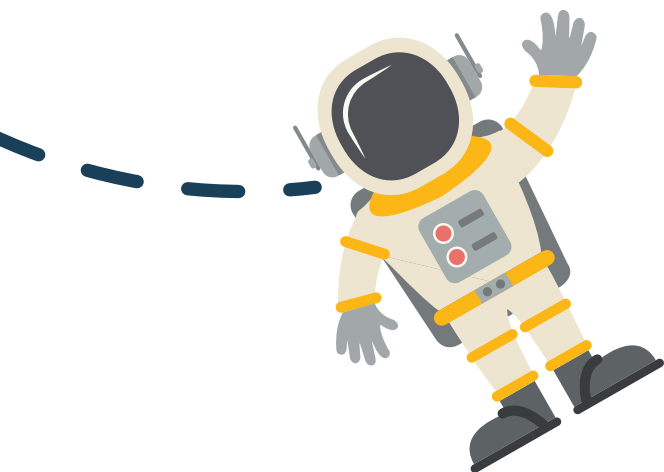
### (achtergrond) kleur

Op dit moment heeft je achtergrond een grijze kleur. Dat komt door het getal dat tussen de () staat van de regel code die je net hebt opgeschreven. Pas dat cijfer maar eens aan met een getal tussen de 0 en 255 en druk daarna op **play linksboven**. Probeer dit maar een paar keer uit! En beantwoord daarna de volgende vragen.

### Omcirkel het juiste antwoord:

Hoe dichterbij de 0, hoe **donkerder** / **lichter** de kleur.

Hoe dichterbij de 255, hoe **donkerder** / **lichter** de kleur.



# KLEUREN!

Toppie, je weet nu hoe je wit, zwart en grijs kunt maken! Nu is het tijd om met kleuren aan de slag te gaan.

In tegenstelling tot grijstinten, heb je voor kleuren niet 1, maar 3 cijfers nodig! Deze staan voor **RGB**, oftewel **Red**, **Green**, **Blue**.

**Red**             $= (255, 0, 0)$

**Green**         $= (0, 255, 0)$

**Blue**           $= (0, 0, 255)$

Het maximale getal dat je per kleur kan gebruiken is 255, je zou dit kunnen vergelijken met 'scheppen'.

Om **rood** te maken heb je 255 scheppen rood nodig, 0 scheppen groen en 0 scheppen blauw. Andere kleuren maak je door deze drie cijfers aan te passen. Om **paars** te maken neem je bijvoorbeeld 140 scheppen rood, 0 scheppen groen en 255 scheppen blauw.

**Vul onderstaande code eens in in je editor. Schrijf daarna op welke kleur je krijgt.**

`background(255,255,0);`      = \_\_\_\_\_

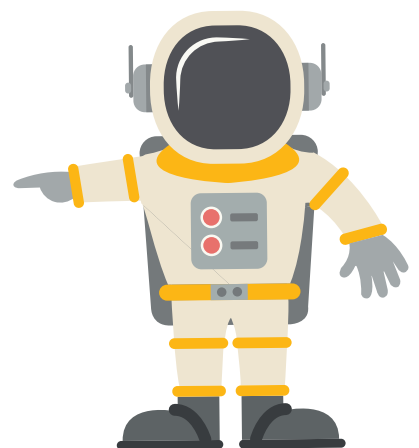
**Probeer nu eens de volgende code. Welke kleur krijg je nu?**

`background(255,56,152);`      = \_\_\_\_\_



## WIST JE DAT?

Je bijna 17 miljoen RGB combinaties kunt maken? Dat zijn er veel he?!





# RASTER EN PIXELS

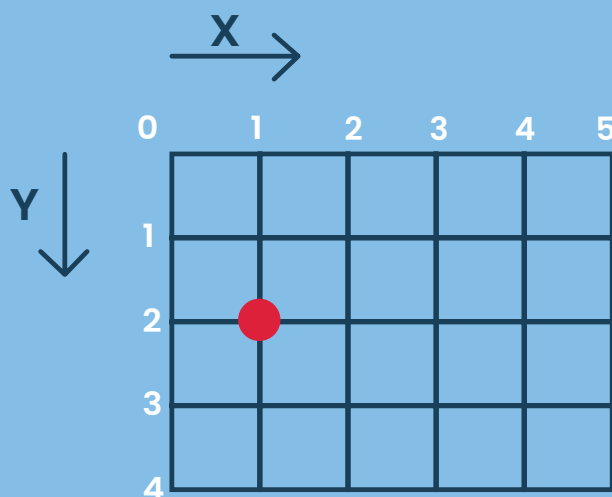
Met coderen werken we niet met centimeters maar met **pixels**.  
Al die pixels staan in een **raster** van heel veel vierkantjes waarbij elk vierkantje 1 pixel breed en 1 pixel hoog is.

Op je canvas op de computer, zit een onzichtbaar raster. In je Setup in de editor, zie achter de code `createCanvas()` 2 getallen staan. Die getallen bepalen hoeveel pixels breed en hoog je canvas is! Hieronder zie je een raster, de horizontale lijn noemen we de x-as, en de verticale lijn noemen we de y-as.

Zie je de stip staan op de coördinaten (1,2) ?

Teken nu zelf een stip op de coördinaten (4,3)

**Als je coördinaten wil bepalen, kijk je altijd eerst naar de x-as en daarna naar de y-as!**



Heb je de stip getekend op (4,3) ?

teken er dan nu een op (2,1), en onthoud:

**Eerst naar rechts, dan naar beneden!**

# VIERKANT

Zoals je misschien al gemerkt hebt, is alle code in het Engels. Dus als we een vierkant willen tekenen hebben we de afkorting nodig van het woord **rectangle**. Het stukje code dat je dan schrijft is:

```
rect();
```

Tussen de haakjes komen in totaal 4 coördinaten.

**De eerste 2 coördinaten bepalen waar je vierkant begint. En dit is bij een vierkant de linkerbovenhoek.**

**coördinaat 1:** dit is de x-as. Hoeveel stappen naar rechts?

**coördinaat 2:** dit is de y-as. Hoeveel stappen naar beneden?

**De laatste 2 coördinaten bepalen hoe groot je vierkant is.**

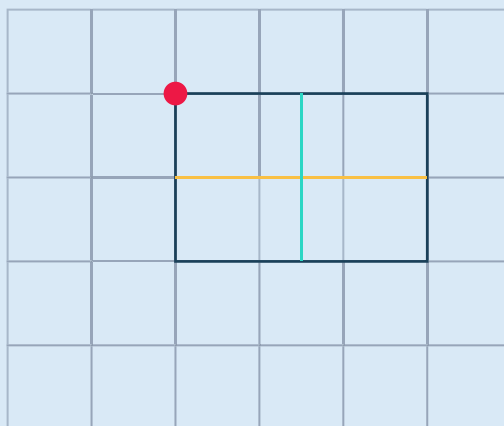
**coördinaat 3:** Hiermee bepaal je de breedte van je vierkant.

**coördinaat 4:** Hiermee bepaal je de hoogte van je vierkant.

Kijk maar eens naar het raster hier beneden, de kleuren van de coördinaten hierboven, zijn hetzelfde als in het raster.

De code die bij dit vierkant hoort is:

```
rect(2, 1, 3, 2);
```



# CIRKEL

Om een cirkel te maken gebruiken we ook een engels woord. Namelijk het woord **ellipse**, dat betekent eigenlijk ovaal. Het stukje code voor een cirkel is:

```
ellipse();
```

Tussen de haakjes komen in totaal 4 coördinaten.

**De eerste 2 coördinaten bepalen waar je cirkel begint. In tegenstelling tot een vierkant is dit niet linksboven, maar het middelpunt van de cirkel.**

**coördinaat 1:** dit is de x-as. Hoeveel stappen naar rechts?

**coördinaat 2:** dit is de y-as. Hoeveel stappen naar beneden?

**De laatste 2 coördinaten bepalen hoe groot je cirkel is.**

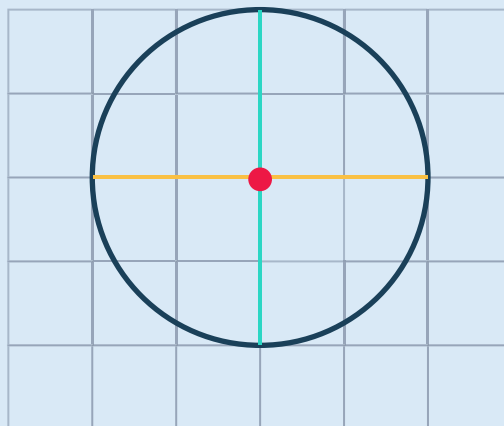
**coördinaat 3:** Hiermee bepaal je de breedte van je cirkel.

**coördinaat 4:** Hiermee bepaal je de hoogte van je cirkel.

Kijk maar eens naar het raster hier beneden, de kleuren van de coördinaten hierboven, zijn hetzelfde als in het raster.

De code die bij deze cirkel hoort is:

```
ellipse(3, 2, 2, 2);
```



# DRIEHOEK

Als derde vorm hebben we ook nog een driehoek. En ja, je raad het al, ook hier gebruiken we een engels woord voor!

Namelijk dit stukje code:

```
traingle();
```

**let op: Een driehoek heeft niet 4, maar 6 coördinaten!**

Voor elke punt van de driehoek zijn 2 coördinaten nodig.

**punt 1:**

**coördinaat 1:** dit is de x-as. Hoeveel stappen naar rechts?

**coördinaat 2:** dit is de y-as. Hoeveel stappen naar beneden?

**punt 2:**

**coördinaat 3:** Dit is de x-as van punt 2.

**coördinaat 4:** Dit is de y-as van punt 2.

**punt 3:**

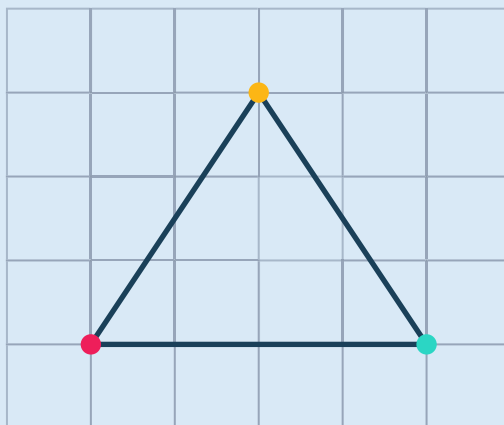
**coördinaat 5:** Dit is de x-as van punt 3.

**coördinaat 6:** Dit is de y-as van punt 3.

Kijk maar eens naar het raster hier beneden, de kleuren van de coördinaten hierboven, zijn hetzelfde als in het raster.

De code die bij deze driehoek hoort is:

```
traingle(1, 4, 3, 2, 5, 4);
```



# KLEURCODES

Als je een vorm kleur wil geven heb je het volgende stukje code nodig:

```
fill():
```

tussen de haakjes kun je 2 verschillende dingen invullen:

- Een getal tussen de 0 en 255, als je een wit, zwart of grijs wil.
- Een RGB waarde van 3 getallen tussen de 0 en 255 voor kleur.

De computer leest altijd van boven naar beneden. Dus je schrijft eerst welke kleur je wil, en daarna pas welke vorm je wil.

```
fill(255,0,0);  
rect(20,20,50,50);
```

Hier heb je een paar kleuren die je kan gebruiken. Maar je mag natuurlijk ook zelf experimenteren!



`fill(0, 159, 227);`



`fill(255, 0, 0);`



`fill(230, 0, 126);`



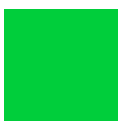
`fill(35, 55, 118);`



`fill(255, 237, 0);`



`fill(229, 109, 0);`



`fill(0, 206, 59);`



`fill(138, 42, 120);`

# OVERZICHT

**Hieronder staat een overzicht met termen en belangrijke dingen waar je altijd eventjes iets kan opzoeken!**

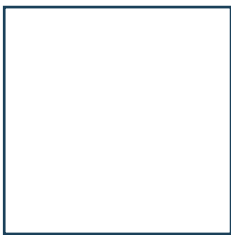
**editor:** Dit is het gedeelte op je scherm waar je de code typt.

**canvas:** Dit is het deel van je scherm waar je alle tekeningen ziet die je maakt.

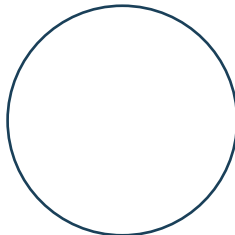
**raster:** Het onzichtbare veld van vierkantjes waar alles op getekend wordt.

**x-as:** De horizontale lijnen van je raster. Oftewel van links naar rechts.

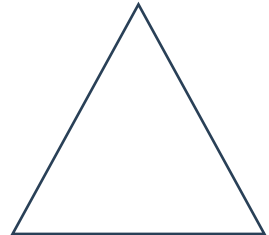
**y-as:** De verticale lijnen van je raster. Oftewel van boven naar beneden.



**rect():**



**ellipse();**



**triangle();**

- Onthoud bij coördinaten bepalen, je gaat **eerst naar rechts, dan naar beneden!**
- Als je een vorm een kleur wil geven, **typ je eerst fill()**, en op de regel daaronder de vorm!
- Als je de volgende vorm een **andere kleur** wil geven, typ je eerst weer de nieuwe fill(). De computer leest namelijk van boven naar beneden!

# JE BENT ER KLAAR VOOR!

fantastisch! Je kent nu de basis van coderen met p5. Nu kunnen we eindelijk de raket repareren.

De kaartjes met code liggen overal verspreid, als je die verzamelt, met het kwartetspel, kun je aan de slag!

Als je even niet weet hoe iets moet, mag je altijd vragen stellen of even teruglezen in het boekje!

**Veel succes!**

