

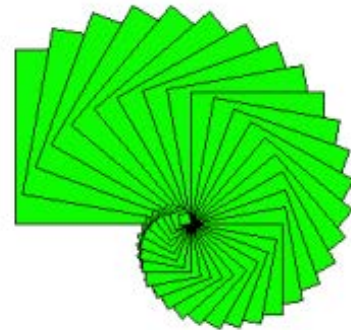
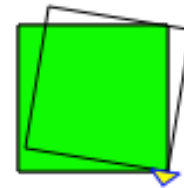


Iteraties: FOR loop

Bepaalde en onbepaalde herhaling

□ **While-loop:** Onbepaalde herhaling:

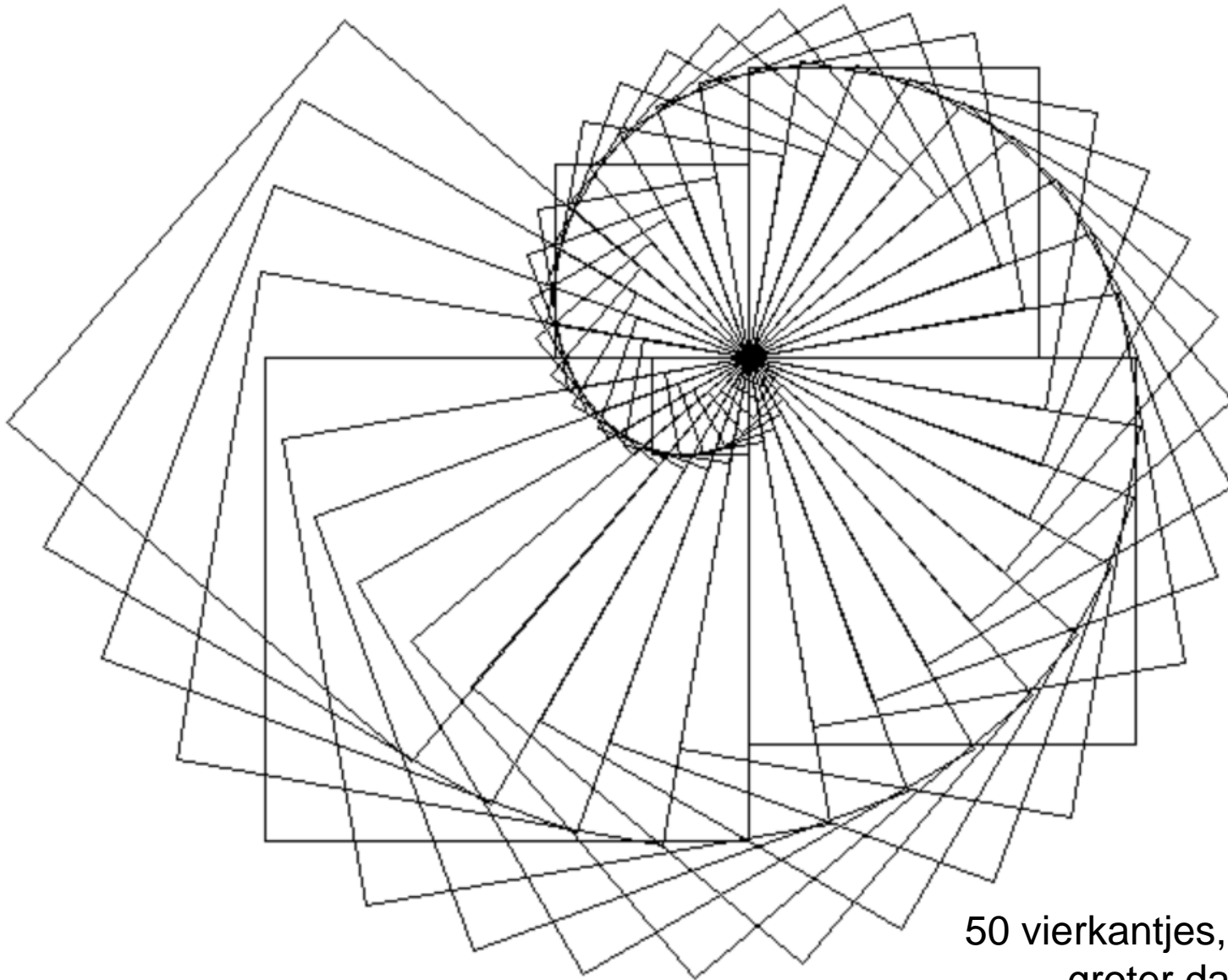
- Zolang ... herhaal
- Aantal keren herhalen afhankelijk van een voorwaarde
- Zorg dat while-loop eindigt! Voorkom oneindige loop!



□ **For-loop:** Bepaalde (of tellende) herhaling:

- Herhaal x keer
- Aantal herhalingen vooraf precies bekend
- Voorbeeld 6 keer

For loop: bepaald aantal herhalingen

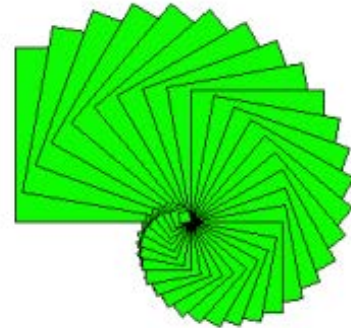
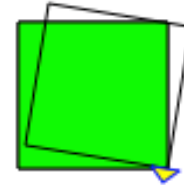


50 vierkantjes, ieder steeds iets
groter dan de vorige

Bepaalde en onbepaalde herhaling

□ **While-loop:** Onbepaalde herhaling:

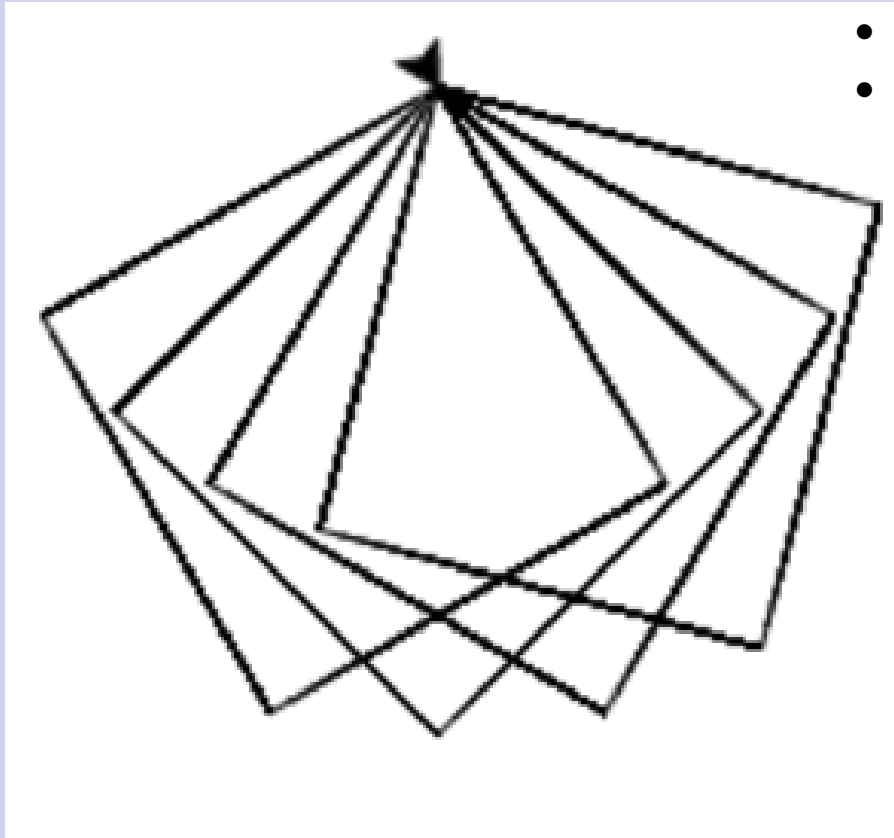
- Zolang ... herhaal
- Aantal keren herhalen afhankelijk van een voorwaarde
- Zorg dat while-loop eindigt! Voorkom oneindige loop!



□ **For-loop:** Bepaalde (of tellende) herhaling:

- Herhaal x keer
- Aantal herhalingen vooraf precies bekend
- Voorbeeld 6 keer

Herhaligen tekenen



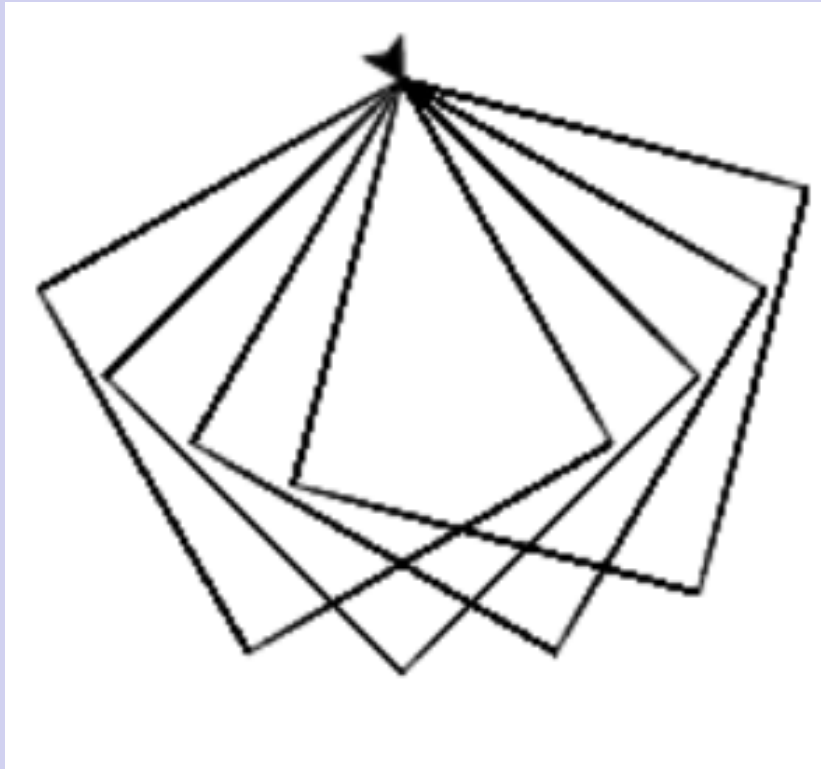
Hoe teken je dit?

- 4 vierkantjes
- Tussendoor 15 graden naar rechts draaien

Probeer het zelf:

- 1) Log in bij repl.it
- 2) Ga naar <http://course.cs.ru.nl/pythonVO>
- 3) Kies Turtle Graphics – For loop
- 4) Maak opgave Vierkantjes herhalen

Slimmer Herhalingen tekenen



Wil je iets x keer herhalen:

```
for teller in range (x):  
    #typ hier wat je wilt herhalen
```

inspringen

Opgave Vierkanten herhalen met een for-loop:

Je wilt 4 keer herhalen:

```
tekenVierkant( 100 )  
turtle.right( 15 )
```

```
for teller in range (4):  
    tekenVierkant( 100 )  
    turtle.right(15)
```



For loop voor herhalingen

```
for x in range(5):  
    print (“Dit wordt 5 keer afgedrukt!”)
```

Resultaat

```
Dit wordt 5 keer afgedrukt!  
Dit wordt 5 keer afgedrukt!  
Dit wordt 5 keer afgedrukt!  
Dit wordt 5 keer afgedrukt!  
Dit wordt 5 keer afgedrukt!
```



For loop

Bijvoorbeeld voor het afdrukken van een rij getallen:

```
for x in range(5):  
    print( x )
```

Resultaat:

0
1
2
3
4

Let op:

- er wordt 5 keer herhaalt,
- maar telling begint bij 0.



Vermenigvuldigingstabel maken

- Schrijf een programma dat de tafel van vermenigvuldiging van het 12 afdruckt, voor de getallen 1 tot en met 10:

1 * 12 = 12

2 * 12 = 24

..

10 * 12 = 120

Ga naar Hoofdstuk 7b - Iteraties: For-loop

Afsluitende opgave 7.10

```
# Tafel van vermenigvuldiging (met een for loop)

tafel = 12
for teller in range (1, 11):
    antwoord = teller*tafel
    print( teller, "*", tafel, "=", antwoord )
```



For met range parameters:

`range (begin, einde, stapgrootte)`

begin: de eerste waarde (telt wel mee)

einde: de laatste waarde (telt zelf **niet** mee)

stapgrootte: stap (mag bv. ook -1 zijn)

```
for x in range(5):  
    print(x)
```

Resultaat:

0
1
2
3
4

```
for x in range(1,11):  
    print(x)
```

Resultaat:

1
2
3
..
10

```
for x in range(1,11,3):  
    print(x)
```

Resultaat:

1
4
7
10

Vermenigvuldigingstabellen maken

- En als je nu ook nog alle tafels t/m 12 wilt afdrukken

1*1 = 1	1*2=2	...	1*12=12
2*1 = 2	2*2=4	...	2*12=24
..
10*1=10	10*2=20	...	10*12=120

- Zelfde als ervoor...
1 tafel afdrukken...

```
# Tafel van vermenigvuldiging (met een for loop)

tafel = 12
for teller in range (1, 11):
    antwoord = teller*tafel
    print( teller, "*", tafel, "=", antwoord )
```

- Maar nu uitbreiden dat ook de tafel doortelt van 1 t/m 12
- Ga naar Hoofdstuk 7b - Iteraties: For-loop
- **Afsluitende opgave 7.11**

```
# Tafels van 12 t/m 10
for teller in range (1, 13):
    for teller in range (1, 11):
        antwoord = teller*tafel
        print( teller, "*", tafel, "=", antwoord )
    print()
```



Geneste for loop – loop in een loop

- Ook als je bijvoorbeeld coördinaten wilt afdrukken

```
for x in range(5):  
    for y in range(5):  
        print(x,y)
```

- Resultaat:

0 0	1 0		4 0
0 1	1 1		4 1
0 2	1 2	...	4 2
0 3	1 3		4 3
0 4	1 4		4 4



For met range parameters:

`range (begin, einde, stapgrootte)`

begin: de eerste waarde (telt wel mee)

einde: de laatste waarde (telt zelf **niet** mee)

stapgrootte: stap (mag bv. ook -1 zijn)

```
for x in range(5):  
    print(x)
```

Resultaat:

0
1
2
3
4

```
for x in range(1,11):  
    print(x)
```

Resultaat:

1
2
3
..
10

```
for x in range(1,11,3):  
    print(x)
```

Resultaat:

1
4
7
10