

**WEEK 1****ONDERWERPEN**

- numbers
- strings
- control statements
- lists

**OPGAVE 1 : LIJSTEN MAKEN**

- a) maak een lijst met elementen 2, 3 en 4
- b) maak een lijst met elementen 'red', 'green' en 'blue'
- c) maak een lijst met elementen 3, 4 en 5 waarbij range() wordt gebruikt
- d) maak een lijst met elementen 'a', 'b', 'c' en 'd'

**OPGAVE 2 : FUNCTIES VOOR LIJSTEN**

Gegeven lijst L = [30, 1, 2, 1, 0, "hello", "Goodbye"]. Geef aan wat het resultaat is na elke statement (waarbij L na elke statement weer is als aan het begin).

- a) L.index(1)
- b) L.count(1)
- c) len(L)
- d) max(L)
- e) L.append(40)
- f) L.insert(1, 43)
- g) L.extend([1, 43])
- h) L.remove("hello")
- i) L.pop()
- j) "Goodbye" in L
- k) L.pop(3)
- l) L.sort()
- m) random.shuffle(L)

**OPGAVE 3 : LIST SLICING**

Gegeven lijst L = ['a', 'b', 'c', 'd', 'e']. Geef aan wat het resultaat is na elke statement (waarbij L na elke statement weer is als aan het begin).

- a) L[1 : -3]
- b) L[-4 : -2]
- c) L[:3]
- d) L[:2] + L[2:]

- e) `L[::-1]`
- f) `L[::1]`
- g) `L[:]`

#### OPGAVE 4 : TWEE LIJSTEN

Gegeven twee lijsten `L1 = [30, 1, 2, 1, 0]` en `L2 = [1, 21, 13]`. Geef aan wat het resultaat is na elke statement (waarbij `L` na elke statement weer is als aan het begin).

- a) `L1 + L2`
- b) `3 * L2`
- c) `L1 > L2`
- d) `[x for x in L1]`
- e) `[x for x in L1 if x in L2]`

Opm. (d) en (e) zijn voorbeelden van list comprehension, komt verder aan de orde in week 2.

#### OPGAVE 5 : LIJSTEN EN STRINGS

Gegeven de string `s = 'Guido van Rossum'`. Converteer `s` naar `L = ['Guido', 'van', 'Rossum']`.

(De uitwerking kan heel kort zijn).

#### OPGAVE 6 : WAT WORDT ER AFGEDRUKT ?

Wat wordt er afgedrukt ?

(a)

```
L = [1, 2, 3, 4, 5, 6]
for i in range(1, 6):
    L[i] = L[i - 1]
print(L)
```

(b)

```
L1 = list(range(1, 10, 2))
L2 = L1
L1[0] = 'abc'
print(L1)
print(L2)
```

(c)

```
a, b = 0, 1
while b < 10:
    print(b)
    a, b = b, a+b
```

## OPGAVE 7 : PALINDROOM

Een palindroom is een woord dat van achteren naar voren hetzelfde woord oplevert. De woorden 'lepel' en 'parterretrap' zijn bijvoorbeeld palindromen. Schrijf een programma dat controleert of een woord een palindroom is.

## OPGAVE 8 : GEVOELSTEMPERATUUR

De gevoelstemperatuur is niet hetzelfde wat de thermometer aangeeft. Door wind kan het een stuk kouder aanvoelen dan de thermometer aangeeft. Dit verschijnsel staat ook wel bekend als "windchill". Het warmteverlies drukken we uit in een gevoelswaarde van de temperatuur: de gevoelstemperatuur.

De formule voor de gevoelstemperatuur (G) op is bij benadering :

$$G \approx 13 + 0,62 \cdot T - 14B^{0,24} + 0,47TB^{0,24}$$

Hierin is T de temperatuur in °C (-20..10) en B de windsnelheid in Beaufort (1..9). Zie <https://nl.wikipedia.org/wiki/Gevoelstemperatuur>.

Voorbeeld: bij een temperatuur van -10 graden en een windkracht van 5 Beaufort is de gevoelstemperatuur -20.7 graden.

Schrijf een programma dat de gebruiker om een temperatuur in °C (-20..10) vraagt en de windkracht volgens de Schaal van Beaufort, waarna de gevoelstemperatuur wordt getoond.

Tip: Kijk eens naar de build-in functies input() en int() en de \*\*-operator voor het machtsverheffen.

## OPGAVE 9 : VIND DE GENEN

Het genoom is de complete genetische samenstelling van een organisme. Een string of sequence met de letters A,C,T,G wordt gebruikt om een genoom te beschrijven. Gegeven is de onderstaande sequence:

```
ACAAGATGCCATTGTCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGGCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACCTTCTTCTGGAAGACCTTCTCCTCTGCAAATAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

- Doorloop de sequence in een lus en zoek hierbij op de triplets ATG. Druk op het scherm de index af van de letter die volgt op een triplet ATG. Dit zijn 5 waarden 8, 110, 278, 336 en 340. Met behulp van slicing kun je triplets uit de sequence halen, bijvoorbeeld met `s[i : i + 3]` waarbij `s` de sequence en `i` een lopende variabele.
- Een gen is een sub-string van het genoom die begint na een triplet ATG en eindigt voor een triplet TAG, TAA of TGA. Bovendien is de lengte van een gen-string een veelvoud van 3 en het gen zelf bevat *niet* een van de triplets ATG, TAG, TAA of TGA. Schrijf een programma dat alle genen uit onderstaande sequence toont. Controleer je antwoord met: `assert (len(gene) == 42 or len(gene) == 12)`

Wat voorbeelden die je zelf eenvoudig kan testen:

```
S = AATGCCCTGA geeft 'CCC'  
S = AATGTGA geeft geen match  
S = AATGCCCTGA geeft geen match  
S = AATGCCCTGATGCCCTAG geeft 'CCC'
```

## OPGAVE 10 : DE GEMIDDELDE TEMPERATUUR

Op Blackboard is een bestand "station\_vl.txt" te vinden met gemeten temperaturen door een weerstation op Vlieland over een periode van 10 dagen (bron : <https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens>) . Elke tekstregel bevat 3 items : nummer van de dag (1..10), uur van de dag (1..24) en temperatuur in 0.1 °C.

Je kan de tekstfile inlezen m.b.v. re-direction, door stdin op de command line te vervangen door de tekstfile, dus "python my\_solution.py < station\_vl.txt". Dan hoeft je niet in je code een tekstfile te openen.

Neem de volgende code over en maak het programma af.

```
from sys import stdin  
  
NUMBER_OF_DAYS = 10  
NUMBER_OF_HOURS = 24  
  
data = []  
  
for i in range(NUMBER_OF_DAYS):  
    data.append([])  
    for j in range(NUMBER_OF_HOURS):  
        data[i].append([])  
  
# read input using input redirection from a file  
for line in stdin:  
    if line[0] == '#':  
        continue  
    L = line.strip().split()  
  
    # ... hier jouw code ...  
  
print("Gemiddelde temperaturen: ")  
  
# find the average daily temperature  
for i in range(NUMBER_OF_DAYS):  
    # ... hier jouw code ...
```