

Korte introductie

in Python

t.b.v. matchingdag

HBO-ICT

Inhoud

[1. Inleiding 3](#_Toc508024875)

[2. Algebraïsche expressies 3](#_Toc508024876)

[3. Nog meer algebraïsche expressies 4](#_Toc508024877)

[4. Booleaanse expressies 5](#_Toc508024878)

[5. Booleaanse operatoren 6](#_Toc508024879)

[6. Toekenningen en variabelen 6](#_Toc508024880)

[7. Strings 7](#_Toc508024881)

[8. Soorten getallen 8](#_Toc508024882)

[9. Het gebruik van de index bij strings 8](#_Toc508024883)

[10. Python standaard bibliotheken 9](#_Toc508024884)

[11. Lists 10](#_Toc508024885)

[12. List-operatoren 10](#_Toc508024886)

[13. List-functies 11](#_Toc508024887)

[14. List-methoden: syntax 11](#_Toc508024888)

[15. Veel gebruikte list-methoden 11](#_Toc508024889)

[16. Opdrachten 13](#_Toc508024890)

[Opdracht 1\_1 13](#_Toc508024891)

[Opdracht 1\_2 13](#_Toc508024892)

[Opdracht 1\_3 13](#_Toc508024893)

[Opdracht 1\_4 14](#_Toc508024894)

[Opdracht 1\_5 14](#_Toc508024895)

[Opdracht 1\_6 14](#_Toc508024896)

[Opdracht 1\_7 14](#_Toc508024897)

[Opdracht 1\_8 (lastig!) 14](#_Toc508024898)

[Opdracht 1\_9 15](#_Toc508024899)

[Opdracht 1\_10 15](#_Toc508024900)

[Opdracht 1\_11 15](#_Toc508024901)

[Opdracht 1\_12 16](#_Toc508024902)

[Opdracht 1\_13 16](#_Toc508024903)

[Opdracht 1\_14 16](#_Toc508024904)

[Opdracht 1\_15 16](#_Toc508024905)

[Opdracht 1\_16 16](#_Toc508024906)

# Inleiding

Bij HBO-ICT krijgen alle studenten in het eerste blok programmeren, waarbij als programmeertaal Python wordt gebruikt. Er zijn een aantal redenen waarom voor deze taal is gekozen:

* Vergeleken met andere programmeertalen, zoals bijv. JAVA, is Python makkelijk en snel te leren, ook als je nog helemaal geen programmeerkennis hebt.
* De concepten zijn bruikbaar als je een andere taal leert.
* Je leert netjes programmeren (o.a. *indentation*)

In deze instructie leer je de beginselen van het programmeren in Python. We maken hierbij gebruik van een online omgeving: repl-it. Om in Python te kunnen programmeren klik je op onderstaande link:

<https://repl.it/languages/python3>

In het middelste (witte) scherm typ je de code in.. Aan de rechterkant van het scherm komt de uitvoer, nadat je het programma ‘gerund’ hebt. Dit doe je door op het groene pijltje te klikken linksboven in het scherm.

# Algebraïsche expressies

Python kan worden gebruikt om algebraïsche expressies(rekenkundige bewerkingen) uit te voeren.

Type in het middelste scherm onderstaande code in en klik vervolgens op de run-knop. (Je moet echt de code intypen en niet onderstaande tekst kopiëren en plakken. Je krijgt dan namelijk foutmeldingen!)

print(5 + 2)

print(12 – 5)

print(5 \* 2)

print(5 / 2)

print(2 \* (3 + 7))

Als je naar de uitvoer kijkt, kun je wel bedenken hoe het werkt. De opdracht (statement in het Engels) print gebruiken we om een statement af te drukken op het scherm. De gebruikte symbolen zijn hetzelfde als de symbolen die we normaal gesproken voor de rekenkundige bewerkingen gebruiken m.u.v. de vermenigvuldiging. Daar wordt dus een ‘\*’ gebruikt i.p.v. een ‘x’. Deze symbolen worden rekenkundige operatoren genoemd.

# Nog meer algebraïsche expressies

Typ onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat. Probeer te achterhalen wat de gebruikte rekenkundige operatoren betekenen.

print(2 \*\* 3)

print(14 // 3)

print(16 // 5)

print(14 % 3)

print(16 % 5)

print(abs(2))

print(abs(-2))

print(min(3, 6, 2, 9))

print(max(3, 6, 2, 9))

We bekijken hieronder de uitkomsten nader.

2 \*\* 3 levert als resultaat 8 op, dus het dubbele sterretje staat voor machtsverheffen (2 tot de macht 3).

14 // 3 levert als resultaat 4 op en 16 // 5 heeft als uitkomst 3. Het gaat hier om de zogenaamde gehele deling. Hoeveel keer past 3 in 14 en hoeveel keer past 5 in 16.

14 % 3 heeft als resultaat 2 en 16 // 5 levert 1 op. Het gaat hierbij om de rest bij de gehele deling.

abs(2) heeft als uitkomst 2 en abs(-2) heeft dat ook. Het gaat hierbij om de zogenaamde absolute waarde van het getal. Als het getal positief is, blijft de waarde hetzelfde. Als het getal negatief is, wordt het hetzelfde getal, maar dan positief.

min(3, 6, 2, 9) heeft als uitkomst 2, dus dit levert de minimumwaarde op van de getallen tussen de haakjes.

(max(3, 6, 2, 9) heeft als uitkomst 9 en geeft dus de maximumwaarde van de getallen tussen de haakjes.

abs, min en max zijn zogenaamde standaardfuncties in Python

# Booleaanse expressies

Booleaanse expressies leveren **True** (waar) of **False** (niet waar) op. Let op: in Python worden True en False met een hoofdletter geschreven!

Typ onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat. Probeer te begrijpen welke opdracht steeds uitgevoerd wordt.

print(2 < 3)

print(2 > 3)

print(2 == 3) (let op: dubbel =-teken!!)

print(2 != 3)

print(24 % 2 == 0)

print(25 % 2 == 1)

We bekijken hieronder de uitkomsten nader.

2 < 3 levert de waarde True op, dus de het <-teken staat voor kleiner dan.

2 > 3 geeft als uitkomst False en het >-teken betekent dus groter dan.

False

2 == 3 geeft als uitkomst False. Je vraagt eigenlijk of 2 gelijk is aan 3. Het symbool == staat voor ‘gelijk aan’.

2 != 3 geeft als uitkomst True. Het symbool != staat voor ‘niet gelijk aan’.

24 % 2 == 0 is True. Je vraagt of de rest van 24 bij delen door 2 gelijk aan 0. Ofwel anders gezegd: is 24 een even getal? Deze expressie wordt heel veel gebruikt!

25 % 2 == 1 is ook True. Is de rest van 25 bij delen door 2 gelijk aan 1? Ofwel, is 25 een oneven getal?

Kleiner of gelijk schrijf je in Python als <= en groter of gelijk schrijf je als >=.

# Booleaanse operatoren

Dit zijn and, or en not

Typ onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat. Probeer te begrijpen wat de booleaanse operatoren doen.

print(2 < 3 and 3 < 4)

print(2 > 3 and 3 < 4)

print(2 > 3 or 3 < 4)

print(2 > 3 or 3 > 4)

print(not(2 < 3))

We bekijken hieronder de uitkomsten nader.

(2 < 3 and 3 < 4) is True en (2 > 3 and 3 < 4) is False. Als je de and-operator gebruikt, is het resultaat alleen True als aan beide kanten de expressie True is. In alle andere gevallen wordt de uitkomst False.

(2 > 3 or 3 < 4) is nu True en (2 > 3 or 3 > 4) is False. Als je de or-operator gebruikt, is deze alleen False, als aan beide kanten de expressie False is.

(not(2 < 3)) levert het tegengestelde resultaat op. Is de expressie True, dan wordt not(expressie) False en andersom.

# Toekenningen en variabelen

Toekenning is in het Engels assignment. Een waarde kan worden toegekend aan een variabele.

Voor een toekenning wordt het =-teken gebruikt. Type onderstaande code in, klik op de run-knop en probeer te begrijpen wat er gebeurt.

x = 3

print(x)

print(y)

We beschrijven heel precies wat er gebeurt, als je het programmaatje runt:

Je kent in de eerste opdracht aan de variabele x de waarde 3 toe door een toekenningsopdracht of assignment statement te gebruiken. Vervolgens geef je de opdracht om de variabele x uit te printen. Je ziet in het uitvoerscherm de waarde 3 staan. Echter daarna gaat het fout. Je geeft de opdracht om de variabele y uit te printen, maar deze heb je geen waarde gegeven. Dus je krijgt een foutmelding in het uitvoerscherm. Daarin staat aangegeven in welke regel van het programma de fout is opgetreden en wat voor een soort fout het is, namelijk een NameError waarin staat aangegeven dat y niet gedefinieerd is.

Voor namen van variabelen kun je de volgende tekens gebruiken:

* de letters a t/m z
* de letters A t/m Z
* underscore: \_
* de getallen o t/m 9

Variabelen kunnen niet beginnen met een getal!

Voor lange namen zijn er twee oplossingen:

Gebruik underscore: dit\_is\_een\_lange\_naam

Gebruik camelcase: ditIsEenLangeNaam

# Strings

Naast getallen en booleaanse waarden bestaan er in Python ook strings.

Een string is een reeks karakters die wordt omsloten met enkele of dubbele quotes:

'Hello, world'

"Hello, world"

Typ onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat.

s = 'rock'

t = 'climbing'

x = 'oc'

print(x in s)

print(x not in s)

print(s + t)

print(s \* 3)

print(s[2])

print(len(s))

We bekijken hieronder de resultaten wat nader.

De eerste drie opdrachten zijn toekenningsopdrachten of assignment statements. Links van het =-teken staat de naam van de variabele en rechts van het –teken staat de waarde die aan de variabele wordt toegekend. In dit geval zijn het allemaal strings.

x in s levert True op. Er wordt gevraagd of de string 'oc' deel uitmaakt van de string 'rock' of iets formeler gezegd. Is x een substring van s?

x not in s vraagt of de string 'oc' geen substring is van'rock'. Dit klopt dus niet en daarom is de uitkomst False.

s + t geeft als resultaat 'rockclimbing'. De twee strings worden ‘aan elkaar geplakt’. Met een moeilijk woord wordt dit concatenatie genoemd.

s \* 3 levert 'rockrockrock' op. Het is de concatenatie van 3 kopieën van de string 'rock'

s[2] geeft als resultaat de letter 'c'. Dit is de derde letter in de string 'rock'. Hoe zit dat dan? Het getal tussen de vierkante haken geeft inderdaad de letter op de betreffende positie, maar Python begint te tellen bij o en niet bij 1! Daarom krijg je bij positie of index 2 de derde letter.

len(s) geeft als uitkomst 4 en dat is de lengte van de string. Dit is een standaardfunctie voor strings.

# Soorten getallen

Python kent twee soorten getallen:

* int: gehele getallen
* float: gebroken getallen

Let op: gebroken getallen worden genoteerd met een punt en niet met een komma! De manier van declareren (bekend maken van de variabele) geeft het soort getal aan, bijv.

x = 3.56

Vanwege de punt wordt x een float getal.

# Het gebruik van de index bij strings

Typ onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat.

woord = 'heelErgLangWoord'

print(woord[0])

print(woord[15])

print(woord[-1])

We bekijken hieronder de resultaten nader.

Het getal tussen de vierkante haken wordt de index genoemd.

De opdracht woord[0] geeft de eerste letter, omdat Python begint te tellen bij 0.

De opdracht print(woord[15]) geeft de laatste letter, want het woord bestaat uit 16 letters.

Gelukkig bestaat er een minder omslachtige manier om de laatste letter van een lang woord te krijgen. Je begint dan namelijk van achteraan te tellen, maar dan met negatieve getallen. In dit geval geeft woord[-1] de letter 'd', woord[-2] de letter 'r', enz.

# Python standaard bibliotheken

In het Engels wordt een bibliotheek een **library** genoemd.

Sommige functies zijn altijd beschikbaar, zoals bijvoorbeeld len, sum, min en max. Veel andere functies zijn beschikbaar in de Python Standard Library om bijvoorbeeld het volgende te ondersteunen:

* programmeren van webapplicaties
* het maken van grafische user interfaces
* voor het toepassen van wiskundige of mathematische functies
* etc.

Een bibliotheek moet geïmporteerd worden, voordat die kan worden gebruikt. Voor mathematische functies is dit import math

Typ onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat.

import math

oppervlakte = math.pi \* 10\*\*2

print(oppervlakte)

schuinezijde = math.sqrt(3\*\*2 + 4\*\*2)

print(schuinezijde)

Wat gebeurt er in de code?

In de eerste opdracht wordt de biblitotheek voor wiskundige functies geïmporteerd.

Wat betekent de notatie math.pi? Dit is de Python-notatie voor het getal pi (∏). Hier wordt de oppervlakte van een cirkel uitgerekend met een straal van 10 . Dit is pi keer de straal in het kwadraat.

De derde opdracht print de oppervlakte uit.

In de vierde opdracht staat math.sqrt. Dit is de Python-notatie voor een √ (wortel). Wat wordt in deze opdracht gedaan? Hier wordt de stelling van Pythagoras toegepast. Je kunt de lengte van de schuine zijde van een rechthoekige driehoek berekenen, als de lengte van de andere twee zijden kent (in dit geval 3 en 4). De Python-notatie math.sqrt(3\*\*2 + 4\*\*2) is de tegenhanger van de wiskundige notatie.

De vijfde opdracht print het resultaat van de vierde opdracht uit.

# Lists

Naast getallen, booleans en strings kun je in Python ook lists gebruiken als datatype.

Declaratie van een list: elementen van de lijst worden gescheiden door komma’s en de lijst zelf wordt omsloten met vierkante haken

Voorbeeld van een lijst bestaande uit vijf gehele getallen:

getallenrij = [0, 2, 4, 6, 8]

Een list bestaande uit vijf strings van drie letters:

pets = ['ant', 'bat', 'cod', 'dog', 'elk']

Een list waarvan de elementen zelf ook weer list zijn: een list in een list.

lijstineenlijst = [[1,3], [2,4], [3,5]]

Bij een list hoeven niet alle elementen van hetzelfde type te zijn. Onderstaande declaratie is dus toegestaan:

lst = [0, 1, 'two', 'three', [4, 'five']]

# List-operatoren

Typ onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat.

lijst1 = [0, 1, 2]

lijst2 = [3, 4, 5]

print(2 in lijst1)

print(2 not in lijst2)

print(lijst1 + lijst2)

print(4\*lijst1)

print(lijst1[2])

We kijken weer naar de resultaten:

2 in lijst1 geeft als resultaat True. De operator in geeft aan of iets in een lijst zit.

2 not in lijst2 geeft als resultaat False. Deze operator is het tegengestelde van in.

Het resultaat van lijst1 + lijst2 is [0, 1, 2, 3, 4, 5]. Het is dus een concatenatie van de lijsten, waarbij de elementen van lijst1 als eerste staan genoemd en daarachter de elementen van lijst2.

4\*lijst1 geeft [0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2] dus een concatenatie van vier keer dezelfde lijst.

lijst1[2] geeft als waarde 2. De notatie kennen we al van strings. Ook hier wordt het element met de gevraagde index teruggegeven, waarbij de index van het eerste element, net als bij strings, de waarde 0 heeft.

# List-functies

Type onderstaande expressies in, klik daarna op de run-knop en kijk naar het resultaat.

lijst1 = [0, 1, 2, 3]

print(len(lijst1))

print(min(lijst1))

print(max(lijst1))

print(sum(lijst1))

Als we kijken naar de resultaten zien we de standaardfuncties len, min en max die we al eerder hebben gezien en bij lists op dezelfde manier werken. De enige nieuwe standaardfunctie is sum. Deze geeft als resultaat 6 en telt de getallen bij elkaar op.

Wat gebeurt er in het volgende geval?

sum(['a', 'b', 'c', 'd'])

Er zal een foutmelding optreden, want de functie sum werkt alleen, als er in de lijst getallen staan. In deze lijst staan strings en dat lukt dus niet.

We praten hier over functies, omdat len, min, max en sum kunnen worden toegepast bij andere datatypen, bijvoorbeeld int of float.

# List-methoden: syntax

Functies die alleen kunnen worden aangeroepen bij lists noemen **listmethoden**.

Schrijfwijze (syntax) van een listmethode is als volgt:

lijst.append(7)

lijst = naam van de list

append = methodenaam

7 = inputargument

(Als er geen inputargument is, staan er alleen haakjes)

# Veel gebruikte list-methoden

Om veel gebruikte list-methoden te demonstreren gebruiken we onderstaand programma. Je kunt het zelf intypen in repl-it.

lijst = ['a', 'b', 'c', 'd', 'e', 'f']

lijst.append('d')

print(lijst)

print(lijst.count('d'))

print(lijst.index('d'))

lijst.insert(2, 'g')

print(lijst)

lijst.remove('d')

print(lijst)

lijst.reverse()

print(lijst)

lijst.sort()

print(lijst)

Als je het programma runt, dan levert dat acht uitvoerregels op, voor elke printopdracht één. Hieronder bekijken we alle printopdrachten om te kijken wat de list-methoden doen.

lijst = ['a', 'b', 'c', 'd', 'e', 'f']

lijst.append('d')

print(lijst)

levert als uitvoer ['a', 'b', 'c', 'd', 'e', 'f', 'd']

De methode append voegt het inputargument (hier de letter 'd') achteraan de lijst toe.

print(lijst.count('d'))

levert als uitvoer 2

De methode count geeft terug hoe vaak het inputargument voorkomt in de lijst.

print(lijst.index('d'))

levert als uitvoer 3

De methode index geeft de index van het *eerste* element van het inputargument

lijst.insert(2, 'g')

print(lijst)

levert als uitvoer ['a', 'b', 'g', 'c', 'd', 'e', 'f', 'd']

Bij de methode insert geeft het eerste inputargument de index aan waar het tweede inputargument moet worden toegevoegd. De overige elementen schuiven allemaal een plaats naar rechts op.

lijst.remove('d')

print(lijst)

levert als uitvoer ['a', 'b', 'g', 'c', 'e', 'f', 'd']

De methode remove verwijdert het eerste element met de waarde van het inputargument.

lijst.reverse()

print(lijst)

levert als uitvoer ['d', 'f', 'e', 'c', 'g', 'b', 'a']

De methode reverse draait de lijst om.

lijst.sort()

print(lijst)

levert als uitvoer ['a', 'b', 'c', 'd', 'e', 'f', 'g']

De methode sort sorteert de lijst alfabetisch of bij getallen van klein naar groot.

# Opdrachten

## Opdracht 1\_1

Bereken de uitkomst van de onderstaande expressies met behulp van Python.

1. Het verschil in leeftijd tussen Annie (25) en Ellie (21)
2. Het totaal van 14.99, 27.95 en 19.83
3. De oppervlakte van een rechthoek met lengte 20 en breedte 15
4. 2 tot de macht 10
5. Het minimum van 3, 1, 8, -2, 5, -3 en 0
6. Is 3 gelijk aan 4 - 2?
7. Is 17 // 5 gelijk aan 3?
8. Is 17 % 5 gelijk aan 3?
9. Is 284 even?
10. Is 284 even **en** deelbaar door 3?
11. Is 284 even **of** deelbaar door 3?

## Opdracht 1\_2

s1 = ‘good’, s2 = ‘bad’ en s3 = ‘silly’

Schrijf expressies voor de volgende opdrachten:

1. Of ‘ll’ voorkomt in s3
2. Of de spatie niet voorkomt in s1
3. Die s1, s2 en s3 verbindt of ‘aan elkaar plakt’
4. Of de spatie voorkomt in de aan elkaar geplakte strings
5. Die 10 keer s3 aan elkaar plakt
6. Die berekent hoeveel karakers er zitten in s1, s2 en s3 samen

## Opdracht 1\_3

Begin met de volgende toekenningen:

s1 = '-'

s2 = '+'

Schrijf stringexpressies die gebruik maken van s1 en s2 en de string operatoren + en \* die onderstaande uitkomsten opleveren. Maak de stringexpressies zo kort mogelijk.

a. '-+'

b. '-+-'

c. '+--'

d. '+--+--'

e. '+--+--+--+--+--+--+--+--+--+--+'

f. '+-+++--+-+++--+-+++--+-+++--+-+++--'

## Opdracht 1\_4

Begin met de volgende toekenning

s= 'abcdefghijklmnopqrstuvwxyz'

Schrijf nu expressies die string s gebruiken en de index operator die de volgende resultaten opleveren: 'a', 'c', 'z', 'y' en 'q'

## Opdracht 1\_5

Start met de volgende toekenning:

s = 'goodbye'

Schrijf daarna booleaanse expressies die het volgende controleren:

a. het eerste karakter van de string is een 'g'

b. het zevende karakter van s is een 'g'

c. de eerste twee karakters van s zijn 'g' en 'a'

d. het één na laatste karakter van s is een 'x'

e. het middelste karakter van s is een 'd'

f. het eerste en het laatste karakter van de string zijn gelijk

g. de laatste vier karakters van de string komen overeen met 'tion'

## Opdracht 1\_6

Schrijf en evalueer Python expressies die de volgende vragen beantwoorden:

1. Hoeveel letters zijn er in 'Supercalifragilisticexpialidocious'?
2. Komt in 'Supercalifragilisticexpialidocious' de tekst 'ice' voor?
3. Is het woord 'Antidisestablishmentarianism' langer dan 'Honorificabilitudinitatibus'?

## Opdracht 1\_7

Schrijf een python expressie die aan de variabele x de volgende waarde toekent

1. De lengte van de schuine zijde of hypotenusa van een rechthoekige driehoek waarvan de rechthoekszijde de waarden 5 en 12 hebben.
2. Een boolean die berekent of de lengte van de hypotenusa van de rechthoek bij onderdeel a kleiner dan 14 is.
3. De oppervlakte van een cirkel met straal 7.

## Opdracht 1\_8 (lastig!)

Dit is een lastige opgave. Het is niet erg, als die niet lukt.

Een ladder wordt tegen een muur gezet. Bekend is hoe lang de ladder is en de hoek die de ladder maakt met de grond. Bereken met behulp van Python hoe hoog de ladder reikt, als de volgende gegevens bekend zijn:

1. hoogte is 10 meter en de hoek is 75 graden.
2. hoogte is 12 meter en de hoek is 0 graden.
3. hoogte is 18 meter en de hoek is 45 graden.
4. hoogte is 18 meter en de hoek is 80 graden.

Voor het berekenen van de hoogte heb je de volgende wiskundige formule nodig:

hoogte = lengte \* sin(hoek in radialen).

De hoek is gegeven is graden, maar Python rekent met radialen. Met onderstaande formule kun je de graden omrekenen naar radialen:

hoek in radialen = (π \* hoek in graden) / 180

## Opdracht 1\_9

1. Ken aan de variabele a de waarde 6 toe en aan variabele b de waarde 7.
2. Ken aan de variabele c het gemiddelde van a en b toe.
3. Ken aan de variabele voorraad de list met de strings 'papier', 'nietjes' en 'potloden' toe.
4. Ken aan de variabelen first, middle en last de strings 'John', 'Fitzgerald' en 'Kennedy' toe.
5. Ken aan de variabele fullname de concatenatie van de drie strings 'John', 'Fitzgerald' en 'Kennedy' toe. Zorg dat er een spatie staat tussen alle drie de delen.

## Opdracht 1\_10

Gebruik de variabelen die je hebt gedeclareerd bij de vorige opdracht. Schrijf booelaanse expressies die het volgende berekenen. (Alle uitkomsten hebben dus de waarde True of False.)

1. De som van a en b is kleiner dan 10.
2. De lengte van de list voorraad is meer dan vijf keer de lengte van de string fullname.
3. c is niet meer dan 24.
4. Het getal 6.75 ligt tussen de waarden van a en b.
5. De lengte van de string middle is langer dan de lengte van string first en korter dan de lengte van string last.
6. De lst voorraad is leeg of de lst voorraad heeft meer dan 10 elementen.

## Opdracht 1\_11

Schrijf python statements volgens onderstaande instructies:

1. Ken aan de variable flowers een list toe met de volgende strings: 'rose', 'bougainvillea', 'yucca', 'marigold', 'daylilly', en 'lilly of the valley'.
2. Schrijf een booleaanse expressie die berekent of de string 'potato' in de list flowers voorkomt.
3. Ken aan de list thorny de sublist toe die bestaat uit de eerste drie elementen van de list flowers.
4. Ken aan de list poisonous de sublijst toe die alleen bestaat uit het laatste element van de list flowers.
5. Ken aan de list dangerous de concatenatie toe van de lists thorny en poisonous.

## Opdracht 1\_12

Start met het toekennen aan een variabele answers een list die bijvoorbeeld de volgende waarden bevat: answers = ['Y', 'N', 'N, 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'N', 'N']

Schrijf python statements (opdrachten) die corresponderen met het onderstaande:

a. De variabele numYes geeft aan hoe vaak 'Y' in de list voorkomt.

b. De variabele numNo geeft aan hoe vaak 'N' in de list voorkomt.

c. De variabele percentYes geeft het percentage van de strings aan dat de waarde 'Y' bevat.

d. Sorteer de list answers.

e. De variabele f geeft de index aan van de eerste keer dat 'Y' voorkomt in de gesorteerde list.

## Opdracht 1\_13

De opdracht is om van een list van getallen het verschil tussen het grootste en kleinste getal van de list te berekenen. Schrijf hiervoor in python een expressie en kijk of de expressie klopt door het te testen op een aantal lists van getallen. Begin met het declareren van een list met een rij getallen.

## Opdracht 1\_14

Schrijf een python statement dat een string van drie letters in omgekeerde volgorde afdrukt. Bijvoorbeeld als s = 'top' dan wordt 'pot' afgedrukt. Begin met het declareren van een string bestaande uit drie letters.

## Opdracht 1\_15

Begin met het declareren van de list maanden bestaande uit de volgende maanden: 'Jan', 'Feb', 'Mrt' en 'Mei'. Voer de onderstaande opdrachten uit en druk het resultaat na iedere opdracht af op het scherm om te controleren of de opdracht juist was.

1. Voeg in de list 'Apr' toe tussen 'Mrt' en 'Mei'.
2. Voeg 'Jun' aan de list toe.
3. Verwijder het laatste element uit de lijst.
4. Verwijder het tweede element uit de lijst.
5. Draai de volgorde van de lijst om.
6. Sorteer de list op alfabetische volgorde.

## Opdracht 1\_16

Maak een variable *myName* en *myAddress* aan. Ken aan deze varabelen jouw eigen waarden toe en print de volgende zin op het scherm.

'Ik ben *myName* en mijn adres is: *myAddress*'