



Unit 3: Programmeren in Python

Oefenblad 3 : Lijsten

In deze Unit gaan we wat dieper in op programmeren in Python.

Lijsten in Python

Doelen:

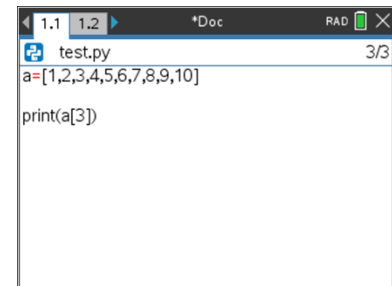
- Lijsten maken
- Verschillende lijst-opdrachten gebruiken.

Een lijst wordt in Python genoteerd als een opsomming van elementen tussen rechte haken waarbij de elementen door een komma gescheiden worden.

Zo is bijvoorbeeld [7,3,5] een lijst met 3 elementen (in dit geval getallen).

Elementen kunnen ook van een ander type zijn bijvoorbeeld tekst of zelf weer een lijst.

Open een nieuw Python programma en maak een lijst a met de getallen 1 t/m 10. Elk element in de lijst heeft een rangnummer waarbij het eerste element nummer 0 heeft. Met a[3] krijg je het element met rangnummer 3. Voeg een regel aan het programma toe waarbij je het derde element afdruckt en ga na dat dit element de waarde 4 heeft.



```
1.1 1.2 *Doc RAD 3/3
test.py
a=[1,2,3,4,5,6,7,8,9,10]
print(a[3])
```

Je kunt elementen toevoegen aan een rij met **lijst.append()**.

Dit kun je vinden in Menu --> Built-ins --> Lists.

Als bijvoorbeeld a = [1,2,3], dan is a.append(5) de lijst a = [1,2,3,5]

We gaan dit gebruiken om een lijst van Fibonaccigetallen te maken.

De rij van Fibonacci is: 1, 1, 2, 3, 5, 8, 13, ... waarbij steeds het volgende getal de som is van de twee vorige getallen.

Begin met de lijst fib = [1,1].

Dan moeten we de eerste keer toevoegen fib[0] + fib[1].

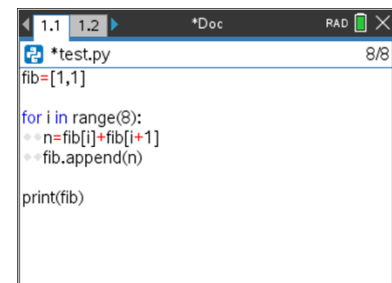
Er geldt dan: fib = [1,1,2]

De tweede keer voegen we fib[1] + fib[2] toe. Dit herhalen we steeds weer opnieuw.

Maak een programma dat de rij van de eerste 10 Fibonaccigetallen maakt en gebruik hiervoor een for-loop.

Een mogelijk programma staat hiernaast.

De loop wordt 8 keer doorlopen want we hoeven maar 8 getallen toe te voegen aan de lijst [1,1].



```
1.1 1.2 *Doc RAD 8/8
*test.py
fib=[1,1]
for i in range(8):
    n=fib[i]+fib[i+1]
    fib.append(n)
print(fib)
```



Op deze manier kun je makkelijk een lijst van 100 Fibonaccigetallen maken.
Ga na dat het programma hiernaast het 100-ste Fibonaccigetal afdruckt.

```
test.py 7/8
fib=[1,1]

for i in range(98):
    n=fib[i]+fib[i+1]
    fib.append(n)

print(fib[99])
```

Een lijst kan ook bestaan uit elementen die zelf een lijst zijn.
Dit is een tweedimensionale lijst of matrix.
We maken nu zo'n lijst waarbij elk element een lijst met Fibonaccigetallen is te beginnen met [1,1].
Verander het vorige programma eerst zo dat de functie f(a) een lijst geeft met Fibonaccigetallen.
Ga na dat f(8) uit het programma hiernaast een lijst met 10 Fibonaccigetallen geeft.

```
*test.py 8/8
def f(a):
    fib=[1,1]
    for i in range(a):
        n=fib[i]+fib[i+1]
        fib.append(n)
    return fib

L=[]
for i in range(4):
    L.append(f(i))
print(L)
```

Definieer nu een lege lijst. (bijvoorbeeld L = [])
Maak dan een for-loop waarbij steeds de lijst f(i) wordt toegevoegd aan L.
Zie hiernaast.
L bestaat uit vier lijsten met Fibonaccigetallen.

```
*test.py 5/12
def f(a):
    fib=[1,1]
    for i in range(a):
        n=fib[i]+fib[i+1]
        fib.append(n)
    return fib

L=[]
for i in range(4):
    L.append(f(i))
print(L)
```

De lijst L uit dit programma bestaat uit lijsten.
Als we die lijsten onder elkaar willen afdrukken dan kan dat met een for-lus.
We gebruiken hiervoor de optie **for i in L**. (dit kun je vinden in Menu --> Built-ins --> Control.
In deze lus doorloopt i alle elementen van de lijst L. (Dus i is steeds zelf een lijst)
Breed het programma uit met zo'n loop en druk alle lijsten uit L af.

```
test.py 13/14
for i in range(a):
    n=fib[i]+fib[i+1]
    fib.append(n)
    return fib

L=[]
for i in range(4):
    L.append(f(i))

for i in L:
    print(i)
```