

Opdracht 2b-1) Maak een zeshoek met een for-lus (extra)

Deze les wordt afgetekend als:

- Je een zeshoek hebt getekend met Python
- Je daarvoor een forlus hebt gebruikt

Vorige week heb je een zeshoek getekend. Dat ging zo:

```
pen.forward(100)
pen.left(60)
```

```
pen.forward(100)
pen.left(60)
```

```
pen.forward(100)
pen.left(60)
```

```
pen.forward(100)
pen.left(60)
```

```
pen.forward(100)
pen.left(60)
```

```
pen.forward(100)
pen.left(60)
```

Opdracht. Teken een zeshoek met een for-lus.

Denk aan deze tips:

- Vergeet de dubbele punt : niet aan het einde van de regel
- Regels in de lus moeten beginnen met 2 spaties!
- Als je het goed doet, heb je maar 3 regels nodig voor de zeshoek
- De regels die we al hadden staan tellen daarvoor niet mee (import, en pen= en pen.speed)

Maak een vierkant met een for-lus (extra)

Deze les wordt afgetekend als:

- Je weet welke regels we moeten veranderen om een vierkant te maken.
- Je een vierkant hebt getekend met Python
- Je daarvoor een forlus hebt gebruikt

In de vorige opdracht heb je een zeshoek getekend met een for-lus. Dat ging zo:

```
for i in range(6):  
    pen.forward(100)  
    pen.left(60)
```

Opdracht. Als je nu een vierkant wilt maken, moet je de getallen 6 en 60 veranderen. Waarin? Schrijf het hieronder op:

- 6 wordt
- 60 wordt ...

Opdracht. Teken nu het vierkant met een for-lus

Tip: Je hoeft niets anders te veranderen, alleen de getallen.

Maak een vierkant met een variabele (extra)

Deze les wordt afgetekend als:

- Je de juiste som voor alle figuren hebt opgeschreven, met de twee variabelen
- Je een vierkant hebt getekend met Python
- Het veranderen van alleen de regel `aantal_hoeken = ...` een ander figuur oplevert

In vorige opdrachten heb je een vierkant, een driehoek en toen een zeshoek getekend. Je gebruikt steeds dezelfde som om uit te rekenen welke hoek je gaat draaien, kijk maar:

Bij de driehoek: de hoek is 360 gedeeld door 3
Bij het vierkant: de hoek is 360 gedeeld door 4
Bij de zeshoek: de hoek is 360 gedeeld door 6

We gaan het aantal hoeken nu in een variabele opslaan.

Dat doen we zo voor de driehoek:

```
aantal_hoeken = 3
```

Opdracht. Nu moet jij de hoek uitrekenen, hoe moet dat?

- Schrijf het eerst op in woorden:

hoek = aantal_hoeken

- Schrijf het nu op in Pythoncode

hoek = aantal_hoeken

Deze code kun je nu gebruiken in je for-lus. Dan verandert de hoek vanzelf als je het aantal hoeken verandert!

Opdracht. Maak deze code af:

```
aantal_hoeken = 4
```

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

```
    hoek = .....
```

```
    pen.forward(100)
```

je code werkt goed als je alleen het aantal hoeken aanpast in de eerste regel, en je dan vanzelf een ander figuur krijgt. Dus aantal_hoeken = 3 maakt een driehoek en aantal_hoeken = 10 maakt een tienhoek.