

# Teken een vierkant

Even kijken of je het nog weet hoor van voor de kerstvakantie!

Tip: je hebt deze Python-opdrachten nodig

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

Je mag iedere opdracht 4 keer gebruiken.

Deze les wordt afgetekend als:

- Je een vierkant hebt getekend.

# Teken een vierkant met herhaling!

Even kijken of je dit nog weet hoor van voor de kerstvakantie...!

Tip: je hebt deze Python-opdrachten nodig

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

Let op!! Je mag iedere opdracht maar één keer gebruiken. Er moet dus ook nog een `for` bij.

Deze les wordt afgetekend als:

- Je een vierkant hebt getekend
- De opdrachten `pen.forward(100)` en `pen.left(90)` ieder maar 1 keer gebruik zijn
- Er een `for`-opdracht in je code staat

## Teken een 'net niet vierkant'!

Dit is een 'net niet vierkant':

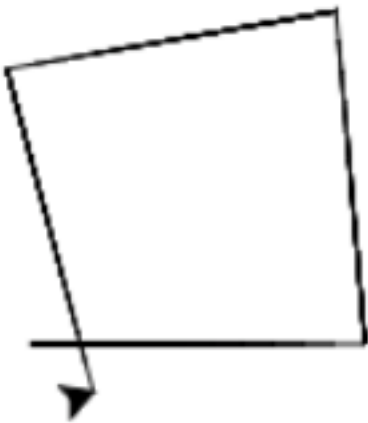


image-20190107134851620

Dat teken je door 4 keer te herhalen, maar net iets meer dan 90 te draaien. Dan kom je net op een andere plek terecht. Neem als waarde: 95.

Deze les wordt afgetekend als:

- Je het figuur hierboven hebt getekend
- Je daarvoor gebruik maakt van een `for`-opdracht

## Teken een 'net niet vierkant' in kleur (extra)

Dit is een 'net niet vierkant' waarbij iedere zijde een andere kleur heeft.

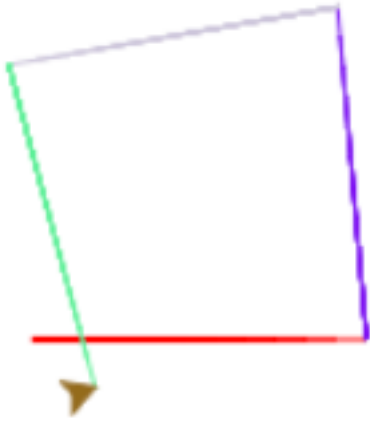


image-20190107135108335

Deze les wordt afgetekend als:

- Je het figuur hierboven hebt getekend
- Met iedere zijde in een andere kleur (het mogen andere kleuren zijn dan hierboven)
- Je daarvoor gebruik maakt van een for-opdracht

# Je eerste spirograaf-figuur

Als je verder tekent met het 'net niet vierkant', dan krijg je na een tijdje dit:

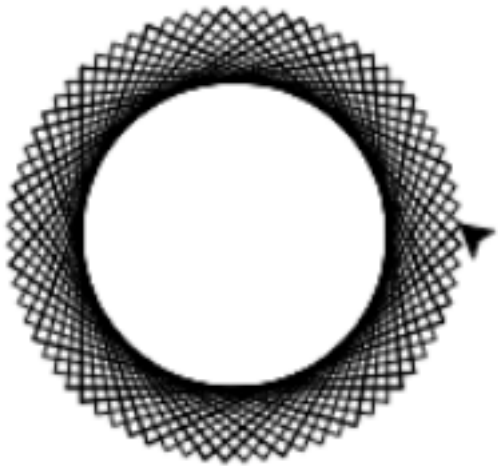


image-20190107135301923

Dat doe je dus door de tekening verder te herhalen. Weet jij welk getal je dan moet aanpassen?

Er zitten nu 3 getallen in je code als het goed is: 4, 100 en 95.

```
for i in range(4):  
    t.forward(100) # beweeg de turtle vooruit  
    t.left(95)     # draai linksom
```

Pas de goede aan!

Deze les wordt afgetekend als:

- Je het figuur hierboven hebt getekend
- Je daarvoor gebruik maakt van een for-opdracht

## Je spirograaf-figuur in een (willekeurige) kleur

Zwart is natuurlijk een beetje saai! Zou het niet leuker zijn als het een mooiere kleur had. Teken het figuur nu in kleur. Je mag zelf een kleur kiezen, of steeds een willekeurige kleur laten kiezen door de computer.

Als je een willekeurige kleur wil, gebruik dan deze code.

```
t.color(random.randint(0,255), random.randint(0,255), random.randint(0,255))
```

Weet jij op welke plek de code moet?

Deze les wordt afgetekend als:

- Je het spirograaf-figuur in een andere kleur van zwart hebt getekend

# Een spirograaf-figuur op willekeurige grootte

Steeds hetzelfde figuur is ook saai! We gaan daarom de computer een grootte laten kiezen.

We kiezen een grootte tussen de 10 en 250.

Dat doen we zo:

```
random.randint(10,250)
```

Dit getal moet worden opgeslagen in een variabele. Die variabele noemen we grootte.

Een van de getallen moet nu vervangen worden door grootte.

Maak jij de code af?

```
grootte = ...
```

```
for i in range(100):
```

```
    t.forward(100) #<--- moet grootte hier?
```

```
    t.left(95)     #<--- of moet grootte hier?
```

Nu kun je hele grote figuren krijgen,

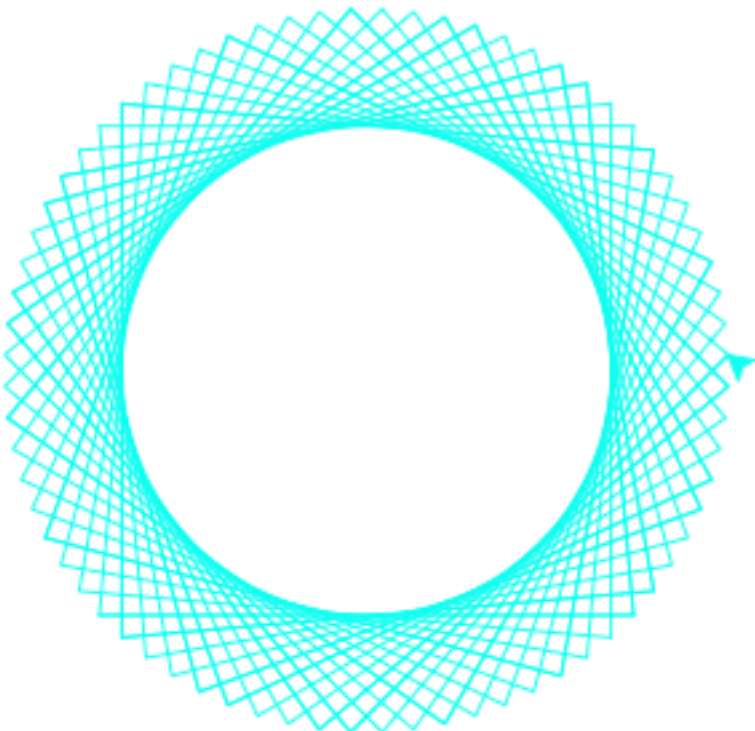


image-20190107143644430

of piepkleintjes, zo:



image-20190107143600261

Deze les wordt afgetekend als:

- Je het spirograaf-figuur hebt getekend
- Het figuur steeds een andere grootte heeft

# Een spirograaf-figuur met een andere vorm

Okee, nu is de grootte steeds anders, maar het is toch hetzelfde figuurtje. Daarom gaan we de computer ook een **hoek** laten kiezen om te draaien.

We kiezen een hoek tussen de 40 en 170.

Weet jij hoe dat moet? Maak deze code maar af:

```
random.randint(...,...)
```

Dit getal moet worden opgeslagen in een tweede variabele. Die variabele noemen we **hoek**.

Maak jij de code af? Kijk goed naar wat je in de vorige stap hebt gedaan:

```
grootte = random.randint(10,250)
```

```
... = random.randint(...,...)
```

```
for i in range(100):
```

```
    t.forward(...)
```

```
    t.left(grootte)
```

Nu krijg je steeds andere vormen, bijv deze:

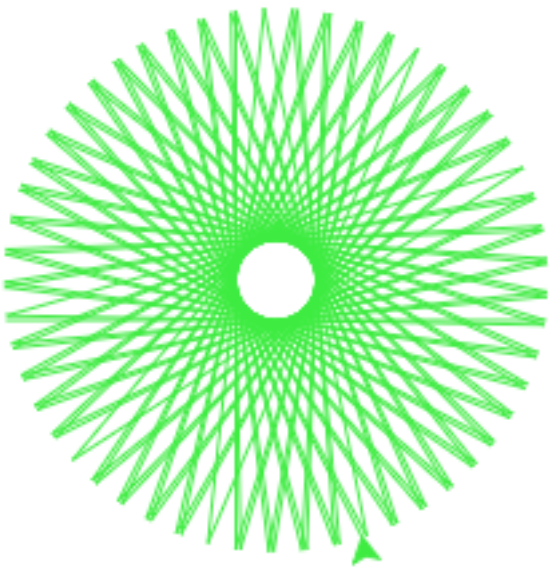


image-20190107144246360

of deze:





image-20190107144305391

Deze les wordt afgetekend als:

- Je het spirograaf-figuur hebt getekend
- Het figuur steeds een andere grootte heeft
- Het figuur steeds een andere vorm heeft

# Op een andere plek

Het figuurtje komt nu steeds in het midden te staan. Maar als we straks meerdere figuren willen maken, dan komen ze over elkaar. Dat is niet handig. We gaan dus steeds naar een andere plek.

Met deze code ga je naar een andere plek:

```
t.setpos(..., ...)
```

Probeer maar eens boven aan je code te zetten, meteen boven de for-opdracht

```
t.setpos(-200, 200)
```

Waar komt het figuur nu steeds te staan?

Dat werkt wel, maar... Nu komt er steeds een streep uit het midden!

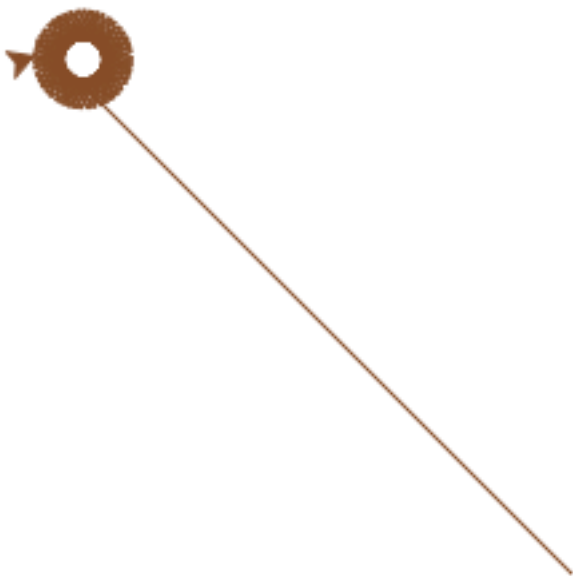


image-20190107144744187

Dat komt omdat de pen aan het begin van het programma automatisch omlaag gaat. Die moet dus eerst omhoog. Pas als je echt gaat tekenen moet die dan weer omlaag. Deze codes

zijn omhoog en omlaag:

```
t.penup()
```

```
t.pendown()
```

Zet jij deze codes op de goede plek in jouw programma?

Dan krijg je een mooi figuur zonder streep:

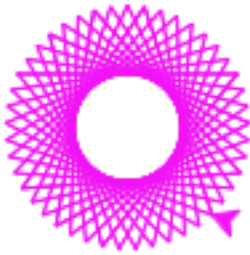


image-20190107150620589

Deze les wordt afgetekend als:

- Je een spirograaf-figuur hebt getekend dat linksbovenin getekend wordt
- Het figuur steeds een andere grootte heeft
- Het figuur steeds een andere vorm heeft
- Er **geen** gek streepje in staat zoals bij het bruine figuurtje hierboven

# Op een willekeurige plek

Nu gaat het figuurtje steeds naar -200, 200, dat is linksbovenin. Maar eigenlijk willen we dat het steedsop een andere plek komt te staan.

Daarvoor gaan we steeds... een willekeurig getal kiezen natuurlijk! Tussen de -200 en de 200.

Vul jij de codes in? Je moet twee keer de code voor willekeurig getal invullen, **op allebei de puntjes dus!**

```
grootte = random.randint(10,250)
```

```
hoek = random.randint(10,250)
```

```
t.penup()
```

```
t.setpos(... , ...) #<-- hier moeten de codes komen!
```

```
t.pendown()
```

```
for i in range(100):
```

```
    t.forward(hoek)
```

```
    t.left(grootte)
```

Deze les wordt afgetekend als:

- Je een spirograaf-figuur hebt getekend **dat steeds op een andere plek** getekend wordt
- Het figuur steeds een andere grootte heeft
- Het figuur steeds een andere vorm heeft

## Meerdere figuren (extra)

Wat is beter dan een spirograaffiguur? Precies! Meerdere spirograaffiguren.

We kunnen alle code die we hebben herhalen, door de code in een for-opdracht te zetten.

Volg deze stappen:

1. Ga boven de regel `grootte = random.randint(10,250)` staan
2. Typ daar een for-opdracht die 5 keer herhaalt. Weet jij nog hoe die moet?
3. Selecteer alle regels onder de nieuwe for.
4. Duw op tab, zodat alle regels 'inspringen'

Probeer je code uit! Krijg je nu 5 figuren?

Deze les wordt afgetekend als:

- Je 5 spirograaf-figuren hebt getekend **die steeds op een andere plek** getekend worden
- De figuren een andere grootte hebben
- De figuren een andere vorm hebben
- Je twee for-opdrachten hebt gebruikt, in elkaar

# Maak het mooi

Je hebt nu een hele hoop willekeurige figuren gezien. Welke vond jij het mooist? Stel alles zo in zodat jij helemaal tevreden bent. Dat mag met of zonder willekeurige getallen erin. Als je tevreden bent, vraag dan aan je leraar om de tekening op te slaan of uit te printen.

Deze les wordt afgetekend als:

- Jij je kunstwerk mooi vindt!