

TODO: idee voor mezelf voor opdrachten voor volgende week, welke code tekent wat? zet een lijntje

Voor in de slides:

Nu gaan we uitvinden wat er gebeurt als we een hoek kiezen waardoor 360 niet mooi deelbaar is. Laat leerlingen eens zo'n getal verzinnen (met wisbordjes bij voorkeur).

Kies een leuk getal (bijv 100) en teken zelf het figuur op het bord.

In het begin zit het er een beetje gek uit:

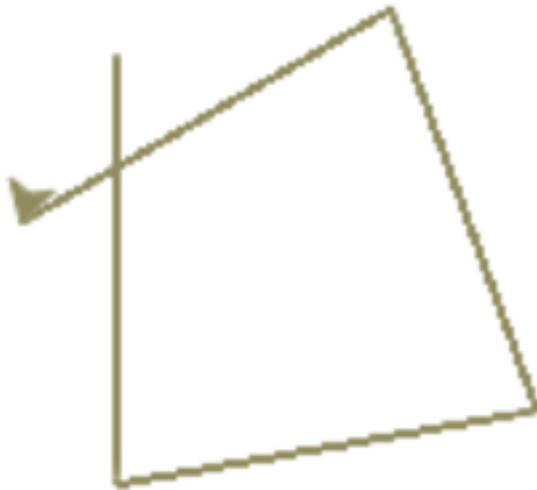


image-20181207082849999

Maar na naar een aantal stappen krijg je dit:



image-20181207082814644

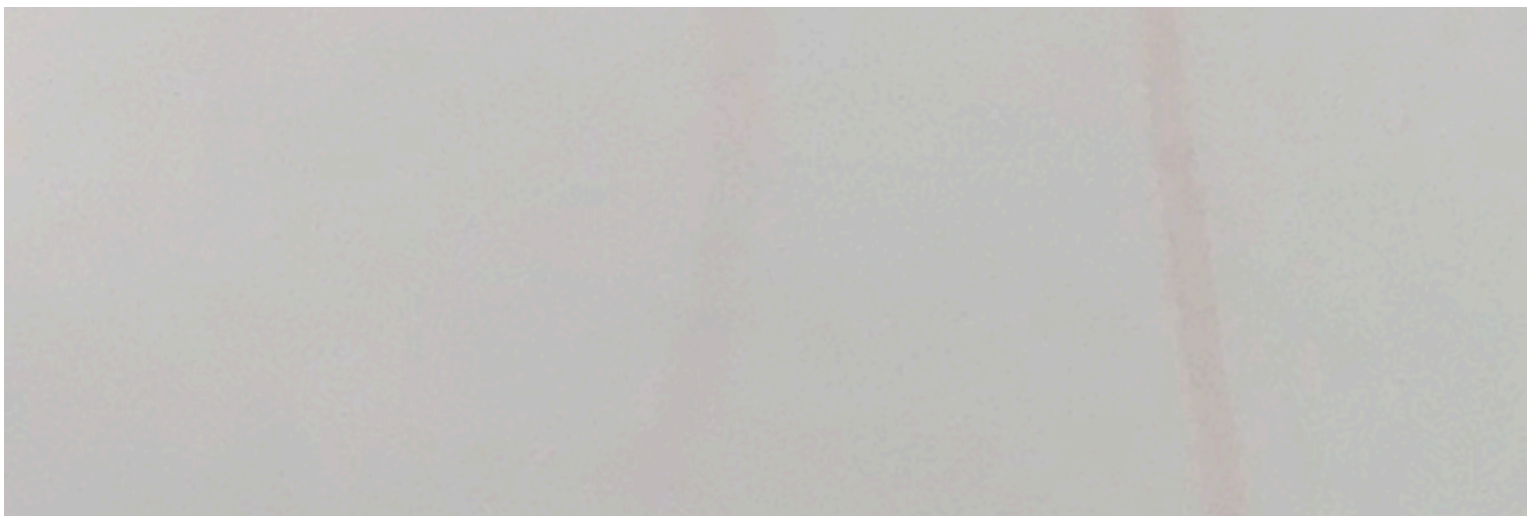
Dit is voor de slides:

Vierkant

Doe deze stappen zelf op het bord voor. Het is ideaal als je bij wiskunde even zo'n grote geo leent om het duidelijk voor te doen. Wat ik zelf ook doe:

- afstanden bij de lijnen
- in een andere kleur de draai aangeven, met de richting en het aantal graden

Dat ziet er dan zo uit:



10 cm

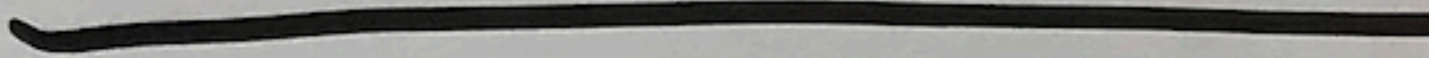




image-20181207090524855

Driehoek

Nu gaan IIn op papier een **driehoek** maken.

Leg uit/herhaal dat een figuur dat terug op zijn beginpunt komt, altijd hieken van samen 360 graden heeft.

"Kijk maar een vierkant is $90 + 90 + 90 + 90$, dan ben ik dus 'rond'. Als ik nu drie hoeken wil, welke som moet ik dan oplossen (dat is $360/3$). Wat is $360/3$? Nou 120"

Nu gaan we IIn vragen een (gelijkzijdige) driehoek te tekenen op dezelfde manier als we het vierkant deden. 90 graden lukte misschien nog wel zonder geo, maar nu zullen ze echt moeten gaan meten, anders wordt het niks bij de meesten.

Klassikale uitleg Python Turtle

Nu de leerlingen bekend zijn met de opdracht, en hun voorkennis van hoeken geactiveerd hebben, is het tijd om wat Python te gaat oefenen. Ook met Python kun je tekenen, dat gaat met iets dat 'Turtle' heet.

Ga naar deze Python-code: <https://repl.it/@Felienne/Kunstles-start>

Dit staat al voor je klaar:

```
import turtle

pen = turtle.Turtle() # hier maken we de pen aan, dat is het driehoekje dat
te kent
pen.speed(1) # zet de snelheid langzamer, dan zie je alles beter
```

Deze code is het al het ware verplicht om de tekentool op te starten. Spreek goed met leerlingen af dat ze hier niet aan rommelen want dan werkt het niet meer goed.

Onder deze regels kan het echte tekenen beginnen. Dezelfde dingen die we op papier hebben gedaan, kunnen ook in Python, met deze codes:

- `pen.forward(10)` betekent ga 10 vooruit. Niet 10 cm, maar 10 pixels! 100 is een betere voorbeeld voor op het digibord, anders zien lln het niet goed.
- `pen.left(90)` betekent draai de pen linksom, 90 graden

Met die stappen kunnen we dus al een vierkant tekenen.

Doe dat eerst voor op het bord, waarbij je duidelijk aangeeft welke stappen welke betekenis hebben in Python, ik doe dat zo:



image-20181207094109380

Als ze dit goed onder de knie hebben, dan kun je de Python code onderaan het document toevoegen:

```
pen.forward(100)
pen.left(90)
```

```
pen.forward(100)
pen.left(90)
```

```
pen.forward(100)
pen.left(90)
```

```
pen.forward(100)
pen.left(90)
```

De lege regels hoeven niet, maar ik vind ze wel overzichtelijk, zo zie je goed dat het vier stappen zijn.

Dit gaan leerlingen ook oefenen op hun aftekenvellen, dus het is goed als ze dit even voor zich zien.

Nog een aantal tips:

- Lees de codes nog even hardop:
- pen punt forward ronde haak openen 10 ronde haak sluiten
- pen punt left ronde haak openen 90 ronde haak sluiten

- Stip ook even de betekenissen (en uitspraak) hier aan:
- forward betekent "ga vooruit"
- left betekent links, maar in deze code betekent het "draai linksom", dus niet "ga naar links"

Na deze uitleg kun je de leerlingvellen met lesdoelen 2, 3, 4 en 5 uitdelen!