

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# 2. Vorlesung, 19.10.17
```

1 Unser erstes Python-Programm

```
# Wir berechnen die Trajektorie eines Balls

import numpy as np

g = 9.81      # Gravitationsbeschleunigung
v0 = 20.      # Anfangsgeschwindigkeit
theta = 45.   # Winkel mit x-Achse (in Grad)
y0 = 2.       # Anfangsposition des Balls ist (0,y0)

theta = theta*np.pi/180 # Winkel in Bogenma
x = 20        # Position auf x-Achse

y = x*np.tan(theta) \
    - g*x**2 / (2*v0**2 * np.cos(theta)**2) + y0 # Hoehe des Balls
print("y= ", y)
```

|y= 12.19

2 Wir definieren uns eine Funktion

```
def y(x):

    import numpy as np

    g = 9.81      # Gravitationsbeschleunigung
    v0 = 20.      # Anfangsgeschwindigkeit
    theta = 45.   # Winkel mit x-Achse (in Grad)
    y0 = 2.       # Anfangsposition des Balls ist (0,y0)

    theta = theta*np.pi/180

    return x*np.tan(theta) \
        - g*x**2 / (2*v0**2 * np.cos(theta)**2) + y0 # Hoehe des Balls

# Berechne die Hoehe des Balls fuer x=20 unter Verwendung der Funktion
print(y(20))
```

|12.19

3 Die While-Schleife

```
# Berechne fr x=0,1,2,3,... die Werte y(x)
# Beende die Berechnung sobald y(x)<0 ist.
```

```

def y(x):
    import numpy as np

    g = 9.81      # Gravitationsbeschleunigung
    v0 = 20.      # Anfangsgeschwindigkeit
    theta = 45.   # Winkel mit x-Achse (in Grad)
    y0 = 2.       # Anfangsposition des Balls ist (0,y0)

    theta = theta*np.pi/180

    return x*np.tan(theta) \
        - g*x**2 / (2*v0**2 * np.cos(theta)**2) + y0 # Hoehe des Balls

x=0.
while y(x)>=0:
    print(x,y(x))
    x = x+1.

```

```

0.0 2.0
1.0 2.975475
2.0 3.9019
3.0 4.779275
4.0 5.6076
5.0 6.386875
6.0 7.1171
7.0 7.798275
8.0 8.4304
9.0 9.013475
10.0 9.5475
11.0 10.032475
12.0 10.4684
13.0 10.855275
14.0 11.1931
15.0 11.481875
16.0 11.7216
17.0 11.912275
18.0 12.0539
19.0 12.146475
20.0 12.19
21.0 12.184475
22.0 12.1299
23.0 12.026275
24.0 11.8736
25.0 11.671875
26.0 11.4211
27.0 11.121275
28.0 10.7724
29.0 10.374475
30.0 9.9275
31.0 9.431475
32.0 8.8864
33.0 8.292275
34.0 7.6491
35.0 6.956875
36.0 6.2156
37.0 5.425275
38.0 4.5859
39.0 3.697475

```

```
40.0 2.76
41.0 1.773475
42.0 0.7379
```

4 Auswertung der Funktion an mehreren Punkten

```
def y(x):
    import numpy as np

    g = 9.81      # Gravitationsbeschleunigung
    v0 = 20.      # Anfangsgeschwindigkeit
    theta = 45.   # Winkel mit x-Achse (in Grad)
    y0 = 2.       # Anfangsposition des Balls ist (0,y0)

    theta = theta*np.pi/180

    return x*np.tan(theta) \
        - g*x**2 / (2*v0**2 * np.cos(theta)**2) + y0 # Hhe des Balls

xliste = [0., 1., 2., 3., 4., 5.] # erzeugt Liste von x-Werten
yvalues = [y(x) for x in xliste] # erzeugt Liste zugehöriger y-Werte

print(yvalues)
```

```
[2.0, 2.9754749999999999, 3.9018999999999995, 4.7792750000000002,
5.6075999999999997, 6.386874999999999]
```

5 Graphische Darstellung

5.1 1. Variante: Wir nutzen die gerade erzeugte Liste

```
import matplotlib.pyplot as plt

def y(x):
    import numpy as np

    g = 9.81      # Gravitationsbeschleunigung
    v0 = 20.      # Anfangsgeschwindigkeit
    theta = 45.   # Winkel mit x-Achse (in Grad)
    y0 = 2.       # Anfangsposition des Balls ist (0,y0)

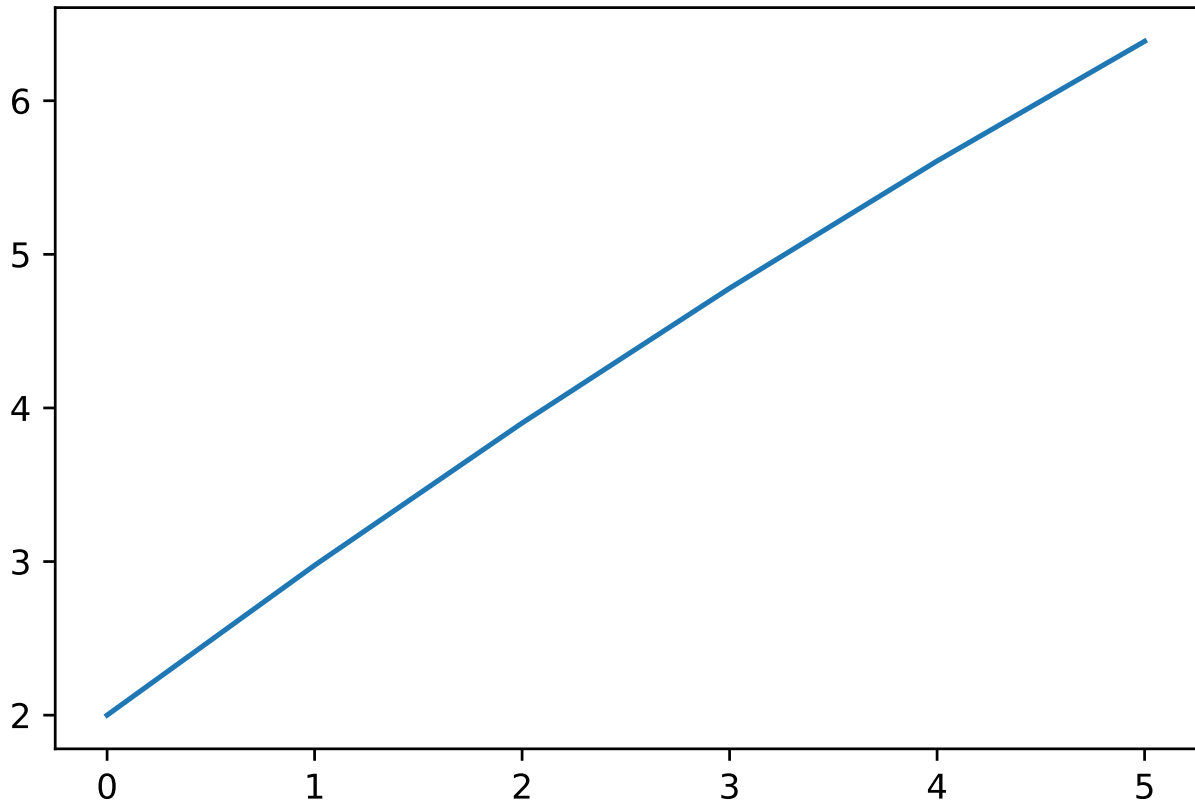
    theta = theta*np.pi/180

    return x*np.tan(theta) \
        - g*x**2 / (2*v0**2 * np.cos(theta)**2) + y0 # Hoehe des Balls

xliste = [0., 1., 2., 3., 4., 5.] # erzeugt Liste von x-Werten
yvalues = [y(x) for x in xliste] # erzeugt Liste zugehöriger y-Werte

plt.plot(xliste,yvalues)
```

```
[<matplotlib.lines.Line2D at 0x1146f5080>]
```



5.2 2. Variante: Wir erzeugen Arrays unter Verwendung von linspace

```
import numpy as np
import matplotlib.pyplot as plt

def y(x):
    import numpy as np

    g = 9.81      # Gravitationsbeschleunigung
    v0 = 20.     # Anfangsgeschwindigkeit
    theta = 45.  # Winkel mit x-Achse (in Grad)
    y0 = 2.     # Anfangsposition des Balls ist (0,y0)

    theta = theta*np.pi/180

    return x*np.tan(theta) \
        - g*x**2 / (2*v0**2 * np.cos(theta)**2) + y0 # Hoehe des Balls

xx = np.linspace(0,42,100);           # erzeugt Array mit 100 gleichmig
                                       # verteilten Werten zwischen 0 und 42
yy = y(xx)                             # erzeugt Array der y-Werte
plt.plot(xx,yy)
```

```
| [<matplotlib.lines.Line2D at 0x114eb38>]
```

