

```

from turtle import *
def pfeilSpitze(stufe, orientierung, laenge):
    if stufe > 0 and laenge > 1:
        lt(60 * orientierung)
        pfeilSpitze(stufe - 1, -orientierung, laenge/2)
        lt(-60 * orientierung)
        pfeilSpitze(stufe - 1, orientierung, laenge/2)
        lt(-60 * orientierung)
        pfeilSpitze(stufe - 1, -orientierung, laenge/2)
        lt(60 * orientierung)
    else:
        fd(laenge)

pfeilSpitze(4, 1, 100)

```

```

from turtle import *
from math import *
def drachen(stufe, orientierung, laenge):
    if stufe > 0 and laenge>1:
        lt(orientierung * 45)
        drachen(stufe - 1, -1, laenge/sqrt(2))
        lt(-orientierung * 90)
        drachen(stufe - 1, +1, laenge/sqrt(2))
        lt(orientierung * 45)
    else:
        fd(laenge)

drachen(5,1,200)

```

```

from turtle import *
def hilbert(stufe, orientierung, laenge):
    if stufe > 0:
        lt(-orientierung * 90)
        hilbert(stufe - 1, -orientierung, laenge)
        fd(laenge)
        lt(orientierung * 90)
        hilbert(stufe - 1, orientierung, laenge)
        fd(laenge)
        hilbert(stufe - 1, orientierung, laenge)
        lt(orientierung * 90)
        fd(laenge)
        hilbert(stufe - 1, -orientierung, laenge)
        lt(-orientierung * 90)

hilbert(3, 1, 10)

```

```
from turtle import *
```

```
def pythBaum(stufe, laenge):  
    alpha = 36.86989765  
    beta = 53.13010235  
    if stufe == 0:  
        fd(laenge)  
    else:  
        fd(laenge)  
        lt(90)  
        fd(laenge)  
        lt(90)  
        fd(laenge)  
        lt(90)  
        fd(laenge)  
        lt(180)  
        fd(laenge)  
        lt(-beta)  
        if stufe > 0:  
            pythBaum(stufe - 1, 0.8 * laenge)  
        else:  
            fd(0.8 * laenge)  
        lt(-90)  
        if stufe > 0:  
            pythBaum(stufe - 1, 0.6 * laenge)  
        else:  
            fd(0.6 * laenge)  
        lt(-alpha)  
        fd(laenge)  
        lt(90)
```

```
pythBaum(4, 50)
```

```

from tkinter import *
from math import *

master = Tk( )
w = Canvas(master, width=600, height=400, bg='gray')
w.pack( )

winkel = 90
x_old = 300
y_old = 200

def move(entfernung):
    global x_old, y_old
    radiant = -1*pi/180*winkel
    x_new = x_old + entfernung * cos(radiant)
    y_new = y_old + entfernung * sin(radiant)
    w.create_line(x_old,y_old,x_new,y_new, fill="black", width=1)
    x_old = x_new
    y_old = y_new

def turn(Drehwinkel):
    global winkel
    winkel = winkel + Drehwinkel
    if winkel >= 360:
        winkel = winkel-360
    if winkel <= -360:
        winkel = winkel+360

move(100)
turn(90)
move(100)
turn(-45)
move(100)

```