

HOOFDSTUK 1

START

Helga Naessens

De eerste 3 vuistregels

- Eerst nadenken, dan programmeren
- Een programma is een leesbare tekst over de oplossing van een probleem en kan op een computer uitgevoerd worden
- Oefening baart kunst: hoe meer je experimenteert hoe beter je programmeert en problemen kan oplossen



Inhoud Hoofdstuk 1

- **Een eerste programma**
- Inhoud programma
- Algoritme
- Testen

Een eerste programma

eerste.py

```
# berekent de oppervlakte en omtrek van een cirkel
# Stap 1: vraag de straal van de cirkel
# Stap 2: bereken de oppervlakte en omtrek
# Stap 3: print de resultaten

import math

straal_str = input("Straal van de cirkel in mm: ")
straal_int = int(straal_str)

omtrek = 2 * math.pi * straal_int # omtrek berekenen
opp = math.pi * (straal_int ** 2) # oppervlakte berekenen
print(f"omtrek: {omtrek:.2f}mm; oppervlakte: {opp}mm^2")
```

Uittesten...



Commentaar

```
# berekent de oppervlakte en omtrek van een cirkel  
# Stap 1: vraag de straal van de cirkel  
# Stap 2: bereken de oppervlakte en omtrek  
# Stap 3: print de resultaten
```

- # ...
- Informatie voor lezer
- Wordt niet uitgevoerd
- Mag ook na een opdracht staan



```
omtrek = 2 * math.pi * straal_int # omtrek berekenen  
opp = math.pi * (straal_int ** 2) # oppervlakte berekenen
```

Module importeren

```
import math
```

- om bestaande programmacode te gebruiken
- `import` staat meestal aan het begin van een programma, maar mag ook vlak voor gebruik staan
- module **math** (zie API) → oplossingen wiskundeproblemen

➤ het getal π

```
math.pi
```

Invoer

```
straal_str = input("Straal van de cirkel in mm: ")
```

- **input(...)**
 - zie API
 - functie: stukje programmacode die een taak uitvoert
 - tekst (tussen " " of ' ') afdrukken
 - resultaat: wat de gebruiker intikt (tekst!)
- **straal_str**
 - variabele
- **=**
 - toekenning

straal_str

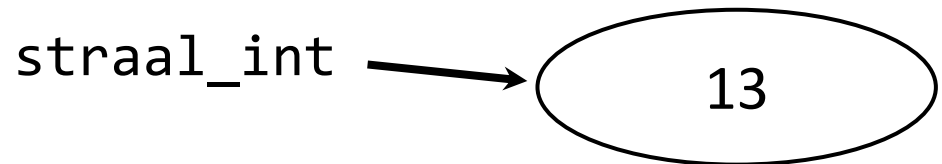


"13"

Conversie

```
straal_int = int(straal_str)
```

- functie **int(...)**
 - tekst → geheel getal
 - string (str) → integer (int)
 - error indien tekst niet kan omgezet worden naar een geheel getal



- functie **float(...)**
 - conversie naar reëel getal

Berekeningen

```
omtrek = 2 * math.pi * straal_int # omtrek berekenen
opp = math.pi * (straal_int ** 2) # oppervlakte berekenen
```

- maal: `*`
- macht: `**`
- andere operatoren: `+` `-`
 - `/` (reële deling) $\Rightarrow 8/3 \neq 8//3$
 - `//` (gehele deling)
 - `%` (modulo = rest na gehele deling)
 - `9%3 = ?`
 - `9%5 = ?`

Uitvoer

```
print(f"omtrek: {omtrek:.2f}mm; oppervlakte: {opp}mm^2")
```

- functie `print(...)`

- informatie op scherm printen (newline op het einde)

- tekst tussen " " of ' '

```
print('omtrek:')
```

- variabele

```
print(omtrek)
```

- combinatie

```
print('omtrek:', omtrek)
```

- combinatie tekst/variabele(n) kan m.b.v. een **f-string**:

- f" " of f' '

- variabelen tussen { }

Een eerste programma

eerste.py

```
# berekent de oppervlakte en omtrek van een cirkel
# Stap 1: vraag de straal van de cirkel
# Stap 2: bereken de oppervlakte en omtrek
# Stap 3: print de resultaten

import math

straal_str = input("Straal van de cirkel in mm: ")
straal_int = int(straal_str)

omtrek = 2 * math.pi * straal_int # omtrek berekenen
opp = math.pi * (straal_int ** 2) # oppervlakte berekenen
print(f"omtrek: {omtrek:.2f}mm; oppervlakte: {opp}mm^2")
```

Inhoud Hoofdstuk 1

- Een eerste programma
- **Inhoud programma**
- Algoritme
- Testen

Inhoud programma

- **Importeren (deel) module**
- Opdrachten en uitdrukkingen
- Witruimte (*whitespace*)
- Commentaar
- Tokens
- Variabelen
- Types
- Tekstopmaak

Als je iets uit een module (zie API) wilt gebruiken, moet je (een deel van) de module importeren

Importeren module

- Om bestaande code voor specifieke problemen te kunnen gebruiken

naam module

```
import module
```

Gebruik: *module*.xxx

naam module

```
import module as var
```

Gebruik: *var*.xxx

Importeren deel module

naam module



```
from module import functie/constante
```

Gebruik: *functie/constante*

(zonder vermelding module)

voorbeeld

```
from math import pi  
print(pi)
```

Module math

- <https://docs.python.org/3/library/math.html>
- Voorbeeld functies

Functie	Betekenis
<code>math.sin</code>	sinus van hoek in radialen
<code>math.cos</code>	cosinus van hoek in radialen
<code>math.degrees</code>	radialen → graden
<code>math.radians</code>	graden → radialen
<code>math.sqrt</code>	vierkantswortel

```
import math
print(math.cos(math.pi))    # -1.0
print(math.sin(math.radians(90))) # 1.0
```