Datum:	Naam:		Nr:	Klas:
snummer:	Punten:			3 De iteratie

1 Begrip en voorbeelden /1

Een sequentiestructuur zoals in bijgevoegd programma is erg **statisch**. Omdat…

✍ Vul het antwoord aan in onderstaande tekst met het juiste aantal:

Indien men op deze manier de getallen 1 t.e.m. 100 wil weergeven, dient men _____ programmaregels te schrijven. Als men daarna de getallen 1 t.e.m. 50 wil weergeven, dient men opnieuw _____ regels te verwijderen. Daarvoor is een **dynamisch** programma dat gebruik maakt van een **herhaal- of iteratiestructuur beter geschikt**. Er zijn **verschillende herhaalstructuren** mogelijk.

```
getal = 1
print(getal)

getal = getal + 1
print (getal)

getal = getal + 1
print(getal)
```

2 De begrensde herhaling /0,5

Gebruik de begrensde herhaling als je **op voorhand weet hoeveel keer** de opdrachten moeten herhaald worden.

Voorbeeld

```
som = 0
for x in range (0,5):
        som = som + x
print("Het totaal is: ", som)
```

De **range()-functie** gebruik je als volgt:

- range(5): x krijgt opeenvolgend de waarden 0, 1, 2, 3 en 4;
- range(1,11): x krijgt opeenvolgend de waarden 1, 2, 3, 4, 5, 6, 7, 8, 9 en 10;
- range(1,11,2): x krijgt opeenvolgend de waarden 1, 3, 5, 7 en 9;
- range(10, 0,-1): x krijgt opeenvolgend de waarden 10, 9, 8, 7, 6, 5, 4, 3, 2 en 1.

✍ Noteer de uitvoer van bovenstaand programma: _____

3 De voorwaardelijke herhaling /0,5

Weet je niet op voorhand hoeveel keer de opdrachten moeten herhaald worden, zet dan de herhaling stop d.m.v. een **voorwaarde**. **Zolang** aan de voorwaarde voldaan is (TRUE), worden de opdrachten herhaald. Is niet of niet meer aan de voorwaarde voldaan (FALSE) dan gaat het programma verder met de opdracht na de herhaalstructuur.

Voorwaardelijke herhaling

```
som = 0
getal = 1
while getal != 10:
        som += getal        <- TRUE
        getal += 1          <- TRUE
print("Het totaal is: ", som)   <- FALSE
```

✍ Noteer de uitvoer van bovenstaand programma: _____

Merk op dat je elke begrensde herhaling kan omzetten in een voorwaardelijke herhaling.

4 Oneindige herhalingen en herhalingen die nooit plaats vinden /1

Het **gevaar** bij het opstellen van een voorwaardelijke herhaling is dat je opdrachten **oneindig** laat herhalen. Dit is het geval als de voorwaarde altijd **True blijft** en **NOOIT** naar **False** wordt **omgezet**.

✍ Plaats in bovenstaand programma een sterretje naast de opdracht die ervoor zorgt dat de voorwaarde kan veranderen van True naar False.

Bij een voorwaardelijke herhaling is het mogelijk dat de te herhalen **opdrachten NOOIT** worden **uitgevoerd**. Dit is het geval als **meteen** aan de voorwaarde voldaan = **True** is.

✍ Noteer de opdracht wanneer dat volgens bovenstaand programma het geval zou zijn: _____

5 Oefening: Tafel van vermenigvuldiging /3

Schrijf een programma **Tafels1.py** dat van een willekeurig ingevoerd getal de tafel van vermenigvuldiging weergeeft.

```
Tafels van vermenigvuldiging
Geef de gewenste tafel: 7
 1 *  7 =  7
 2 *  7 =  14
 3 *  7 =  21
 4 *  7 =  28
 5 *  7 =  35
 6 *  7 =  42
 7 *  7 =  49
 8 *  7 =  56
 9 *  7 =  63
10 *  7 =  70
```

Laat controleren door de leerkracht.

Ontwerp het programma **Tafels2.py** dat alle tafels van vermenigvuldiging **onder elkaar afdrukt**. Hier vind je een deel van de uitvoer:

```
De tafel van 9:
  1 *  9 =   9
  2 *  9 =  18
  3 *  9 =  27
  4 *  9 =  36
  5 *  9 =  45
  6 *  9 =  54
  7 *  9 =  63
  8 *  9 =  72
  9 *  9 =  81
 10 *  9 =  90

De tafel van 10:
  1 * 10 =  10
  2 * 10 =  20
  3 * 10 =  30
  4 * 10 =  40
  5 * 10 =  50
  6 * 10 =  60
  7 * 10 =  70
  8 * 10 =  80
  9 * 10 =  90
 10 * 10 = 100
```

Laat controleren door de leerkracht.

6 Oefening: Hoger Lager /4

Ontwerp het programma **HogerLager.py** waarbij de computer een willekeurig getal van 1 tot 100 genereert, bv. 12. De computer moet het getal raden en doet een willekeurig gok van 1 tot 100, bv. 87. Het zoekdomein wordt beperkt tot 86 en de computer waagt opnieuw een gokje. De computer herhaalt dit proces totdat hij correct gokt. De computer geeft dan het te raden getal weer, samen met het aantal pogingen die nodig waren om het getal te raden.

Uitvoer:

```
1 [1,100] --> computer gokt 87
2 [1,86] --> computer gokt 28
3 [1,27] --> computer gokt 5
4 [6,27] --> computer gokt 16
5 [6,15] --> computer gokt 6
6 [7,15] --> computer gokt 13
7 [7,12] --> computer gokt 10
8 [11,12] --> computer gokt 12
De computer had 8 pogingen nodig om het getal 12 te raden.
```

Laat controleren door de leerkracht.