

Informatiebundel

inhaal cursus informaticawetenschappen 'Python'

- ☐ Tweede graad secundair onderwijs, van basis tot "booleans"



Omschrijving

- ☐ Deze cursus is bedoeld voor leerlingen die instromen in het vierde jaar van de tweede graad richting 'Bedrijfswetenschappen' en nog geen voorkennis hebben van de programmeertaal Python.
- ☐ Aan het einde van de cursus staan enkele extra oefeningen. Deze zijn essentieel om te maken. Ze geven je een idee of je de leerstof al dan niet goed beheerst.
- ☐ Alle oefeningen en opdrachten vormen samen de voorbereiding om succesvol met het vak informaticawetenschappen in 4BW te starten.



Wat heb je nodig om deze leerstof in te halen?

- ☐ Werkboek 'Python' van Pelckmans p.6-39 (vanaf p.40 in het 4^{de} jaar)
 - ☐ 18 bijhorende video's
https://www.youtube.com/playlist?list=PL2iW_rkiCt7UqxL47IGkBaTgApj_QHJ0K
- ☐ PDF-doc. "samenvatting + oefeningen"
 - ☐ boek/cursus Python programmeerleerling" (ter ondersteuning van de samenvatting + oefeningen)
- ☐ bundel "stroonschema maken"
- ☐ Deze informatiebundel
- ☐ Computer



Hoe ga je te werk?

- ☐ Gebruik PDF-doc "samenvatting + oefeningen" als een korte inhaal cursus van het boek "Programmeerleerling". Lees elke uitleg aandachtig.

- ☐ Bekijk alle 18 video's die bij je werkboek Python bij horen en maak je eigen samenvatting van elke video-uitleg (link en info in "samenvatting + oefeningen")
- ☐ Maak de oefeningen volledig en test je oplossingen in Thonny (meer uitleg in de "samenvatting + oefeningen"), vergelijk jouw oplossingen indien nodig met de mogelijke oplossingen die bij elke opgave in dit informatiebundel staan. Let op: de getoonde oplossingen zijn niet de enige juiste antwoorden, maar wel geldige varianten. Je kan evenzeer andere correcte oplossingen hebben!

TOELICHTING "samenvatting + oefeningen" DEEL 2



Opgave: Algoritme schrijven voor een boterham met choco

Stel je voor dat iemand een robot is die je moet programmeren om een boterham met choco te smeren. Schrijf (in je eigen woorden) een duidelijk stappenplan waarmee deze persoon die taak goed kan uitvoeren.

Stap 1: Schrijf je eigen stappenplan.

Denk eraan: de machine voert alleen letterlijk uit wat je schrijft. Wees heel precies!

Stap 2: Bekijk het filmpje "Leer programmeren met boterham met choco" (YouTube).

Stap 3: Reflecteer:

- Zou jouw stappenplan tot een goed resultaat geleid hebben?
- Wat liep er mis?
- Hoe pas je je stappenplan aan om het beter te maken?

Stap 4: Verbeter je stappenplan.

Probeer je instructies zo te herschrijven dat een domme robot exact weet wat te doen (debuggen).

Voorbeeld:

1. Neem één boterham uit de zak.
2. Leg de boterham op een bord.
3. Open de pot choco.

4. Neem een mes.
5. Steek het mes in de pot.
6. Schep choco met het mes.
7. Smeer choco op de boterham.

Fout voorbeeld:

1. Smeer choco op je boterham. (niet duidelijk wat eerst moet gebeuren)

Extra: Dit type opdracht toont het belang van een duidelijke volgorde van taken en logica in programmeren. Zelfs simpele taken moeten stap voor stap uitgelegd worden.

Opgave "algoritme voor het berekenen van het aantal boompjes"

☐ Algoritme in stappen (in gewone taal)

1. Bepaal de afmetingen van de akker.
2. Bereken de oppervlakte van het trapezium.
3. Bereken de oppervlakte die één boom nodig heeft.
4. Deel de oppervlakte van het trapezium door de oppervlakte per boom.
5. Rond het resultaat af naar **beneden** (want je kan geen halve boom planten).
6. Druk het aantal bomen af.

TOELICHTING DEEL 5

VERTALEN van het algoritme naar Python "boompjes"

```
# Gegeven afmetingen
```

```
a = 120
```

```
b = 80
```

```
h = 100
```

```
# Bereken de oppervlakte van het trapezium
```

```
oppervlakte_akker = ((a + b) / 2) * h
```

```
# Oppervlakte nodig per boom
```

```
oppervlakte_per_boom = 8 * 8
```

```
# Bereken aantal bomen (afgerond naar beneden)
```

```
aantal_bomen = oppervlakte_akker // oppervlakte_per_boom
```

```
print("De boer moet", int(aantal_bomen), "bomen aankopen.")
```



Opgave: Bussen en overgebleven personen

Schrijf een programma dat berekent hoeveel volle bussen nodig zijn om een groep personen te vervoeren, en hoeveel personen er nog overblijven nadat de bussen gevuld zijn.

Voer in je programma de volgende gegevens in:

- Aantal personen (bijvoorbeeld 137)
- Capaciteit van één bus (bijvoorbeeld 50)

Je programma moet uitrekenen:

1. Het aantal volle bussen dat nodig is
2. Het aantal personen dat overblijft (aantal dat lager is dan het aantal beschikbare plaatsen in een lege bus)

Gebruik hiervoor:

- De operator `//` voor gehele deling (voor het aantal bussen)
- De operator `%` voor de rest (personen die overblijven)

Voorbeeldcode:

```
aantal_personen = 137
```

```
buscapaciteit = 50
```

```
volle_bussen = aantal_personen // buscapaciteit
```

```
overblijvend = aantal_personen % buscapaciteit
```

```
print("Aantal volle bussen:", volle_bussen)
```

```
print("Personen zonder plaats in een bus:", overblijvend)
```

Vragen:

- Wat gebeurt er als er exact genoeg personen zijn om alle bussen te vullen?
- Wat als de groep kleiner is dan één bus?

Extra opdracht:

Laat het programma zelf vragen hoeveel personen er zijn en wat de capaciteit van de bus is (gebruik `input()`).



Opgave: Typefout bij invoer met `input()`

Script:

```
getal = input("Geef getal in:")
```

```
dubbel_getal = 2 * getal
```

```
print("Het ingegeven getal was", getal)
```

```
print("Het dubbele van het getal is", dubbel_getal)
```

Uitleg:

In Python wordt alles wat je invoert via `input()` automatisch behandeld als een string (tekst), ook al voer je een cijfer in. Bijvoorbeeld: als je 5 invoert, dan wordt dit opgeslagen als tekst "5".

Wanneer je dan `2 * getal` doet, krijg je niet de wiskundige vermenigvuldiging (10), maar de string wordt 2 keer na elkaar geplakt: "5" wordt "55". Dit noemt men stringherhaling.

Om dit te vermijden, moet je de invoer eerst omzetten naar een getal met `int()` of `float()`:

Verbeterde versie:

```
getal = int(input("Geef een getal in:"))
```

```
dubbel_getal = 2 * getal  
  
print("Het ingegeven getal was", getal)  
  
print("Het dubbele van het getal is", dubbel_getal)
```

Opgelet: als je een decimaal getal wilt invoeren, gebruik dan `float()` in plaats van `int()`.

TOELICHTING DEEL 6



Opgave "operatoren"

- ☐ Bereken 5 tot de macht 3

```
resultaat = 5 ** 3  
  
print(resultaat) # Output: 125
```

- ☐ Uitleg: `**` is de machtsverheffing in Python. $5^3 = 125$.

- ☐ Bereken de vierkantswortel van 64 (gebruik $(1/2)$ -macht)

```
wortel = 64 ** (1/2)  
  
print(wortel) # Output: 8.0
```

- ☐ Uitleg: De vierkantswortel van een getal = $\text{macht } 0.5 = \text{**(1/2)}$.

- ☐ Wat is de uitkomst van $17 // 4$?

```
print(17 // 4) # Output: 4
```

- ☐ Uitleg: `//` is de gehele deling, dus 17 gedeeld door 4 geeft 4 met een rest.

- ☐ Wat is de rest van $17 \% 4$?

```
print(17 \% 4) # Output: 1
```

- ☐ Uitleg: `%` geeft de rest van de deling. 17 gedeeld door 4 = 4 rest 1.

- ☐ Maak een zin met een naam via concatenatie (+)

```
naam = "Sara"

zin = "Hallo " + naam + ", welkom in de les Python!"

print(zin)
```

- ☐ Uitleg: Strings kun je aan elkaar plakken met +. Dit heet concatenatie.

- ☐

- ☐ Bereken de vierkantswortel van 625 (gebruik (1/2)-macht)

```
resultaat = 625 ** (1/2)

print(resultaat) # Output: 25.0
```

- ☐ Uitleg: $625 ** 0.5$ is hetzelfde als $\sqrt{625} = 25.0$.

Opgave "oppervlakte cirkel"

- ☐ Onderstaand codefragment kent de straal van een cirkel toe aan de variabele *straal*. De waarde van π wordt toegekend aan de variabele *pi*.

- ☐ oplossing:

- ☐ Je krijgt een **startcode** waarin de straal en pi al zijn toegekend. Je moet zelf de berekening van de oppervlakte toevoegen en het resultaat tonen.

- ☐

```
# straal van de cirkel

straal = 2.73

# waarde van pi

pi = 3.141592653589793


# oppervlakte van de cirkel berekenen

oppervlakte = pi * (straal ** 2)


# resultaat afdrukken
```

```
print("De oppervlakte van de cirkel is", oppervlakte)
```

- ☐ # oppervlakte cirkel deel 2 uitbreiding:
- ☐ # waarde van pi
- ☐ pi = 3.141592653589793
- ☐
- ☐ # vraag straal aan gebruiker
- ☐ straal = float(input("Geef de straal van de cirkel in: "))
- ☐
- ☐ # bereken oppervlakte
- ☐ oppervlakte = pi * (straal ** 2)
- ☐
- ☐ # resultaat afdrukken
- ☐ print("De oppervlakte van de cirkel is", oppervlakte)
- ☐ Uitleg
 - input() vraagt de straal aan de gebruiker.
 - float() zorgt ervoor dat een kommagetal juist wordt verwerkt.
 - straal ** 2 berekent het kwadraat van de straal.
 - Het resultaat wordt opgeslagen in de variabele *oppervlakte* en met print() weergegeven.

Opgave "snoepautomaat"

- ☐ Algoritme:
 1. Vraag het bedrag in euro aan de gebruiker.
 2. Zet het bedrag om naar centen (1 euro = 100 cent).
 3. Bereken hoeveel snoepjes je kunt kopen (gedeeld door 25).
 4. Bereken hoeveel cent overblijven (modulo 25).

5. Toon het aantal snoepjes en de rest aan de gebruiker.

```
# vraag bedrag in euro aan gebruiker
bedrag_euro = float(input("Hoeveel euro wil je ingeven? "))

# omzetten naar cent
bedrag_cent = int(bedrag_euro * 100)

# bereken aantal snoepjes
snoepjes = bedrag_cent // 25

# bereken wisselgeld
rest_cent = bedrag_cent % 25

# toon resultaat
print("Je kunt", snoepjes, "snoepjes kopen.")
print("Je krijgt", rest_cent, "cent terug.")
```

Opdracht "Korte scripts schrijven"

- ☐ 1. Functie $f(x) = x^2 - x - 2$

Schrijf een programma dat een x-waarde vraagt, de functie $f(x) = x^2 - x - 2$ berekent en het resultaat toont.

```
x = float(input("Geef een x-waarde in: "))
f = x**2 - x - 2
print("De functiewaarde van f(x) =", f)
```

-
- ☐ 2. Celsius naar Fahrenheit

Zet een ingegeven waarde in °C om naar °F via de formule:

$$^{\circ}\text{F} = (^{\circ}\text{C} \times 9/5) + 32$$

```
celsius = float(input("Geef temperatuur in °C: "))
```

```
fahrenheit = (celsius * 9/5) + 32
```

```
print("Dat is", fahrenheit, "°F")
```

☐ 3. Leeftijd berekenen

☐ Opdracht:

Vraag je geboortjaar, bereken je leeftijd. (Rekening houden met het huidige jaar, bv. 2025.)

```
geboortjaar = int(input("In welk jaar ben je geboren? "))
```

```
huidig_jaar = 2025
```

```
leeftijd = huidig_jaar - geboortjaar
```

```
print("Je bent ongeveer", leeftijd, "jaar oud.")
```

☐ 4. Voornaam compliment

Vraag je voornaam, geef een persoonlijke boodschap.

```
voornaam = input("Wat is je voornaam? ")
```

```
print("Hallo,", voornaam + ", je bent geweldig!")
```

☐ Voorbeeld: voornaam = Sarah >>> Hallo, Sarah, je bent geweldig!

5. Gemiddelde snelheid berekenen

☐ Vraag voor 3 gelopen parcours de afstand (in km) en de tijd (in uren), bereken en toon de gemiddelde snelheid.

```
afstand1 = float(input("Afstand parcours 1 (km): "))
```

```
tijd1 = float(input("Tijd parcours 1 (uur): "))
```

```
afstand2 = float(input("Afstand parcours 2 (km): "))
```

```
tijd2 = float(input("Tijd parcours 2 (uur): "))
```

```
afstand3 = float(input("Afstand parcours 3 (km): "))
```

```
tijd3 = float(input("Tijd parcours 3 (uur): "))
```

```
totaal_afstand = afstand1 + afstand2 + afstand3
```

```
totaal_tijd = tijd1 + tijd2 + tijd3
```

```
gemiddelde_snelheid = totaal_afstand / totaal_tijd
```

```
print("Je gemiddelde snelheid is", gemiddelde_snelheid, "km/u")
```

- ☐ Voorbeeld: totaal 12 km in 1.5 uur >>> Gemiddelde snelheid = 8.0 km/u

Antwoorden p66-86

- ☐ p.66 A
- ☐ p.67 B of D(om eventuele fouten, door het combinatie met afkappingsteken, te vermijden)
- ☐ p.68 C
- ☐ p.69 D
- ☐ p.70 B
- ☐ p.71 D
- ☐ p.72 A
- ☐ p.73 naam(variabele) =(toekennen, assignment) "Kuifje"(waarde)
- ☐ p.74 C
- ☐ p.75 A
- ☐ p.76 A
- ☐ p.77 B
- ☐ p.78 D

- ☐ p.79 B
- ☐ p.80 A
- ☐ p.81 D
- ☐ p.82 A
- ☐ p.83 A
- ☐ p.84 C (Python is hoofdlettergevoelig, dus B is niet b, maar het zijn 2 verschillende namen van de variabelen)
- ☐ p.85 (Test dit in Thonny! het moet 11 zijn, dus het juiste antwoord staat er niet tussen)
- ☐ p.86 D

Bronnen:

Gratis Boek "Programmeerleerling"