

Herhaalstructuren



BIN-OWE1

Studiewijzer

Les	Onderwerp	
	Algemeen	Python
1	Linux	
2	Git	Input/output
3	Pseudocode	If/elif/else Booleans
4		For loop
5		Lists and tuples Files (CSV bestanden)
6	Flowchart	Functies
7		Strings CSV bestanden

Studiemateriaal

- Boek: “Starting Out with Python, third edition”
- Reader: “Linux voor Bio-informatici”
- Onderwijs Online
- Kom je er niet uit? → Google



Doelstellingen

- Aan het eind van deze week kan je
 - For en while loops gebruiken in je code
 - Een keuze maken voor de beste loop voor de situatie
- Aan het eind van deze week begrijp je
 - Het verschil tussen count controlled en condition controlled looping

Inhoud

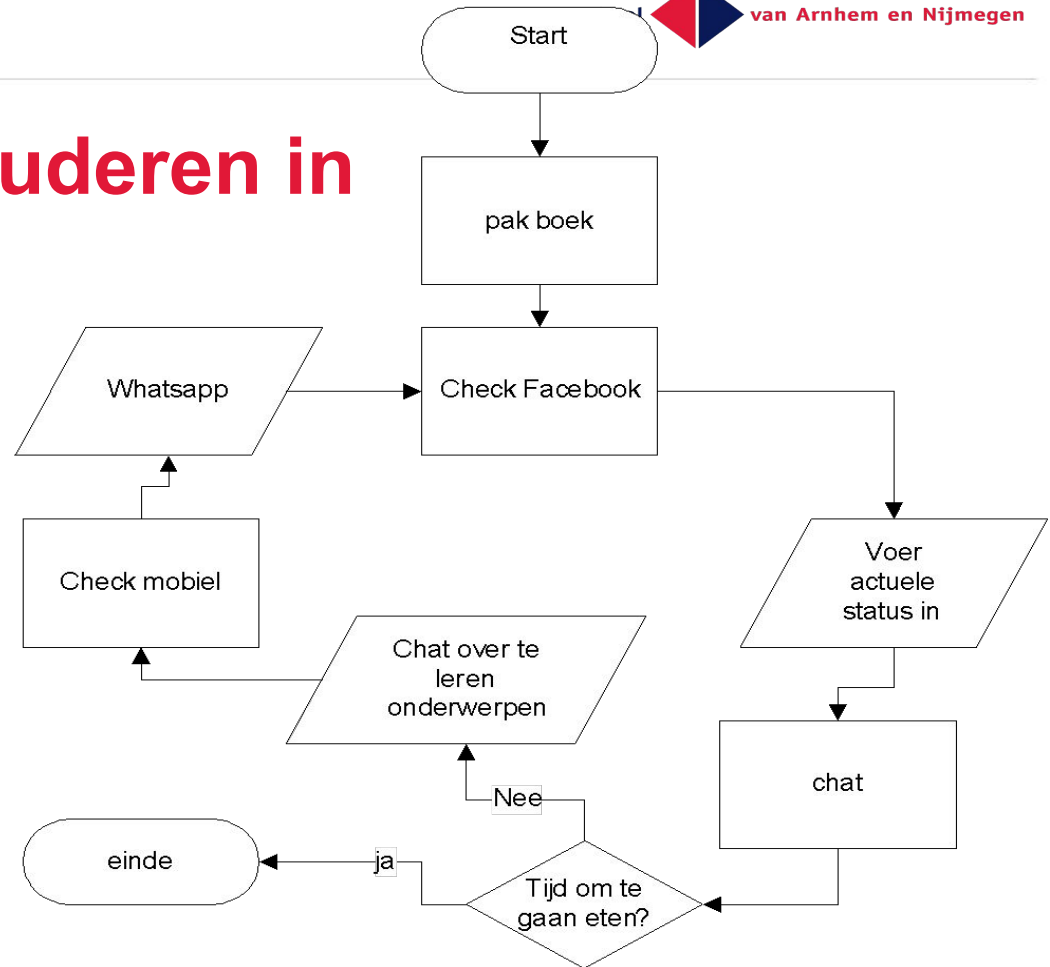
- **Introductie herhaalstructuren**
- While loop
- For loop
- Berekenen van totalen
- Sentinels
- Nested loops
- Input validatie

Vorige keer: beslisstructuren

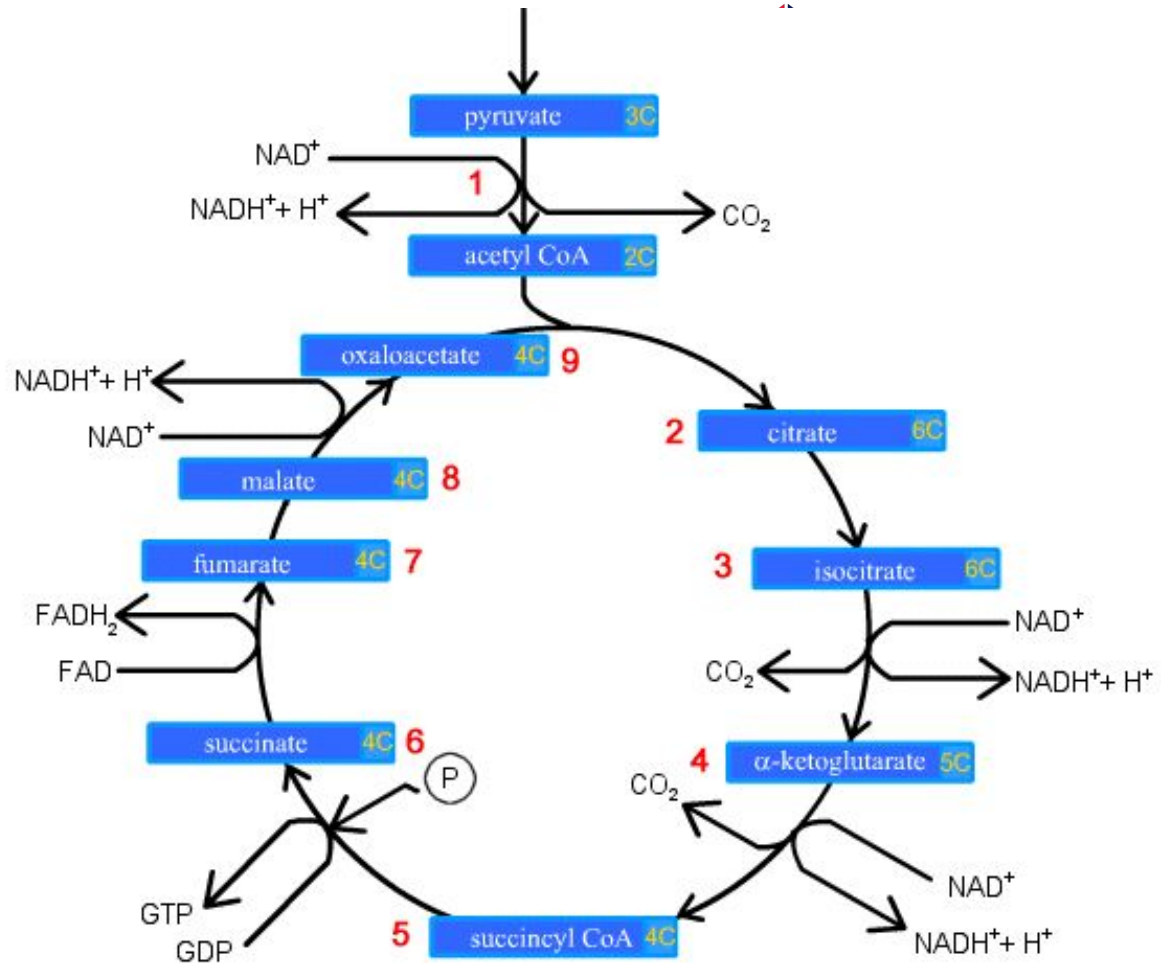
```
if <conditie>:  
    statement(s)  
elif <conditie>:  
    statement(s)  
elif <conditie>:  
    statement(s)  
else:  
    statement(s)
```



Flow van studeren in een loop



Een ander soort loop



Nog een ander soort loop

-



Loops

- Loops draaien om herhaling
- Herhaling op basis van
 - Conditie
 - Aantal
- **Veel woorden voor hetzelfde**
 - Iteratieve controle
 - Herhaalstructuur
 - Loop
 - Repetitieve structuur

Procedureel programmeren

- Drie kenmerken van procedurele programmeertalen
 - Variabelen
 - Conditionele controle
 - Iteratieve controle

Inhoud

- Introductie herhaalstructuren
- **While loop**
- For loop
- Berekenen van totalen
- Sentinels
- Nested loops
- Input validatie

While loop

- While loop is een herhaalstructuur met zogenaamde conditionele controle
- Hetzelfde als bij beslisstructuren
- Zolang een bepaalde voorwaarde (conditie) geldig is, blijft de loop doorgaan

Voorbeeld

```
doorgaan = "ja"  
while doorgaan == "ja":  
    doorgaan = input("Doorgaan? (ja/nee): ")
```

Pretest loop

- De while loop is een zogenaamde pretest loop
- Ofwel, van te voren test je de conditie
 - Is deze waar: doe de loop
 - Is deze onwaar: sla de loop over

Voorbeeld

```
doorgaan = "ja"  
while doorgaan == "ja":  
    doorgaan = input("Doorgaan? (ja/nee): ")
```

Inhoud

- Introductie herhaalstructuren
- While loop
- **For loop**
- Berekenen van totalen
- Sentinels
- Nested loops
- Input validatie

For loop

- For loop herhaalt voor een gegeven aantal waardes
- Is dus een herhaalstructuur met een zogenaamde count controlled loop
- Vooraf wordt dus bepaald hoe vaak de loop doorlopen wordt

Voorbeeld

```
for getal in [1,2,3,4]:  
    print(getal)
```

Range functie

- De range functie is een handige functie om te combineren met de for loop

```
range (10)
```

```
range (4, 10)
```

```
range (4, 10, 2)
```

```
for i in range(4,10):  
    print(i)
```

Wat print deze loop?

Loopen over tekst

- We kunnen ook lopen over tekst

```
for c in "Hello World!":  
    print (c)
```

H

e

l

l

o...



Loopen over woorden

- En ook nog over woorden

```
for w in ["Hello", "World", "!"]:  
    print (w)
```

En wat zou deze printen?

Inhoud

- Introductie herhaalstructuren
- While loop
- For loop
- Berekenen van totalen
- Sentinels
- Nested loops
- Input validatie

Augmented Assignment Operators

Operator	Voorbeeld	Gelijk aan	Uitkomst (a = 12)
<code>+=</code>	<code>a += 4</code>	<code>a = a + 4</code>	16
<code>-=</code>	<code>a -= 4</code>	<code>a = a - 4</code>	8
<code>*=</code>	<code>a *= 4</code>	<code>a = a * 4</code>	48
<code>/=</code>	<code>a /= 4</code>	<code>a = a / 4</code>	3
<code>%=</code>	<code>a %= 4</code>	<code>a = a % 4</code>	0

For loop voorbeeld

```
i = 0
for w in ["Hello", "World", "!"]:
    i += 1
    print (w)
    print (i)
```

Inhoud

- Introductie herhaalstructuren
- While loop
- For loop
- Berekenen van totalen
- **Sentinels**
- Nested loops
- Input validatie

Sentinels

- Letterlijk: schildwachter
- Bewaakt het proces



Voorbeeld

```
totaal = 0
doorgaan = True
while doorgaan == True:
    getal = int(input("Geef een getal: "))
    totaal = totaal + getal
    print("Totaal: ", totaal)
    inv = input("Wil je doorgaan?")
    if inv == "Nee":
        doorgaan = False
```

Korter Voorbeeld

```
totaal = 0
doorgaan = True
while doorgaan == True:
    totaal += int(input("Geef een getal: "))
    print("Totaal: ", totaal)
    inv = input("Wil je doorgaan?")
    if inv == "Nee":
        doorgaan = False
```

Nog Korter Voorbeeld

```
while getal > 0:  
    totaal += int(input("Geef een getal: "))  
    print("Totaal: ", totaal)
```

Maar hoe kom ik hier nu uit?

Inhoud

- Introductie herhaalstructuren
- While loop
- For loop
- Berekenen van totalen
- Sentinels
- **Nested loops**
- Input validatie

Nesting

- Net als bij if-elif-else kan je ook for en while loops nesten
- Dit is dan een geneste loop



Voorbeeld

```
for i in range(1,11):  
    for j in range(1,11):  
        print(i, "*", j, "=", i*j)
```

Wat zou deze geneste for loop doen?

Inhoud

- Introductie herhaalstructuren
- While loop
- For loop
- Berekenen van totalen
- Sentinels
- Nested loops
- **Input validatie**

Controleer input voor verwerken

```
getal = 0
totaal = 0
while getal >= 0:
    getal = int(input("Cijfer: "))
    if getal < 0 or getal > 10:
        print("Dit getal voldoet niet")
    else:
        totaal += getal
        print("Totaal: ", totaal)
```

Samenvatting

- Iteratieve controle
 - While loop
 - Condition controlled
 - Pretest
 - For loop
 - Count controlled

Opdracht

- Ga naar OnderwijsOnline
- Maak Afvinkopdracht 3
- Bij problemen:
 - Google
 - Klasgenoten
 - Docent
- Lever deze in bij de praktijkdocent





Verantwoording

- In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen
- Alle teksten © Martijn van der Bruggen/Esther Kok/HAN tenzij expliciet externe bronnen zijn aangegeven
- Screenshots op basis van eigen werk auteur en/of vernoemde sites en/of fair use
- Eventuele images zijn opgenomen met vermelding van bron