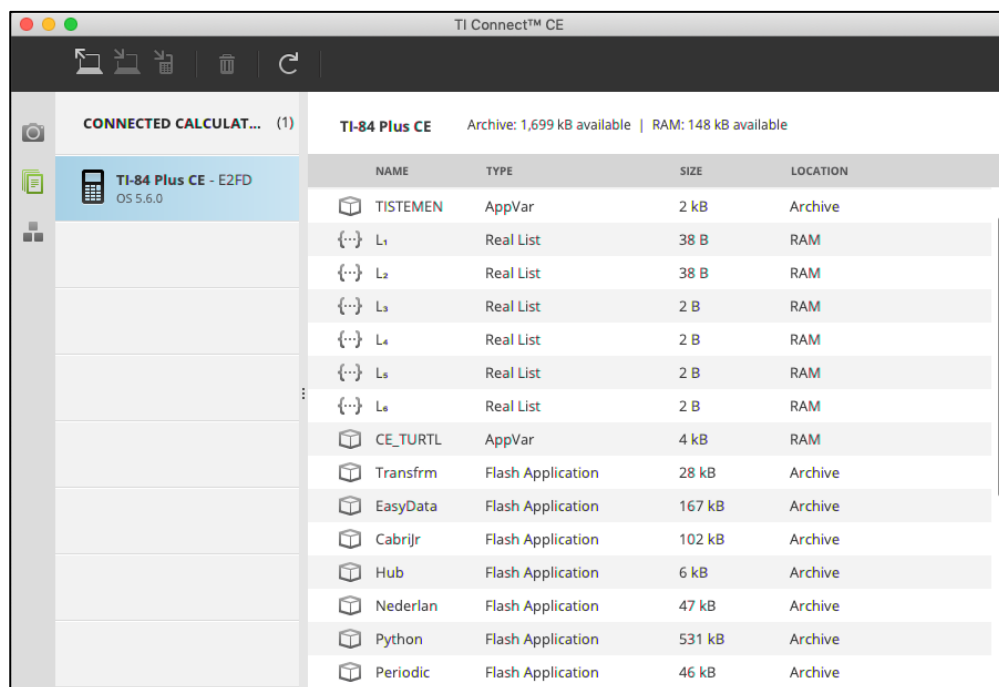


De TI Python-module Turtle visualiseert het programmeren in Python op een zeer eenvoudige manier. Met de Turtle-functionaliteit teken je op een makkelijke manier lijnen, vierkanten, cirkels, ...

Turtle biedt je handige toolbox om in combinatie met de buit-in Python-functie voor het maken van een allerlei tekeningen en het creatief om te gaan met graphics.

## 1. Installatie van de TURTLE-Module

Kopieer het bestand CE\_TURTL.8xv naar het geheugen van een TI-84 Plus CE-T Python Edition, gebruikmakend van TI Connect™ CE.

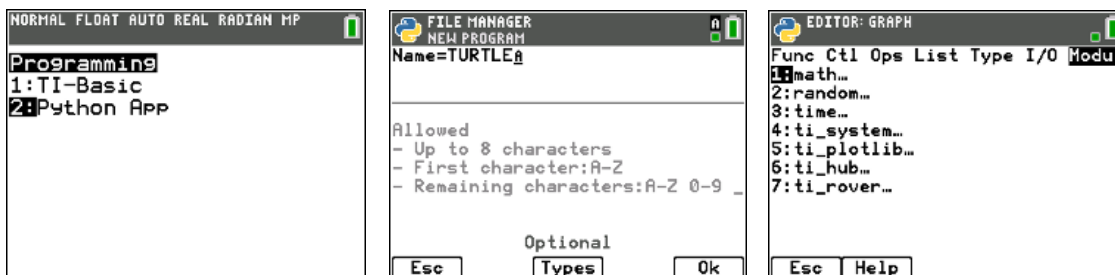


CE\_TURTL zal als aa App variabele in het geheugen geplaatst worden.

Voor het gebruik van de TURTLE-module in de TI-SmartView™ CE-T software kopieer dit bestand naar de Emulator Explorer workspace.

## 2. Activeren Turtle-module

- Start de Python-App, **prgm**, en creër een nieuw Python programma TURTLE. Op dit moment is de Turtle-module nog niet beschikbaar in het module-menu.



b. Selecteer de code **from PROGRAM import \*** uit de catalog.

```
EDITOR: TURTLE
CATALOG
>#
%      modulo;remainder
//     integer divide;floor
[a A #]
a      gradient; slope
abs()
acos()
and
.append(x)
as
asin()
Esc
```

```
EDITOR: TURTLE
CATALOG
>#
%      modulo;remainder
//     integer divide;floor
[a A #]
a      gradient; slope
abs()
acos()
and
.append(x)
as
asin()
Esc
```

```
EDITOR: TURTLE
CATALOG
for i in range(size):
for i in range(start,stop):
for i in range(strt,stp,step):
str.format()      string format
frexp()
from PROGRAM import *
from math import *
from random import *
from time import *
from ti_system import *
from ti_hub import *
Esc
```

c. Vul de code aan met ce\_turtle. Het liggend streepje is beschikbaar via **a A #**.

```
EDITOR: TURTLE
PROGRAM LINE 0001
from _import *
Fns... a A # Tools Run Files
```

```
EDITOR: TURTLE
PROGRAM LINE 0002
from ce_turtl import *
Fns... a A # Tools Run Files
```

```
EDITOR: TURTLE
PROGRAM LINE 0001
--
# " ' : , ; . ! ? [ \ _
a b c d e f g h i j k l m
n o p q r s t u v w x y z

() [] {} * ** % //
= == != < <= > >=
and or not True False
<< >> & | ^ ~
Esc Select Paste
```

d. De module Turtle is nu toegevoegd aan het module-menu.

```
EDITOR: TURTLE
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:ce_turtl...
Esc Help
```

e. De module Turtle beschikt over drie submenu's met Python-statements: Turtle, Draw en Properties.

```
EDITOR: TURTLE
Turtle Draw Properties
1:turtle.home()
2:turtle.penup()
3:turtle.pendown()
4:turtle.clear()
5:turtle.show()
Esc Modul
```

```
EDITOR: TURTLE
Turtle Draw Properties
1:turtle.forward(pixels)
2:turtle.backward(pixels)
3:turtle.right(degrees)
4:turtle.left(degrees)
5:turtle.goto(x,y)
6:turtle.circle(radius)
7:turtle.dot(radius)
Esc Modul
```

```
EDITOR: TURTLE
Turtle Draw Properties
1:turtle.heading()
2:turtle.setheading(angle)
3:turtle.position()
4:turtle.color(r,g,b)
5:turtle.pensize(size 0,1,2)
6:turtle.speed(speed 0,1)
7:turtle.isdown()
Esc Modul
```

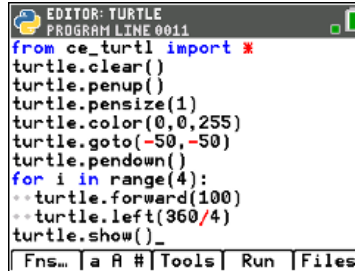
### 3. Twee voorbeelden

#### 3.1. Tekenen van een vierkant

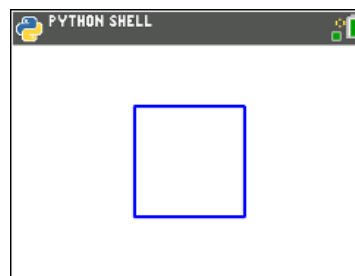
```
from ce_turtl import *
```

```
turtle.clear()
turtle.penup()
turtle.pensize(1)
turtle.color(0,0,255)
turtle.goto(-50,-50)
turtle.pendown()
```

```
for i in range(4):
    ♦♦ turtle.forward(100)
    ♦♦ turtle.left(360/4)
turtle.show()
```



```
EDITOR: TURTLE
PROGRAM LINE 0011
from ce_turtl import *
turtle.clear()
turtle.penup()
turtle.pensize(1)
turtle.color(0,0,255)
turtle.goto(-50,-50)
turtle.pendown()
for i in range(4):
    .. turtle.forward(100)
    .. turtle.left(360/4)
turtle.show()_
Fns... | a A # | Tools | Run | Files
```



#### 3.2. Koch-kromme

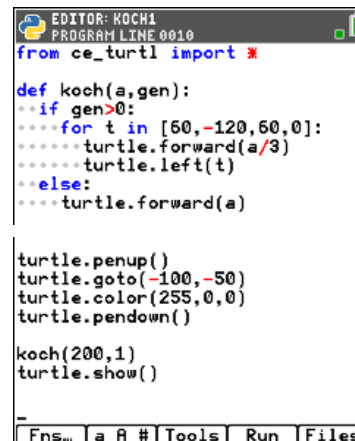
##### Generatie 1

```
from ce_turtl import *
```

```
def koch(a,gen):
    ♦♦ if gen>0:
        ♦♦♦♦ for t in [60,-120,60,0]:
            ♦♦♦♦♦ turtle.forward(a/3)
            ♦♦♦♦♦ turtle.left(t)
        ♦♦ else:
            ♦♦♦♦ turtle.forward(a)
```

```
turtle.penup()
turtle.goto(-100,-50)
turtle.color(255,0,0)
turtle.pendown()
```

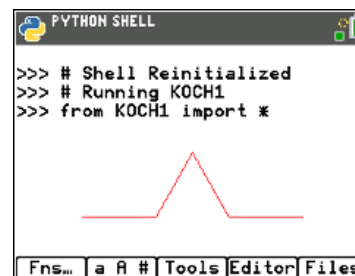
```
koch(200,1)
turtle.show()
```



```
EDITOR: KOCH1
PROGRAM LINE 0010
from ce_turtl import *
def koch(a,gen):
    .. if gen>0:
        .. for t in [60,-120,60,0]:
            .. turtle.forward(a/3)
            .. turtle.left(t)
        .. else:
            .. turtle.forward(a)

turtle.penup()
turtle.goto(-100,-50)
turtle.color(255,0,0)
turtle.pendown()

koch(200,1)
turtle.show()
_
Fns... | a A # | Tools | Run | Files
```



**Recursie**

```
from ce_turtl import *

def koch(a,gen):
    ♦♦ if gen>0:
        ♦♦♦♦ for t in [60,-120,60,0]:
            ♦♦♦♦♦♦ koch(a/3,gen-1)
            ♦♦♦♦♦♦ turtle.left(t)
        ♦♦ else:
            ♦♦♦♦ turtle.forward(a)

turtle.penup()
turtle.goto(-100,-50)
turtle.pendown()
turtle.pensize(0)
turtle.color(255,0,0)

koch(200,4)
turtle.show()
```

```
EDITOR: KOCH
PROGRAM LINE 0011
from ce_turtl import *

def koch(a,gen):
    ♦♦ if gen>0:
        ♦♦♦♦ for t in [60,-120,60,0]:
            ♦♦♦♦♦♦ koch(a/3,gen-1)
            ♦♦♦♦♦♦ turtle.left(t)
        ♦♦ else:
            ♦♦♦♦ turtle.forward(a)

turtle.penup()
turtle.goto(-100,-50)
turtle.pendown()
turtle.pensize(0)
turtle.color(255,0,0)
koch(200,4)
turtle.show()
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running KOCH
>>> from KOCH import *
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running KOCH
>>> from KOCH import *
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running KOCH
>>> from KOCH import *
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running KOCH
>>> from KOCH import *
```