



Unit 5 : Combinatie van ti_system en ti_plotlib

Oefenblad 3: De vrije val

In deze les maken we een Python programma dat verschillende weergaves geeft van de vrije valbeweging.

Doelen :

- Plotten van lijsten.
- Lijsten exporteren naar de List applicatie van de TI84

Een bal valt van een hoogte van 18 meter.

De hoogte van de bal als functie van de tijd kan worden benaderd met de formule $h = 18 - \frac{1}{2} \cdot g \cdot t^2$. Hierin is g de valversnelling ($g \approx 9.81$)

In het programma nemen we de formule: $h = 18 - 4.9 \cdot t^2$

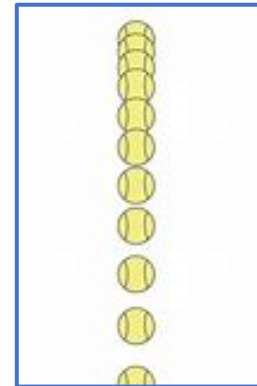
We meten gedurende twee seconden de hoogte in stappen van 0.1 seconden.

We maken twee verschillende soorten grafieken.

Eerst een die de positie van de bal weergeeft in de tijd.

Daarvoor geldt dat de x-coördinaat telkens dezelfde is.

Daarna een die de hoogte van de bal weergeeft als functie van de tijd.



We beginnen het programma met het importeren van de ti_plotlib module.

Daarna definiëren we drie lijsten. Een voor de x-coördinaat, een voor de y-coördinaten en een voor de tijd.

Bij de start zijn ze alle drie leeg.

```

EDITOR: US
PROGRAM LINE 0006
import ti_plotlib as plt

x=[]
y=[]
tijd=[]

```

We maken nu een for-lus waarin 20 keer de hoogte (h) wordt berekend en waarbij elke keer de tijd (t) met 0.1 seconde wordt opgehoogd.

In de lus worden steeds de lijsten aangevuld met de nieuwe waarden.

(De lijst met x-coördinaten bestaat uit alleen maar nullen.)

De startwaarde van t is 0.

```

EDITOR: US
PROGRAM LINE 0011
y=[]
tijd=[]

t=0
for i in range(20):
  **h=18-4.9*t**2
  **x.append(0)
  **y.append(h)
  **tijd.append(t)
  **t=t+0.1

```



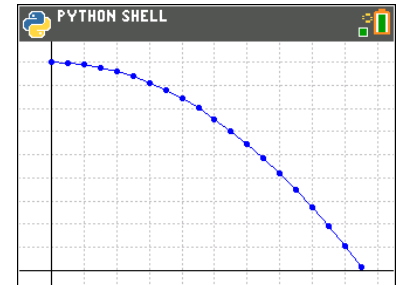
We maken eerst een grafiek van de **positie** (x, y) van de bal in de tijd.
We gebruiken daarvoor de scatterplot uit het ti_plotlib menu.

```
EDITOR: US  
PROGRAM LINE 0020  
**x.append(0)  
**y.append(h)  
**tijd.append(t)  
**t=t+0.1  
  
plt.cls()  
plt.window(-1,1,0,20)  
plt.color(0,0,255)  
plt.scatter(x,y,"o")  
plt.show_plot()
```

De grafiek van de hoogte als functie van de tijd ziet er anders uit.
Op de horizontale as staan nu niet de x-waarden, maar de tijd-waarden.
We hebben daarvoor een andere vensterinstelling nodig.
Een handige optie is **auto_window()** uit het ti_plotlib menu.
Verder is in het programma nog een rooster getekend en is de puntenwolk (scatter) vervangen door de plot().

```
EDITOR: US  
PROGRAM LINE 0022  
**tijd.append(t)  
**t=t+0.1  
  
plt.cls()  
plt.auto_window(tijd,y)  
plt.grid(0.2,2,"dot")  
plt.axes("axes")  
plt.color(0,0,255)  
plt.plot(tijd,y,"o")  
plt.show_plot()
```

De grafiek zie er dan als volgt uit:



Als we nu nog drie regels toevoegen kunnen we ook de lijsten uit de rekenmachine gebruiken voor een grafiek.

Eerst moeten we de `ti_system` module importeren.

Daarin staat de functie `store_list()`.

Met deze functie kun je lijsten exporteren naar de grafische rekenmachine.

Deze functie heeft twee argumenten. Eerst komt het nummer van de lijst uit de grafische rekenmachine (tussen aanhalingstekens). Daarna de lijst uit het Python programma.

```

EDITOR: US
PROGRAM LINE 0026
plt.auto_window(tijd,y)
plt.grid(0.2,2,"dot")
plt.axes("axes")
plt.color(0,0,255)
plt.plot(tijd,y,"o")
plt.show_plot()

from ti_system import *
store_list("1",tijd)
store_list("2",y)
    
```

Sluit de Python-app af en ga na dat de lijsten L_1 en L_2 van de grafische rekenmachine de waarden bevatten van de lijsten uit het Python programma.

L1	L2	L3	L4	L5	2
0	18	---	---	---	
0.1	17.951				
0.2	17.804				
0.3	17.559				
0.4	17.216				
0.5	16.775				
0.6	16.236				
0.7	15.599				
0.8	14.864				
0.9	14.031				
1	13.1				

L2(1)=18

Een plot van deze lijsten kan er dan uitzien zoals hiernaast:

