



Programmeren in Python – Deel 4/5

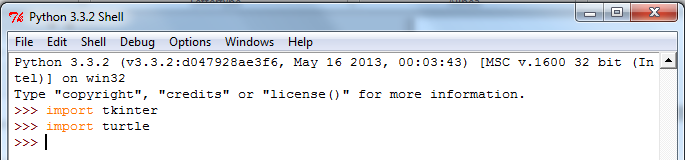
Bert Smits voor

Basisschool “De Esdoorn”  
Bankstraat 29  
2811 Hombeek (Mechelen)

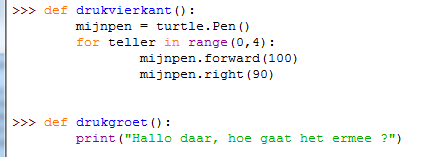
1. Vandaag gaan we nog even verder op het gebruik van als…dan beslissingen, maar dan met behulp van een nieuwe module met knoppen en vensters op het scherm. Met een duur woord noemen we dat een “Grafische User Interface”, een GUI (je spreekt het uit als GOEWIE). “User” is Engels voor “Gebruiker”, en dus is een GUI een grafische manier (met drukknopjes, vensters en wat nog meer) om de gebruiker met het programma te laten werken.

Het eenvoudigste voorbeeld is een drukknopje dat, als je er met de muis op klikt, een boodschap op het scherm afdrukt.

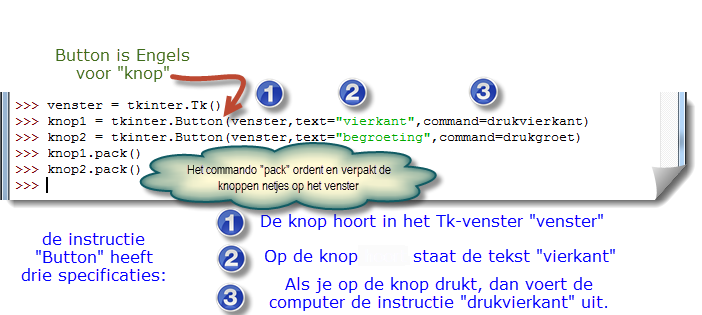
De module die we gaan gebruiken heet “tkinter”. We gebruiken tegelijk ook de “turtle” module, dus we voeren in:



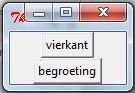
Vervolgens maken we de functie(s) die we willen koppelen aan onze knop:



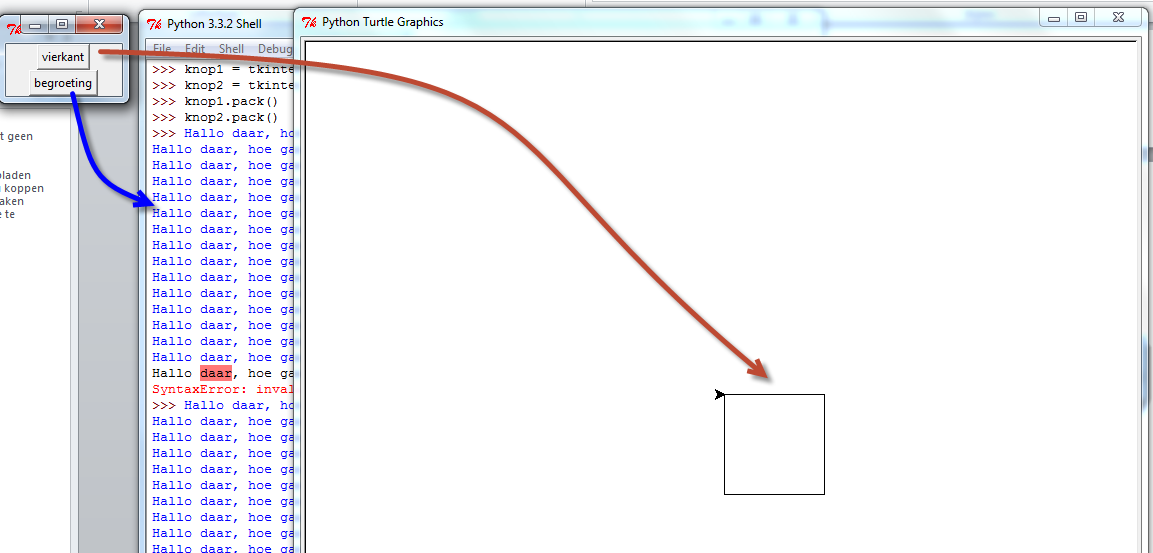
En dan kunnen we een venster maken met twee knoppen, ééntje gaat “Hallo daar, hoe gaat het ermee” afdrukken; een andere knop gaat, als je erop drukt, een vierkant van zijde 100 met turtle afdrukken op het scherm.



Na het uitvoeren van de pack-instructie verschijnt er een venster met twee knoppen :



En als je op de knoppen drukt, gebeurt er het volgende:

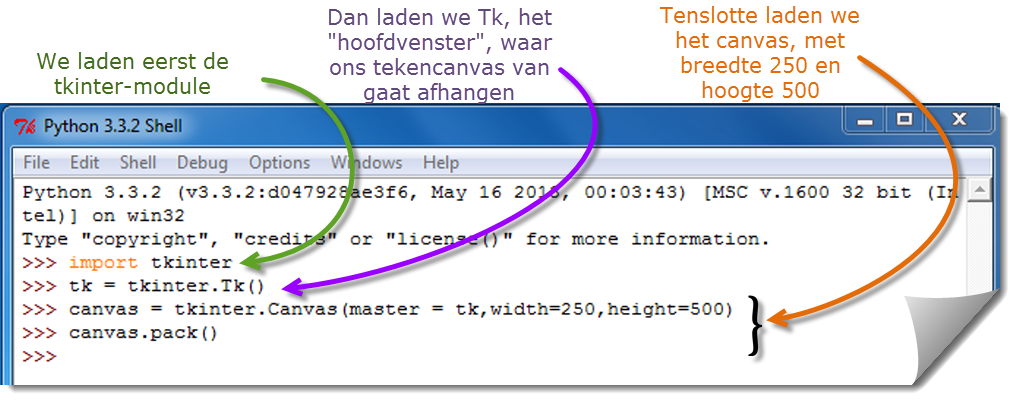


1. Er zijn veel verschillende mogelijkheden als grafische user interface, naast de eenvoudige knop.

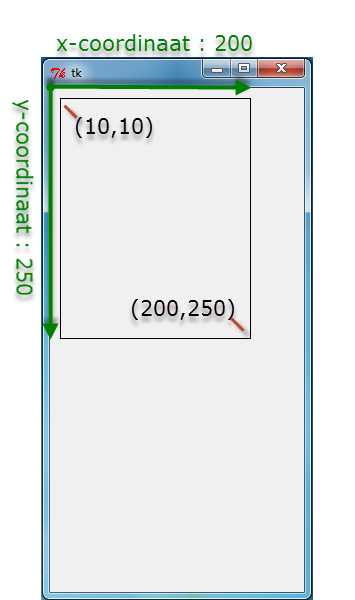
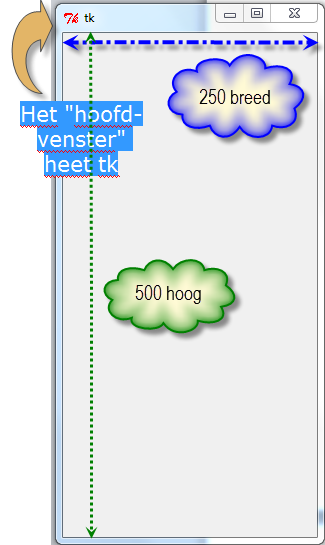
Zo kan je met verschillende vensters werken, in allerlei kleurtjes, met knoppen, menu’s, keuzerondjes, enz… en die allemaal linken aan je programma.

Een “canvas” is eigenlijk een tekendoek, die de kunstschilder dan op een schildersezel plaatst om bijvoorbeeld een olieverfschilderij te maken (zie ook het plaatje). Maar in een GUI wordt de naam “canvas” gebruikt om een plek op het scherm aan te duiden waar op getekend kan worden.

De code in Python gaat als volgt:



Daarmee verschijnt dan een (leeg) canvasvenster, met van grootte 250 bij 500:



Vervolgens kan je allerlei figuurtjes tekenen op het canvas. Dit kan aan de hand van een waslijst van functies die we op het canvas kunnen aanroepen. Om een rechthoek te tekenen bijvoorbeeld, doe je

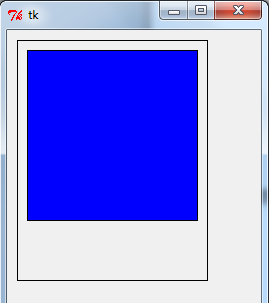
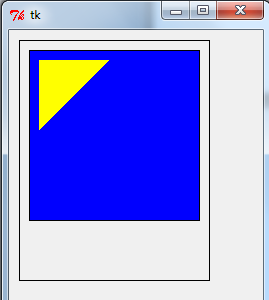
canvas.create\_rectangle(10,10,150,200)

De getallen tussen de haakjes zijn de coördinaten van de uiterste punten van de rechthoek, zoals op de rechterfiguur is aangegeven.

Je kan de figuren ook kleuren, met bijvoorbeeld

canvas.create\_rectangle(20,20,190,190,fill=”blue”)

teken je een blauw vierkant (binnen in de rechthoek die er al stond):



En je hoeft je natuurlijk niet te beperken tot rechthoeken of vierkanten. Een veelhoek (Engels:”polygon”) teken je gewoon met de coördinaten van de opeenvolgende hoekpunten. De gele driehoek rechts is er gekomen via

canvas.create\_polygon(30,30,100,30,30,100,fill=”yellow”)

En je kan ook tekst laten verschijnen met

canvas.create\_text(120,120,text=”7 x 7 = 49”,fill=”white”)

We kiezen wit als vulkleur om goed leesbaar te zijn op de blauwe achtergrond. Tenslotte kan je ook foto’s op een canvas laden. Dat gaat dan zo:

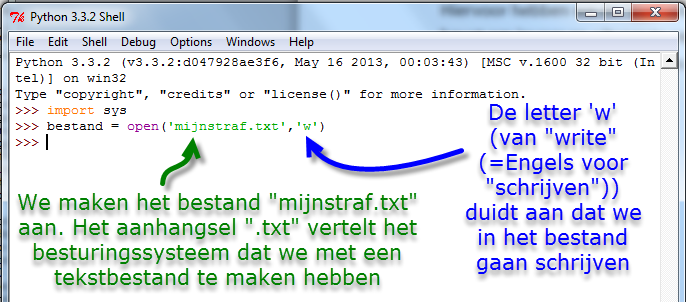
 

1. Tenslotte gaan we leren hoe je je werk kan wegschrijven naar een bestand op de computer, dat dan kan afgedrukt worden, of door een ander programma gebruikt kan worden.

Stel dat we van meester Guido straf gekregen hebben en dat we 100 keer moeten schrijven “Ik mag niet in de gang lopen”. Je herinnert je nog het programmaatje in de schelp dat we zo in de eerste les gemaakt hebben. Maar dat daar hadden we niet zoveel aan, omdat het resultaat op het scherm werd afgedrukt.

We kunnen het resultaat evengoed wegschrijven naar een bestand. Hiervoor hebben we de “**sys**” module nodig, die een heleboel functies bevat om invoer en uitvoer te regelen (Herinner je in de vorige les : het was ook een sys-module die ons toeliet om invoer op het scherm te kunnen inlezen met **readline**).

We beginnen dus met



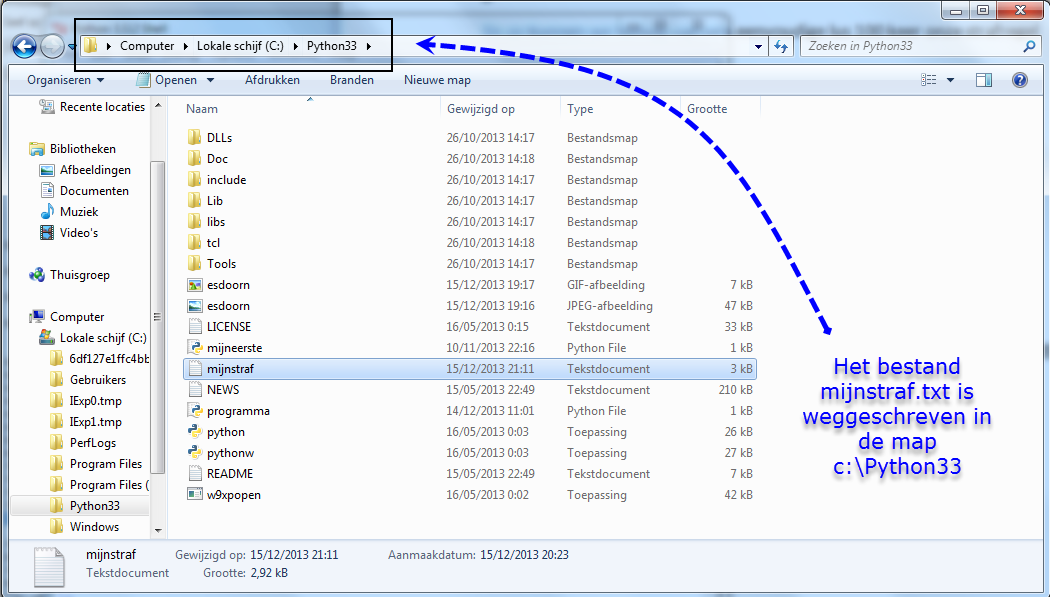
En zo kunnen we behulp van een eenvoudige lus 100 keer onze strafregel in het bestand schrijven :

for teller in range(0,100):

bestand.write(“Ik mag niet in de gang lopen\n”)

bestand.close()

Als we nu gaan kijken in de mappen op de computer, dan vinden we het aangemaakte bestand “mijnstraf.txt” terug :



Als we het openen, dan zien we hoe het zinnetje er netjes 100 keer is in weggeschreven:



**En wat hebben we vandaag geleerd ?**

**Ten eerste** : Python beschikt over verschillende mogelijkheden om via tekeningen en knoppen op het scherm invoer en uitvoer te laten verlopen. Dit noemt men een grafische user interface of een GUI (spreek uit GOEWIE).

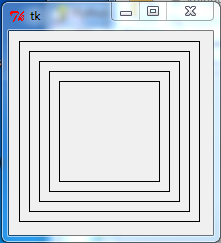
**Ten tweede** : met behulp van de module **tkinter** kan je, sneller dan met de schildpad, allerlei figuren op het scherm tekenen, en zelfs foto’s bewerken.

**Ten derde** : Uitvoer kan ook naar een bestand. De module “sys” bevat functies om bestanden aan te maken en de resultaten van berekeningen en programma’s weg te schrijven.

Woordenlijst

|  |  |  |
| --- | --- | --- |
| Engels | Nederlands | Commentaar |
| Graphical User Interface | Grafische Gebruikersomgeving | Afgekort : GUI (spreek uit GOEWIE) |
| Button | Knop |  |
| Pack | Inpakken | Alle grafische vensters en objecten worden met de instructie “pack()” juist op het scherm geplaatst |
| Rectangle | Rechthoek | create\_rectangle(..) verwacht 2 x 2 coördinaten |
| Polygon | Veelhoek | create\_polygon(..) verwacht 2 coördinaten per hoekpunt |
| Write | Schrijven |  |
| Open | Openen |  |
| Close | Sluiten |  |

1. Programmeerpuzzels

Schrijf de volgende programmaatjes in Python:

1. Een programma dat een vierkanten canvas aanmaakt (zijde 200) en daar dan vijf vierkanten intekent zoals op de figuur.
2. Een programma dat alle tafels van 1 tot 10 wegschrijft in een bestand “tafels.txt”. m.a.w.

1 x 1 = 1  
1 x 2 = 2  
enz…  
10 x 10 = 100

1. Naschrift

Lieve Ouders,

Het was mij een genoegen om jullie zoon of dochter een inleiding in het programmeren te mogen geven. Uiteindelijk hebben we gezien

* Hoe een programma werkt
* Dat een computer kan rekenen en tekenen
* Wat je kan doen met lussen en als…dan… bewerkingen
* Hoe je je werk kan wegschrijven naar een bestand

Natuurlijk kun je in 5 lestijden moeilijk een volledige programmeertaal zien. Dat lukt bij volwassenen trouwens ook niet. Maar ik heb gezien hoe de kinderen spontaan begonnen te experimenteren\* en dat is veelbelovend voor de toekomst.

U zou uw kind – als jullie daar allebei zin in hebben – zelf verder kunnen begeleiden. Zelf heb ik veel gehad aan het boekje “Python for Kids”, a playful introduction to programming, door Jason R. Briggs (twee andere referenties waren minder nuttig).



Python is een volledige, object-georiënteerde programmeertaal, met ondersteuning voor zowat alle gebruikelijke toepassingen (internet, grafische user interfaces, multimedia, wiskunde, enz…).

En door te programmeren leert uw kind natuurlijk logisch redeneren, methodisch te werk gaan en een zekere robuustheid in het overkomen van technische problemen (ik heb hen geleerd dat computers DOM zijn, en dat je hen héél precies moet vertellen wat ze moeten doen).

Bert Smits  
Hombeek, 15 December 2013  
  
*\*Gebruiken van andere variabele-namen, een lus 1000 keer laten uitvoeren in plaats van 100 keer, enz…*