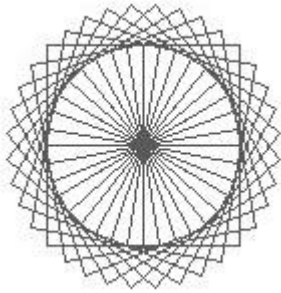


# Logo, een vergeten taal? (4)

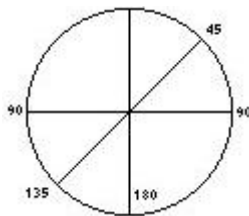
De vorige keer sloot ik af met de vraag:



*hoe is deze tekening gemaakt?*

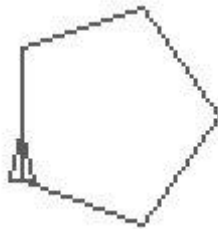
Als u goed kijkt ziet u dat de basis een vierkant is, die telkens een paar stapjes opschuift naar rechts (of links), helemaal rond in een cirkel.

Een cirkel is 360 graden

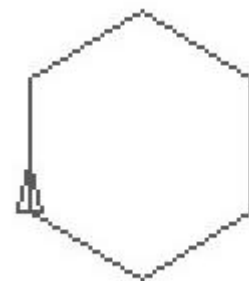


De grootte van de hoek hangt af van het aantal stappen of zijden dat je wilt maken. Om de juiste hoekgrootte te verkrijgen, dient men het aantal graden te delen door het aantal hoeken of zijden. Voor een vijfhoek wordt elke hoek  $360 : 5 = 72$  graden; voor een zeshoek  $360 : 6 = 60$  graden.

to **vijfhoek**  
repeat 5 [fd 50 rt 72]  
end

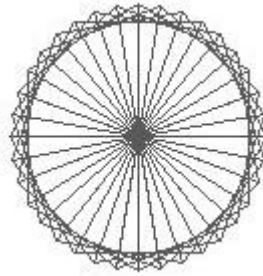
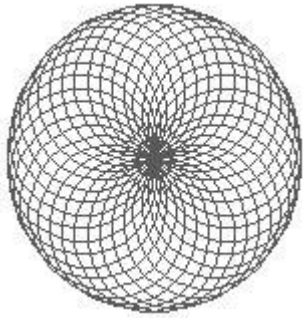


to **zeshoek**  
repeat 6 [fd 50 rt 60]  
end



Voor de overige veelhoeken (zeven-, acht-, negenhoek, enz. geldt hetzelfde. Een zevenhoek krijgt een hoek van  $360 : 7 = 51,5$  graden (afgerond); een achthoek  $360 : 8 = 45$  graden.

Terug naar de ster. De ster kan - behalve met een vierkant - ook met een



**cirkel** of **driehoek** als basis getekend worden.

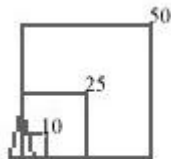
Alle tekeningen zijn vanuit het centrum van het scherm getekend, daar waar de turtle terecht komt als je 'home' intikt. Er verschijnt dan nog steeds een spoor. Om de turtle te verplaatsen zonder te tekenen wordt het commando penup (pu) gebruikt, dan move <n> om hem op een andere plek te zetten en pendown (pd) om vanaf daar weer verder te gaan met tekenen. Alvorens een nieuwe tekening te beginnen, kun je ook het clearscreen commando (cs) intikken

De lengte van zijden is niet vaststaand, maar kan variabel gemaakt worden. De dubbele punt geeft de variabele aan:

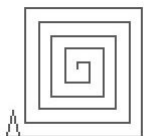
```
to vierkant :zijde  
repeat 4 [ fd :zijde rt 90 ]  
end
```

Aanroepen gaat dan niet meer met alleen 'vierkant', doch met vierkant en de gewenste lengte van de verschillende zijden. LadyBug zal dit vermelden met: 'vierkant needs more inputs'.

```
vierkant 10  
vierkant 25  
vierkant 50
```



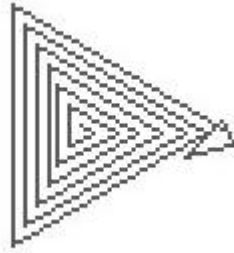
Een van de belangrijkste eigenschappen van een Logo procedure is, dat hij zichzelf in zijn eigen procedure kan oproepen en dat voortdurend herhaalt tot de taak volledig is uitgevoerd. Dit noemt men recursiviteit. Zo'n procedure kan heel lang tot in het oneindige doorgaan. Veelal wordt daarom om te stoppen <escape> of F1 gebruikt. Het is ook mogelijk een wachtlus in te bouwen, zoals in onderstaande procedure getoond wordt.



```
to spiraal :zijde  
if :zijde > 100 [stop]  
fd :zijde  
rt 90  
spiraal :zijde +5  
end
```

Er kunnen meerdere variabelen worden benoemd in een procedure:

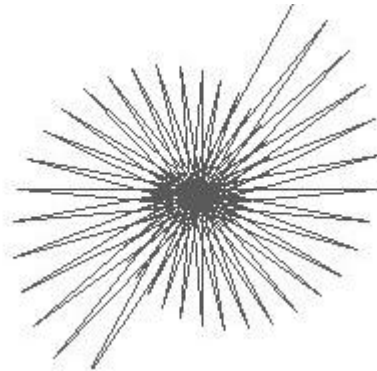
```
to spiraal :zijde :hoek  
if :zijde > 100 [stop]  
fd :zijde  
rt :hoek  
spiraal :zijde +5 :hoek  
end
```



Het aanroepen van deze spiraal gaat met twee getallen. Probeer eens: spiraal 0 90 of spiraal 0 91

Het gebruik van 3 variabelen 'spiraal 10 175 5' geeft dit resultaat

```
to spiraal :zijde :hoek :grootte  
fd :zijde  
rt :hoek  
spiraal :zijde + :grootte :hoek :grootte  
end
```



### ***Met kleur meer variatie.***

De tekeningen hoeven niet alleen zwart-wit te zijn, maar kunnen ook gekleurd worden. Dat gaat door de pen een kleurtje te geven met pencolor (pc) en keuze uit de kleuren 0 tot 4, of de achtergrond te kleuren met het commando background (bg), kiezend uit de kleuren 0 tot 16.

Vraagt u zich - net als ik trouwens - nu ook af of Logo eigenlijk wel geschikt is voor kinderen?

<wordt vervolgd>

Janny Looyenga.