

## OEFENINGEN PYTHON – REEKS 1

### Vraag 1: Python als een eenvoudige rekenmachine

Python maakt gebruik van enkele vaak voorkomende (data)types. Zo zijn er **integers** die behoren tot de gehele getallen (VB: 3) en zijn er **float's** die behoren tot de reële getallen; weliswaar met beperkte nauwkeurigheid (VB: 3.1415). Voor reële getallen gebruikt men een PUNT in plaats van een komma.

Python respecteert net als andere programmeertalen de volgorde van bewerkingen zoals in de wiskunde: eerst de zaken binnen de haakjes, vervolgens machten, dan product en deling en vervolgens som en aftrekking. Verder bestaan er ook speciale operatoren zoals bijvoorbeeld de modulo (%) operator. Enig idee wat deze operator zou doen?

- a) Bekijk de volgende expressies. Voorspel wat het resultaat zal zijn als je ze ingeeft in Python. Denk goed na over de waarde die de uitkomst zal krijgen. Voer daarna de bewerking uit binnen Python.

Expressie	Voorspelde uitkomst	Python uitkomst
$1 + 2 * 3$		
$(1 + 2) * 3$		
$2 ** 4$		
$13 \% 5$		
$7 / 2$		
$-7 / 2$		
$7.0 / 2$		
$7 / 2.0$		
$7 // 2$		
$9 \% -2$		

- b) We hebben reeds kennis gemaakt met twee datatypes binnen Python: integers (gehele getallen) en floating point getallen (reële getallen). Men kan tussen deze datatypes wisselen door gebruik te maken van de ingebouwde functies **int(x)** en **float(x)**. Enkele andere ingebouwde functies zijn **round(x)** om af te ronden en **abs(x)** voor de absolute waarde. Voorspel wat het resultaat van de volgende expressies zal zijn als je ze ingeeft in Python. Denk goed na over de waarde die de uitkomst zal krijgen. Noteer eerst de uitkomst die je verwacht op dit blad, voer daarna de bewerking uit in de editor.

Expressie	Voorspelde uitkomst	Python uitkomst
<code>int(5)</code>		
<code>int(3.0)</code>		
<code>int(5.99)</code>		
<code>int(-12.84)</code>		
<code>float(13)</code>		
<code>float(15.2)</code>		
<code>round(0.5)</code>		
<code>round(1.5)</code>		
<code>round(2.5)</code>		

round(3.5)		
round(-0.5)		
round(-1.5)		
round(-2.5)		
round(-3.5)		
abs(-2)		
abs(-3.1415)		

c) Zoek de oplossing van de volgende vraagstukken door middel van de berekeningen in te geven in Python. Let erop dat je de juiste types gebruikt!

1. Het volume van een bol met straal  $r$  is  $\frac{4}{3}\pi r^3$ . Wat is het volume in  $\text{cm}^3$  van een bol met straal 5 cm als  $\pi = 3.14159$ ?
2. Tom heeft deelgenomen aan een loopwedstrijd over 6 mijl in New York. Hij legde de afstand af in 53 minuten en 30 seconden. Aan welke snelheid, uitgedrukt in km/u, liep Tom gemiddeld als je weet dat één mijl gelijk is aan 1.61 kilometer?

Opmerking: Er worden 2 verschillende versies van Python. Python 3 is de recentste en heeft enkele belangrijke wijzigingen waardoor compatibiliteit niet gegarandeerd is. Veel externe bibliotheken (voorgeprogrammeerde functies) zijn beschikbaar bij Python 2 maar nog niet altijd bij Python 3. Daarom wordt Python 2 nog veelvuldig gebruikt voor complexe projecten. Python 2 code werkt daarom niet altijd of foutief binnen Python 3 en vice versa. De meest opmerkbare verschillen zitten in de andere werking van deling tussen 2 integers (standaard een float deling bij Python 3, integer deling bij Python 2 waardoor verlies van precisie kan optreden) en bij de print functie.

## Vraag 2: Variabelen

Variabelen zijn namen voor waarden die zich bevinden in het geheugen. Een waarde wordt toegewezen (=assignment) aan een variabele door een = teken (VB: *degrees\_celcius* = 26). Deze variabele *degrees\_celcius* kan nu gebruikt worden in een expressie en zal omgezet worden naar de waarde 26.

- a) Maak twee nieuwe variabelen aan met naam *pi* en *r*, dewelke je de respectievelijke waarden 3.14159 en 5 toekent. Bereken nu opnieuw het antwoord op vraag 1c punt 1 waarbij je de nieuwe variabelen gebruikt voor je berekening.
- b) Een variabele heeft geen vaste waarde, daarom noemen we het ook “variabele”. Na een eerste “assignment” kan je de waarde van de variabele overschrijven door er een andere waarde aan toe te kennen, t.t.z. een nieuwe assignment te doen. Maak twee nieuwe variabelen: *x* met waarde 4 en *y* met waarde 5. Bekijk nu de drie expressies in de tabel hieronder. Schrijf eerst op dit blad op wat je denkt dat Python zal antwoorden als je de expressies één voor één ingeeft. Nadien doe je dit effectief binnen Python.

Expressie	Voorspeld antwoord	Python antwoord
-----------	--------------------	-----------------

x = 4		
y = 5		
x = x + y		
x		
y		

### Vraag 3: Strings

Een **string** is een type bestaande uit een opeenvolging van karakters. De waarde van een string bevindt zich steeds tussen ' of " aanhalingstekens.

- Definieer twee string-variabelen: *naam* en *voornaam*. Je kent deze variabelen je eigen achternaam en voornaam als waarde toe. Maak nu een derde variabele aan en probeer hierin je volledige naam op te slaan zonder dat je die moet opnieuw typen. Let op: tussen je voornaam en je achternaam staat een spatie!
- Creëer de string 'bla bla bla bla bla ' door slechts 1x het woordje 'bla' te typen. Maak gebruik van de '\*' operator.

### Vraag 4: Booleaanse logica

Het **boolean** type heeft **True** en **False** als mogelijke waarden. Let hierbij op dat de eerste letter een hoofdletter is en dat er geen leestekens voorkomen; anders is het een string.

Een **boolean** is een type dat enkel de waarden **True** en **False** (zonder aanhalingstekens en met hoofdletter) kan aannemen. De **booleaanse operatoren** bestaan uit **and**, **or**, en **not**. De **not** operator keert de waarde om: True wordt False en vice versa. Bij **and** moeten alle waarden waar zijn opdat het antwoord waar zou zijn; bij **or** moet slechts één waarde waar zijn.

**Relationele operatoren:** groter dan > of gelijk >= , kleiner dan < of gelijk <= , is gelijk aan == (dubbel anders variabele toekenning) en is niet gelijk aan != geven steeds een boolean terug.

- Voorspel wat het resultaat van de volgende expressies zal zijn als je ze ingeeft in Python. Noteer eerst de uitkomst die je verwacht op dit blad, voer daarna de bewerking uit in de editor.

Expressie	Voorspelde uitkomst	Python uitkomst
3 > 5		
2 <= 2		
10.0 == 10		
5 != 8		
"abc" < "bcd"		
not False		
True and False		
True or False		
False or not False		
((not True) or False) and (not ((not False) or True))		
not True or False and not not False or True		

We gaan nu niet meer in de shell werken maar alle code schrijven in een nieuw Python document. Maak eerst een nieuw document aan in Wing (Ctrl + N). Dit document sla je vervolgens op als een bestand met extensie 'py' (noem het bijvoorbeeld 'reeks1vraag5.py'). Sla dit bestand bij voorkeur op in een nieuwe map 'python' op de **Z:** (netwerk)schijf. Het programma kan je uitvoeren via 'Run'. Let op: dit zal alle code in het werkschrift evalueren, maar niet automatisch het resultaat weergeven zoals wel gebeurde toen je in de shell werkte. Je zal dus telkens de '**print(...)**' functie moeten gebruiken als je een waarde wilt zien verschijnen in de shell. Voer de berekeningen voor vraag 1c nu opnieuw uit maar door gebruik te maken van een uitgevoerd python bestand.

## Vraag 5: Keuzes maken

De mogelijkheid om keuzes te maken afhankelijk van de resultaten van een bepaalde conditie is erg belangrijk. Men maakt hierbij gebruik van booleaanse logica (zie hierboven) om bepaalde stukken code wel of niet uit te voeren.

```
if (X >= 10):
    print("X is groter dan of gelijk aan 10!")
elif (X > 5):
    print("X is kleiner dan 10 maar groter dan 5!")
elif (X > 0):
    print("X is kleiner dan of gelijk aan 5 maar groter dan 0!")
else:
    print("X voldoet aan geen enkele van bovenstaande condities.")
```

- a) Maak code die print of het tijd is voor pauze of niet, afhankelijk of de les al 2u bezig is. Ken op voorhand de waarden toe aan de gebruikte variabele(n).

## Vraag 6: Een eerste functie

Functies in programmeertalen zijn vergelijkbaar met functies in wiskunde. Beide hebben één of meerdere argumenten waarop bewerkingen worden uitgevoerd om tot een resultaat te komen. VB in wiskunde:  $z = f(x, y) = \sqrt{x^2 + y^2}$  waarbij x en y het argument zijn en z het resultaat. In Python zou de functie als volgt zijn:

```
def pythagoras(x, y):
    resultaat = (x**2 + y**2)**0.5
    return resultaat
```

De vaste structuur van een functie bestaat steeds uit **def functienaam(argumenten)**: waarna er wordt geïndenteerd (ingesprongen). Een **return** wordt gebruikt indien de functie een waarde (resultaat) moet teruggeven. Over het algemeen valt bij functies return sterk te verkiezen boven print.

Functies schrijven we standaard bovenaan in het python document. Dit hoeft echter niet, maar onthoud dat je een functie pas kunt aanroepen in code die ONDER de functie-definitie in het document staat. Functies en variabelen die gedefinieerd zijn in een ander werkschrift of bibliotheek (VB math bibliotheek) kan je beschikbaar maken door middel van de IMPORT functie. Ingebouwde Python functies zoals print() zijn altijd beschikbaar.

- a) Definieer in dit werkschrift een functie die een ingegeven temperatuur omzet van Celsius naar Fahrenheit:  $^{\circ}\text{F} = ^{\circ}\text{C} * \frac{9}{5} + 32$ . Test deze functie door ze op te roepen in het document

onder de functie-definitie. Nadat je het programma hebt uitgevoerd (via 'run') kan je de functie ook oproepen in de shell. Probeer dit eens!

- b) Schrijf een functie die teruggeeft of een bepaald jaartal een schrikkeljaar is of niet. Een schrikkeljaar is deelbaar door 4, maar niet deelbaar door 100. Indien het jaar deelbaar is door 400 dan is het weer wel een schrikkeljaar. Zorg ervoor dat het resultaat een boolean is.
- c) Los bovenstaande vraag opnieuw op door enkel gebruik te maak van de booleanse AND, OR, NOT operatoren. Maak dus geen gebruik van if/elif/else structuren.
- d) Schrijf een functie waaraan je twee jaartallen kunt meegeven. De functie geeft aan of er minstens één van de twee jaren een schrikkeljaar is. Tip: je kunt de functie die je in vraag 6b) hebt gemaakt gebruiken door hiernaar een functieaanroep te doen, zo kun je bestaande functies hergebruiken zonder alles te moeten copy-pasten met alle mogelijke problemen van dien.

## Vraag 7: Lijsten en Tupels

De types **list** en **tuple** zijn types die meerdere objecten van om het even welk type in een bepaalde orde kunnen bevatten. Deze objecten worden elementen genoemd en kunnen aangesproken door te refereren naar een index, gaande van 0 tot de lengte – 1 (VB: mylist[1] geeft het 2<sup>e</sup> element terug). Een list is muteerbaar (inhoud veranderbaar) en gebruikt vierkante haakjes (VB: [1,2,3] ) terwijl een tuple niet muteerbaar is en ronde haakjes gebruikt (VB: ('a', 'b', 'c') ).

- a) Definieer een variabele 'mylist' en sla hierin een lijst op die de eerste 5 letters van het alfabet bevat.
- b) Gebruik de ingebouwde **len()** functie om het aantal elementen in de lijst terug te krijgen. Print enkel het laatste element in de lijst af door gebruik te maken van de bijhorende index.
- c) Wijzig het derde element uit 'mylist' door de letter 'z' en print de lijst af ter controle
- d) Maak een nieuwe variabele 'mylist2' en sla hierin een lijst op die twee strings en een integer bevat, namelijk jouw voornaam, jouw achternaam en jouw leeftijd. Print deze variabele af. Voeg nu de twee lijsten samen en sla het resultaat op in een nieuwe variabele 'mylist3'. Print 'mylist3' af ter controle.
- e) Idem aan vraag 5a), maar nu maak je een tuple in plaats van een lijst (gebruik hiervoor ronde haken ipv vierkante). Wat gebeurt er als je nu net zoals in vraag 5b) de waarde van het derde element wilt wijzigen?

## Vraag 8: Methodes

Sommige types die in Python worden gebruikt bieden de programmeur methodes aan. Deze kan je oproepen door middel van een '.' achter een variabele van het desbetreffende type.

- a) Definieer eerst een nieuwe lijst: `mylist = [100,200,400,500]`. Nu willen we op de derde plaats in de lijst de waarde '300' toevoegen. Probeer dit te verwezenlijken door gebruik te maken van de methode `insert` die beschikbaar is in het lijst-type (arg: index, waarde). Daarna voeg je aan de lijst de waarde 1000 toe door middel van de methode `append` (arg: waarde). Ten slotte print je de gesorteerde lijst af met de methode `sort` (geen arg).
- b) Sla de volgende zin op in een nieuwe string-variabele:

*“Zeven Zottegemse zotten zullen zes zomerse zondagen zwemmen zonder zwembroek.”*

- a. Tel het aantal woorden in de zin. Hint: een string heeft een methode `split` die een lijst teruggeeft met woorden gescheiden door spaties. `len(mylist)` geeft je de lengte van de lijst
- b. Vervang voor elk woord de eerste letter door een 'p'. Hint: een string heeft een methode `replace(old,new)` die een substring `old` vervangt door de substring `new`. Let op: deze methode is hoofdlettergevoelig!