



## IT Essentials

# Hoofdstuk 1

## Expressies

### **DE HOGESCHOOL MET HET NETWERK**

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](https://www.pxl.be/facebook)



# Inhoud

- 0. Inleiding
- 1. Resultaten tonen: `print()`
- 2. Datatypes: string, integer en float
- 3. Expressies
  - 1. Eenvoudige berekeningen
  - 2. Complexe berekeningen
  - 3. String expressies
  - 4. Type casting: `int()`, `float()` en `str()`



# 1.0 Inleiding

Wat is een **algoritme**?

- is een reeks opdrachten die vanuit een gegeven begintoestand naar een beoogd doel leiden.
- Geef een voorbeeld uit het dagelijks leven.



# Wat is **programmeren**?

- Een computer vertellen wat hij moet doen.
- Een computerprogramma is een opeenvolging van instructies (= opdrachten) die de computer verstaat.

*In de cursus IT Essentials gaan we leren algoritmes bedenken. Die algoritmes gaan we leren omzetten in instructies (een programma) m.b.v. de programmeertaal Python.*



# 1.1 Resultaten tonen

## In de shell:

2 + 14 → 16

```
C:\>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2 + 14
16
>>> 6 + 9
15
>>> exit()
C:\>
```




## In een programma:

- `print(2 + 14)` ➔ 16
- `print` is een functie. Heeft functiehaken!
- `print("Welkom in het vak ITEssentials!")`



# 1.2 Datatypes

## 1.2.1 Strings

- `print("Welkom in het vak ITEssentials!")`  
`= print('Welkom in het vak ITEssentials!')`
- `print("Hallo,"  "dag",  "Jan")` → Hallo, dag Jan
- `print("dia's")` → dia's = goed  
`print('dia's') ` = fout. Waarom?
- `print('Jan zegt: "Hallo"')` → Jan zegt: "Hallo" = goed  
`print("Jan zegt: "Hallo"") ` = fout. Waarom?

# Bestudeer waarom onderstaande voorbeelden correct zijn:

Als je wilt afdrukken?

- Jan zegt: "Waar zijn de dia's?"

```
print("Jan zegt: \"Waar zijn de dia's?\"")  
print('Jan zegt: "Waar zijn de dia\'s?"')
```

- \: Dit is een **backslash**




```
print("\: Dit is een backslash")  
print('\: Dit is een backslash')
```

- \"Dit is een **backslash**\"

```
print("\\\"Dit is een backslash\\\"")  
print(\"\\\"Dit is een backslash\"")
```



## 1.2.2 Integers (= gehele getallen)

- `print(3)` → 3
- `print(+3)` → 3
-  `print("3")` → ?
-  `print("+3")` → ?
- `print(-16)` → -16
- `print(2, 5, 7)` → 2 5 7
- `print(20000000000)` → 20000000000
- `print(2,000,000,000)` → ?
-  `!!! print(2.000.000.000)` → ?

## 1.2.3 Floats (= floating-point getallen = getallen met decimalen)

- `print(3.14159265)` → 3.14159265  
↔ `print(3,14159265)` → ?
- `print(3.0)` → 3.0 (is een float)  
↔ `print(3)` → 3 (is een integer)
- `print( (431 / 100) * 100 )` → 430.999999999999994  
i.p.v. 431 (door de manier waarop  
Python floats opslaat )

# 1.3 Expressies

= combinatie van één of meerdere waarden (strings, integers of floats) met behulp van operatoren, die dan een nieuwe waarde oplevert.



## 1.3.1 Eenvoudige berekeningen

+	optelling
-	aftrekking
*	vermenigvuldiging
/	deling
//	integer deling
**	machtsverheffing
%	modulo

Als er een deling in de berekening zit, is het resultaat een float!!!

Als er een float in de berekening zit, is het resultaat een float!!!

- `print(5 + 3)` → 8
- `print(5 - 3)` → 2
- `print(5 * 3)` → 15
- `print(5 / 3)` → 1.6666666666666667
- `print(5 // 3)` → 1 (rondt af naar beneden)!
- `print(6 / 3)` → 2.0
- `print(6.0 / 2)` → 3.0
- `print(5.0 // 3)` → 1.0
- `print(5 ** 3)` → 125
- `print(5 % 3)` → 2 = rest na deling



## 1.3.2 Complexe berekeningen

		prioriteit
()	haakjes	1
**	machtsverheffing	2
*	vermenigvuldiging	3
/	deling	3
//	integer deling	3
%	modulo	3
+	optelling	4
-	afrekking	4

Bewerkingen met dezelfde prioriteit worden van links naar rechts uitgevoerd

Als er een float in de berekening zit, is het resultaat een float!!!

- `print(5 * 2 - 3 + 4 / 2)` → 9.0 i.p.v. 9!
- `print(6 - 10 / 2 + 9)` → ?
- `print((6 - 10) / 2 + 9)` → ?
- `print(15 / 6 * 2)` → ?
- `print(10 - 6 % 4 / 2)` → ?

## Opgave 1.1

Schrijf een programma dat het aantal seconden in 2 weken berekent. Je moet de berekening doen in Python code.





### 1.3.3 String expressies

Je kunt de plus (+) gebruiken om twee of meerdere strings aan elkaar te "plakken".

- `print("tot" + "ziens")` → totziens
- `print("tot " + "ziens")` → tot ziens
- `print("tot" + " ziens")` → tot ziens
- `print("tot" + " " + "ziens")` → tot ziens



## 1.3.4 Type casting

- **int()** geeft de waarde tussen haakjes terug als integer (= afkappen)
- **float()** geeft de waarde tussen haakjes terug als float
- **str()** geeft de waarde tussen haakjes terug als string.



- `print(15 / 4)` → 3.75
- `print(int(15 / 4))` → 3
- `print(15 + 4)` → 19
- `print(float(15 + 4))` → 19.0
- `print(float(-17))` → ?
- `print("Ik wil minimum een " + str(10) + " halen.")` → ?
- `print("Ik wil minimum een " + 10 + " halen.")` → ?

## Opgave 1.2

Een boek kost in de winkel €24,95, maar boekwinkels krijgen 40 procent korting bij inkoop. Het verscheppen van boeken kost €3 voor het eerste boek, en 75 cent voor ieder volgende boek. Bereken hoeveel de winkel betaalt voor 60 boeken.

