

# BASIS PROGRAMMEREN

## THEORIEBUNDEL

**Copyright: Tom Vanhoutte**

# INLEIDING

## 1. Voorkennis

Deze cursus vereist geen voorkennis.

## 2. Omgeving

De omgeving die gebruikt zal worden om logisch procedureel te leren programmeren is python.

## 3. Wat is een computer programma?

- Een manier om te communiceren (computers, mensen)
- Geschreven in een programmeertaal



Copyright: Tom Vanhoutte

## 4. Soorten programmeertalen?

- Machine taal (taal van de computer)

1		00000000	00000100	0000000000000000
2	01011110	00001100	11000010	0000000000000010
3		11101111	00010110	0000000000000101
4		11101111	10011110	0000000000001011
5	11111000	10101101	11011111	0000000000010010
6		01100010	11011111	0000000000010101
7	11101111	00000010	11111011	0000000000010111
8	11110100	10101101	11011111	0000000000011110
9	00000011	10100010	11011111	0000000000100001
10	11101111	00000010	11111011	0000000000100100
11	01111110	11110100	10101101	
12	11111000	10101110	11000101	0000000000101011
13	00000110	10100010	11111011	0000000000110001
14	11101111	00000010	11111011	0000000000110100
15		01010000	11010100	0000000000111011
16			00000100	0000000000111101

- Hogere programmeertalen (PHP, C#, JAVA, PYTHON,...)
- Wij gebruiken PYTHON om u de beginselen van het programmeren aan te leren. Het is de bedoeling dat u in **deze cursus procedureel leert programmeren**, ondanks dat PYTHON ook een object georiënteerde taal is. Als instaptaal in het programmeren is deze taal het eenvoudigst. Wanneer we later Object Georiënteerd zullen programmeren maken we de overstap naar PHP.

## 5. Python installatie

Download en installeer python vanaf de onderstaande link: (zie screenshot)



### 5.1. Verschillen in programmeertalen?

- Programmeertalen verschillen van elkaar:
  - Syntax (de manier van schrijven)
  - Standaard bibliotheken/SDKs/functions
- Java:

```
if (a>b){  
    System.out.println("a is groter dan b");  
}  
else{  
    System.out.println("a is kleiner of gelijk aan b");  
}
```

- Pascal:

```
if a>b then  
    writeln('a is groter dan b');  
else  
    writeln('a is kleiner of gelijk aan b');
```

### 5.2. Structuur van programmeertalen?

- De syntax is gestructureerd
  - *if (...) {...} else {...};*
  - *for (;;;){...}*
  - *while () {...}*
- Basis componenten:
  - *Variabelen*
  - *selectie*
  - *sequentie*
  - *iteratie*
  - *OOP (klassen, methodes en eigenschappen)*

### 5.3. Wat is een variabele?

- Plaatsbepaling voor een veranderende waarde
- We gebruiken variabelen in ons dagdagelijks leven
- Wat is je naam?
  - *Naam = variabele*
- Welke cursus is dit?
  - *Cursus = variabele*

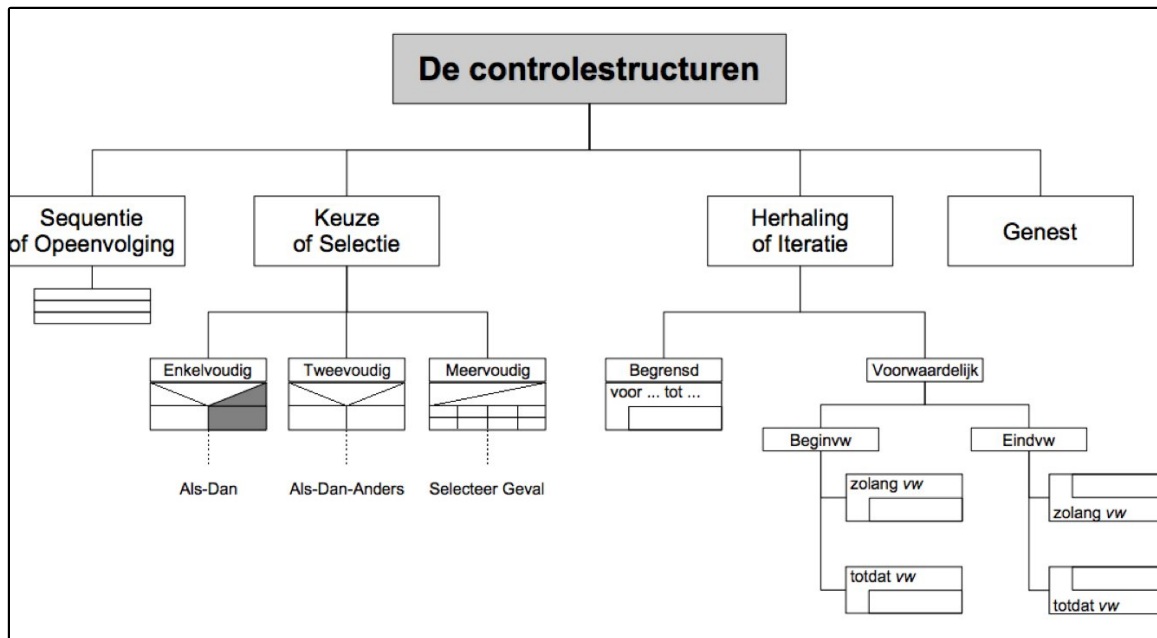
## 6. Jij als full stack developer

- De computer volgt enkel instructies. Het lost zelf geen problemen op. (AI is momenteel nog niet volledig mogelijk)
- De programmeur/webontwikkelaar moet:
  - *Een logische oplossing kunnen ontwerpen.*
  - *Deze logica omzetten in een programmeertaal (implementatie)*
  - *valideren van deze logica en implementatie (testing)*

## 7. Controlestructuren

In alle programmeertalen heeft men 3 basis controlestructuren die men dagdagelijks gebruikt als programmeur, nl:

- sequentie (opeenvolging)
- selectie (keuze)
- iteratie (herhaling)



**Copyright: Tom Vanhoutte**

### 7.1. sequentie

De sequentie is dus een opeenvolging van commando's die worden gegeven via instructies (syntax) eigen aan een programmeertaal.

### 7.2. Selectie

De selectie kan een programma een bepaalde richting uitsturen afhankelijk van een voorwaarde (vraag).


#### 7.2.1. enkelvoudige selectie

NS-diagram	Pseudocode	VBA	PHP
	<pre> <b>Als</b> &lt;voorwaarde&gt; <b>Dan</b>   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; </pre>	<pre> <b>If</b> &lt;voorwaarde&gt; <b>Then</b>   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; <b>End If</b> </pre>	<pre> <b>if</b> (&lt;voorwaarde&gt;) {   &lt;instructie 1&gt;;   &lt;instructie 2&gt;;   ...   &lt;instructie n&gt;; } </pre>

#### 7.2.2. tweevoudige selectie

De tweevoudige keuze			
NS-diagram	Pseudocode	VBA	PHP
	<pre> <b>Als</b> &lt;voorwaarde&gt; <b>Dan</b>   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; <b>Anders</b>   &lt;instructie a&gt;   &lt;instructie b&gt;   ...   &lt;&gt; </pre>	<pre> <b>If</b> &lt;voorwaarde&gt; <b>Then</b>   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; <b>Else</b>   &lt;instructie a&gt;   &lt;instructie b&gt;   ...   &lt;&gt; <b>End If</b> </pre>	<pre> <b>if</b> (&lt;voorwaarde&gt;) {   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; } <b>else</b> {   &lt;instructie a&gt;;   &lt;instructie b&gt;;   ...   &lt;&gt;; } </pre>

### 7.2.3. meervoudige selectie

De tweevoudige keuze			
NS-diagram	Pseudocode	VBA	PHP
	<pre> <b>Als</b> &lt;voorwaarde&gt; <b>Dan</b>   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; <b>Anders</b>   &lt;instructie a&gt;   &lt;instructie b&gt;   ...   &lt;&gt; </pre>	<pre> <b>If</b> &lt;voorwaarde&gt; <b>Then</b>   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; <b>Else</b>   &lt;instructie a&gt;   &lt;instructie b&gt;   ...   &lt;&gt; <b>End If</b> </pre>	<pre> <b>if</b> (&lt;voorwaarde&gt;) {   &lt;instructie 1&gt;   &lt;instructie 2&gt;   ...   &lt;instructie n&gt; } <b>else</b> {   &lt;instructie a&gt;;   &lt;instructie b&gt;;   ...   &lt;&gt;; } </pre>

### 7.3. iteratie

Een iteratie is een herhaling van één of meerdere sequenties of selecties, ook wel een lus genoemd.

#### 7.3.1. Soorten iteraties (herhalingen of lussen)

- Voorbeeld:
  - wanneer je aan een gebruiker honderd getallen dient te vragen, dan dien je geen honderd keer die vraag via een sequentie voor te stellen. Je laat de sequent zich honderd maal herhalen.
- De eerste iteratie die wij zullen overlopen is de For Loop. Daarnaast hebben we nog de while, repeat en endless loop die we later zullen bespreken.

### 7.4. Arrays (Reeksen)

- Een array is de Engelse benaming voor een rij of een reeks.
- In Python heeft een array meerdere benamingen:
  - list = geordende, veranderlijke lijst. Kan dubbele waarden bevatten
  - tuple = geordende, onveranderlijke lijst. Kan dubbele waarden bevatten
  - set = ongeordende en niet-geïndexeerde lijst. Geen dubbele waarden mogelijk
  - dictionary = ongeordende, veranderlijke en geïndexeerde lijst. Geen dubbele waarden mogelijk
- Het is 1 variabele die meerdere elementen kan bevatten van getallen, tekst, ...
- Het eerste element krijgt de index 0, het tweede 1, ...., We beginnen dus vanaf 0 al te tellen.
- Bijvoorbeeld: Een array met als variabele "Namen".
  - `Namen=["Dirk","Luc","Tom"]`
    - `Namen[0]="Dirk"`
    - `Namen[1]="Luc"`
    - `Namen[2]="Tom"`
- De Methodes die een list in **Python** gebruikt zijn:

```
mijnLijst = ["Bloem", "Suiker", "Melk"]
```

- len()** = geeft de lengte van een array terug, m.a.w. het aantal elementen in een lijst.
  - `x = len(mijnLijst)`
  - Resultaat van `x = 3`
- append()** = voegt een nieuw element aan de lijst toe.
  - `mijnLijst.append("Eieren")`
  - Resultaat van `mijnLijst` = ["Bloem", "Suiker", "Melk", "Eieren"]
- pop()** = verwijder een element uit de lijst
  - `mijnLijst.pop(1)`
  - = ["Bloem","Melk"]
- remove()** = verwijder een element uit de lijst met een bepaalde waarde
  - `mijnLijst.remove("Bloem")`
  - Resultaat= ["Suiker", "Melk"]
  - Opgelet: remove wist enkel de eerste waarde die voldoet in de lijst.

## 8. Operatoren

### 8.1. Vergelijkings operatoren

Operator	Beschrijving	Voorbeeld
<code>==</code>	Als de waarden van twee operanden gelijk zijn, wordt de voorwaarde waar.	<code>(a == b)</code> is niet waar.
<code>!=</code>	Als de waarden van twee operanden niet gelijk zijn, wordt de voorwaarde waar.	<code>(a != b)</code> is waar
<code>&gt;</code>	Als de waarde van de linkeroperand groter is dan de waarde van de rechteroperand, wordt de voorwaarde waar.	<code>(a &gt; b)</code> is niet waar.
<code>&lt;</code>	Als de waarde van de linkeroperand kleiner is dan de waarde van de rechteroperand, wordt de voorwaarde waar.	<code>(a &lt; b)</code> is waar.
<code>&gt;=</code>	Als de waarde van de linkeroperand groter is dan of gelijk aan de waarde van de rechteroperand, wordt de voorwaarde waar.	<code>(a &gt;= b)</code> is niet waar.
<code>&lt;=</code>	Als de waarde van de linkeroperand kleiner is dan of gelijk is aan de waarde van de rechteroperand, wordt de voorwaarde waar.	<code>(a &lt;= b)</code> is waar.

### 8.2. Meerdere vergelijkings operatoren (ook chaining operators genoemd)

**`1 < 2 < 3`**  
**Waar**

De bovenstaande statement controleert of 1 minder was dan 2 en of 2 minder dan 3 was. We hadden dit kunnen schrijven met een AND statement in Python:

**`1<2 and 2<3`**  
**Waar**

De AND wordt gebruikt om ervoor te zorgen dat twee controles waar moeten zijn en ervoor te zorgen dat de totale controle waar is. Laten we nog een voorbeeld bekijken:

**`1 < 3 > 2`**  
**Waar**

Het bovenstaande controleert of 3 groter is dan beide andere nummers, dus je zou het kunnen gebruiken en herschrijven als:

**`1<3 and 3>2`**  
**Waar**

Het is belangrijk op te merken dat Python beide instanties van de vergelijkingen controleert. We kunnen ook vergelijkingen in Python gebruiken of schrijven. Bijvoorbeeld:

**`1==2 or 2<3`**  
**Waar**

Merk op hoe het waar was; dit komt omdat we met de operator of alleen de ene of de andere nodig hebben om waar te zijn. Laten we nog een voorbeeld bekijken:

**`1==1 or 100==1`**  
**Waar**

## 9. Datatypes in Python

Naam	Type	Beschrijving
integers	<b>int</b>	Nummers: 5 33 555 3333
floating point	<b>float</b>	Nummers met een decimal point: 4.5 8.0 30020.9
strings	<b>str</b>	Tekst: "hallo wereld" "Tom" "1234"
lists	<b>list</b>	Geordende sequentie van objecten [10, "hallo", 200.3]
Dictionaries	<b>dict</b>	Sleutels en waarden {"sleutel": "waarde", "sleutel": "waarde"}
Tuples	<b>tup</b>	Geordende mix van datatype (22, "ok", 345.22)
Sets	<b>set</b>	Niet geordende collectie van UNIEKE objecten {"a", "b"}
Booleans	<b>bool</b>	2 waarden: waar of onwaar

TIP: Wens je een datatype in je programma te achterhalen dan kan je het gereserveerde woord **type** gebruiken.

Voorbeeld

```
a = 500
```

```
type(a)
```

```
Resultaat = int
```

### 9.1. Strings

Strings of tekenreeksen kan je tussen enkele of dubbele quotes gebruiken.

```
'developers'
```

```
"developers"
```

```
"smorgens"
```

Voorkeur = dubbele quotes!

#### 9.1.1. Index en manipulatie van strings

In strings spreken we ook van index positions. Ieder karakter heeft een index.

Voorbeeld:

- Character : Z O M E R
- Index : 0 1 2 3 4
- -Index : -5 -4 -3 -2 -1

Voorbeeld:

- woord = "zomer"
- woord[2] geeft als resultaat de letter **M**
- woord[2:] geeft als resultaat **MER**. Het dubbele punt zorgt ervoor dat de rest van de string wordt weergegeven.
- woord[:2] geeft als resultaat **ZO** terug. Het dubbele punt zorgt ervoor dat 2 karakters vanaf het begin van de string worden weergegeven.
- woord[1:3] geeft als resultaat **OM** terug.
- woord[:] = toont de volledige string **ZOMER**.
- woord[::2] = stappen per 2. Resultaat **ZMR**. Telkens een karakter wordt overgeslagen.
- woord[start:stop:step]. Je kan kiezen op welk karakter je start en welke positie je stopt gecombineerd met eventueel een stap.
- woord[::-1] = REMOZ = de volledige string omgedraaid.

Voorbeeld:

Onder string concatenation verstaan we het samenvoegen van strings.



Dit voeren we uit met het **+** teken

- **woord** = "zomer"
- **"De " + woord** geeft als resultaat **De zomer**
- letter = 'z'
- **letter \* 4** geeft als resultaat = **'zzzz'**

Voorbeeld: **upper,lower** en **split** methodes:

Methodes of functions kunnen zelf worden geschreven in Python. Maar Python heeft ook enkele methodes of functies die we reeds kunnen aanwenden bij het gebruik van strings.

- woord= "Hallo Wereld"
- woord.**upper()** geeft als resultaat **HALLO WERELD**
- woord.**lower()** geeft als resultaat **hallo wereld**
- woord.**split()** geeft als resultaat **['Hallo','wereld']**

Voorbeeld: **format** methode

- print('Dit is de tekenreeks {}'.format('UITLEG'))
- Resultaat = **Dit is de tekenreeks UITLEG**. De accolades zorgen voor het invoegen van een variabele waarde.
- print('Dit is de {} {}'.format('cursus','full stack developer'))
- Resultaat = **Dit is de cursus full stack developer**. Meerdere accolades worden sequentieel ingevuld.
- print ('Dit is de {1} {0}'.format('cursus','full stack developer'))
- Resultaat = **Dit is de full stack developer cursus**. Je kan de index zelf bepalen bij het weergeven van de volgorde in een string.
- print('Dit is de {a} {b}'.format(a='cursus', b='full stack developer'))
- Resultaat = **Dit is de cursus full stack developer**. Je kan ook gebruik maken van variabelen.

### 9.1.2. Concatenation

Onder string concatenation verstaan we het samenvoegen van strings.

Dit voeren we uit met het **+** teken

Voorbeeld:

- **woord** = "zomer"
- **"De " + woord** geeft als resultaat **De zomer**
- 

### 9.1.3. Speciale karakters

**\n** = **backslash n** = nieuwe lijn

Voorbeeld: print('Hallo \n Wereld')

- Resultaat: 2 lijnen output
- Hallo
- Wereld

**\t** = **tab** = zorgt voor een tabulator spatie tussen 2 woorden

## 10. FUNCTIES

Functies zijn algoritmes met herbruikbare code. Als voorbeeld schrijven een functie die de som van 2 getallen optelt. Deze functie gebruiken we dan in de code van ons programma zelf. Een functie definiëren doe je met het gereserveerde woord **def** met **daarna de naam van de functie** zelf.

Een functie kan al dan niet één of meerdere parameters bevatten. In ons onderstaand voorbeeld zijn dit respectievelijk de parameters met de namen **a** en **b**.

Het gereserveerde woord **return** zorgt voor de overdracht van het resultaat naar de programmacode zelf.

Schrijfwijze van de functie:

```
def som(a,b):  
    return a+b
```

Gebruik van de functie in je programma:

je vraagt eerst 2 getallen aan de gebruiker. Vervolgens gebruik je de functie om het resultaat op te halen.

```
getal1 = int(input("Geef getal 1 in:"))  
getal2 = int(input("Geef getal 2 in:"))
```

#Hieronder gebruik je de som functie om uiteindelijk het resultaat in de variabele mijnResultaat te steken en deze af te printen op het scherm. Getal1 en Getal2 worden dus overgedragen naar de functie en worden dus in principe a en b!

```
mijnResultaat = som(getal1,getal2)  
print(mijnResultaat)
```

**OPMERKING:** HET IS OOK MOGELIJK OM ALLE FUNCTIES IN EEN APART PYTHON BESTAND TE BEWAREN. VERVOLGENS KAN JE DEZE DAN MET DE IMPORT FUNCTIE GEBRUIKEN IN GELIJK WELK ANDER PYTHON BESTAND. DEZE 2 BESTANDEN DIENEN WEL BIJ ELKAAR TE STAAN OM DIT TE DOEN WERKEN.

DE IMPORT CODE IS DE VOLGENDE:

```
from functiebestand.py import * => hiermee lees je alle functies die zich in dit bestand zouden bevinden.  
from functiebestand.py import som => hiermee lees je enkel de som functie uit het bestand.
```

## 11. Gezamenlijke oefeningen : oplossingen

### 11.1. Oefeningen op de sequentie

#### 11.1.1. Oef 01

- Vraag 2 getallen op
- Geef de som van de getallen weer op het scherm

##### PYTHON

```
# MODULE: BASIS PROGRAMMEREN
# GEZAMENLIJKE OEFENING 1
# DOCENT: TOM VANHOUTTE
```

```
eersteGetal = int(input("Geef een eerste getal in: "))
tweedeGetal = int(input("Geef een tweede getal in: "))
```

```
som = eersteGetal + tweedeGetal
```

```
print(som)
print("De som is:",som)
print(eersteGetal,"+",tweedeGetal,"=",som)
```

### 11.2. Oefeningen op de selectie

#### 11.2.1. Oef 02 (enkelvoudige selectie)

- Ontwerp een toepassing waarmee je de leeftijd van de gebruiker opvraagt. Als de ingevoerde leeftijd 16 jaar of meer bedraagt, krijgt de gebruiker toegang en verschijnt de tekst: "Je bent x jaar oud. Veel plezier in Tomorrowland."

##### PYTHON

```
# MODULE: BASIS PROGRAMMEREN
# OEFENINGEN OP DE SELECTIE
# ENKELVOUDIGE SELECTIE
# GEZAMENLIJKE OEFENING 2
# DOCENT: TOM VANHOUTTE
```

```
leeftijdGebruiker = int(input("Geef uw leeftijd in: "))
```

```
if(leeftijdGebruiker >= 16):
    print("U bent",leeftijdGebruiker,"jaar oud. Veel plezier in
Tomorrowland.")
```

### 11.2.2. Oef 03

- Om toegang te krijgen tot Tomorrowland moet je minimum 16 jaar oud zijn. Ontwerp een toepassing waarmee je de leeftijd van de gebruiker opvraagt. Als de ingevoerde leeftijd 16 jaar of meer bedraagt, krijgt de gebruiker toegang en verschijnt de tekst: "Je bent x jaar oud. Veel plezier in Tomorrowland.". Indien niet voldaan is aan deze voorwaarde verschijnt de tekst: "Helaas! Je moet minimum 16 jaar zijn om toegang te krijgen tot Tomorrowland.".

#### PYTHON

```
# MODULE: BASIS PROGRAMMEREN
# OEFENINGEN OP DE SELECTIE
# ENKELVOUDIGE SELECTIE
# GEZAMENLIJKE OEFENING 3
# DOCENT: TOM VANHOUTTE
```

```
leeftijdGebruiker = int(input("Geef uw leeftijd in: "))

if(leeftijdGebruiker >= 16):
    print("U bent",leeftijdGebruiker,"jaar oud. Veel plezier in
Tomorrowland.")
else:
    print("Helaas! Je moet minimum 16 jaar oud zijn om toegang te
krijgen tot Tomorrowland.")
```

Copyright: Tom Vanhoutte

### 11.2.3. Oef 04 (tweevoudige selectie)

- Indien de gebruiker 21 jaar of ouder is, krijgt hij/zij toegang tot de VIP-lounge van Tomorrowland. De volgende tekst verschijnt: "Je krijgt toegang tot onze exclusieve VIP-lounge!".

#### PYTHON

```
# MODULE: BASIS PROGRAMMEREN
# OEFENINGEN OP DE SELECTIE
# TWEEVoudIGE SELECTIE
# GEZAMENLIJKE OEFENING 4
# DOCENT: TOM VANHOUTTE
```

```
leeftijdGebruiker = int(input("Geef uw leeftijd in: "))

if(leeftijdGebruiker >= 21):
    print("U bent",leeftijdGebruiker,"jaar oud. Veel plezier in de
VIP lounge van Tomorrowland.")
elif(leeftijdGebruiker >= 16):
    print("U bent",leeftijdGebruiker,"jaar oud. Veel plezier in
Tomorrowland.")
else:
    print("Helaas je bent",leeftijdGebruiker,"jaar oud. Je moet 16
jaar zijn om toegang te krijgen tot Tomorrowland.")
```

#### 11.2.4. Oef 05 (meervoudige selectie)

- Groepen krijgen groepskorting op Tomorrowland. De kortingen zijn als volgt:
  - een groep van 3 personen krijgt 10% korting;
  - een groep van 4 personen 20%;
  - een groep van 5 personen 50 %;
  - groepen vanaf 6 personen mogen gratis binnen.

#### PYTHON

```
# MODULE: BASIS PROGRAMMEREN
# OEFENINGEN OP DE SELECTIE
# MEERVOUDIGE SELECTIE
# GEZAMENLIJKE OEFENING 5
# DOCENT: TOM VANHOUTTE
```

```
AantalPersonen = int(input("Hoeveel personen: "))
```

```
if(AantalPersonen >= 6):
    print("U mag gratis binnen op Tomorrowland")
elif(AantalPersonen == 3):
    print("U krijgt 10% korting")
elif(AantalPersonen == 4):
    print("U krijgt 20% korting")
elif(AantalPersonen == 5):
    print("U krijgt 50% korting")
else:
    print("Helaas je komt niet in aanmerking voor korting")
```

### 11.3. Oefeningen op de iteratie

#### 11.3.1. Oef 06 (For loop)

- Schrijf een programma die aan de gebruiker een eindgetal vraagt.
- Weergave op het scherm: van 1 tot en met het eindgetal
- Voorbeeld:
  - 1
  - 2
  - 3
  - *eindgetal*

#### PYTHON

```
# MODULE: BASIS PROGRAMMEREN  
# OEFENINGEN OP DE ITERATIE  
# FOR  
# GEZAMENLIJKE OEFENING 6  
# DOCENT: TOM VANHOUTTE
```

```
eindGetal = int(input("Geef uw eindgetal in: "))
```

```
for x in range(eindGetal):  
    print(x)
```

```
for x in range(1, eindGetal+1):  
    print(x)
```

```
for x in range(1, 50, 3):  
    print(x)
```

### 11.3.2. Oef 07 (While loop)

- Schrijf een programma die een aantal getallen inleest en eindigt wanneer de gebruiker een negatief getal ingeeft.
- Schrijf daarna de som van de getallen op het scherm.

#### PYTHON

```
# MODULE: BASIS PROGRAMMEREN
# OEFENINGEN OP DE ITERATIE
# WHILE
# GEZAMENLIJKE OEFENING 7
# DOCENT: TOM VANHOUTTE

som = 0
checkGetal = int(input("Geef een getal in: "))
while checkGetal >= 0:
    som += checkGetal
    checkGetal = int(input("Geef een getal in: "))
    if checkGetal < 0:
        print("De som is:", som, "")
```

### 11.4. Oef 08 (arrays of reeksen)

- Lees 3 getallen in en steek ze in een array.
- Lees de 3 getallen vervolgens weer uit de array en schrijf de som van deze 3 getallen op het scherm

#### PYTHON

```
# MODULE: BASIS PROGRAMMEREN
# OEFENINGEN OP ARRAYS
# GEZAMENLIJKE OEFENING 8
# DOCENT: TOM VANHOUTTE

# ARRAY VULLEN MET GETALLEN

aantalGetallen = int(input("Hoeveel getallen wenst u in te geven: "))
reeksGetallen = []

for x in range(0, aantalGetallen):
    reeksGetallen.append(int(input("Geef een getal in: ")))

print(reeksGetallen)

# UITLEZEN VAN DE ARRAY EN DE SOM VAN DE GETALLEN WEERGEVEN
som = 0
for x in range(0, aantalGetallen):
    som = som + reeksGetallen[x]

print("De som van de getallen is: ", som, "")
```

### 11.5. Oef 09 (functies SCHRIJVEN)

- Schrijf een functie die de factuliteit van een getal weergeeft.

```
# MODULE: BASIS PROGRAMMEREN
# OEFENINGEN OP FUNCTIES
# GEZAMENLIJKE OEFENING 9
# DOCENT: TOM VANHOUTTE

# FUNCTIE SCHRIJVEN: FACULTEIT VAN EEN GETAL

def faculteitGetal(x):
    if x == 0:
        return 1
    return x * fact(x - 1)

x=int(input())
print fact(x)
```

# Copyright: Tom Vanhoutte