



**Unit 4 : Het gebruik van de ti\_plotlib module**

**Oefenblad 1 : Het grafische scherm**

In deze unit maken we kennis met de ti\_plotlib module waarmee je grafische voorstellingen kunt maken

**Doelen :**

- De plotlib module verkennen.
- Het scherm configureren en punten tekenen.

Om in Python grafische voorstellingen te maken is de ti\_plotlib module nodig. In deze module vind je opdrachten voor het tekenen van punten en lijnen en voor het configureren van het grafische scherm.

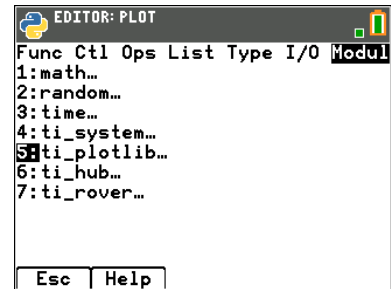
Start een nieuw Python programma.

Voeg nu eerst de ti\_plotlib module in met bijvoorbeeld » en dan 5:ti\_plotlib.

Kies voor 1:import ti\_plotlib as plt.

Plot opdrachten moeten nu altijd voorafgegaan worden door plt.

Dit is een eigenschap van Python als je een module op deze manier invoegt.



We beginnen het programma met een leeg scherm. Daarvoor is de opdracht cls() die je kunt vinden in het ti\_plotlib menu (»). Het voorvoegsel plt. wordt automatisch ingevoegd.



We gaan het punt (2,3) tekenen.

Daarvoor gaan we weer naar het ti\_plotlib menu en nu naar het tweede tabblad (Draw). Daarin staat optie 6:plot().

Als je dit kiest wordt eerst gevraagd welk symbool je wilt gebruiken.

Kies optie 1:o. Dit zorgt straks voor een grote stip op het scherm.

Daarna kun je de coördinaten van het punt invullen, in ons geval (2,3).

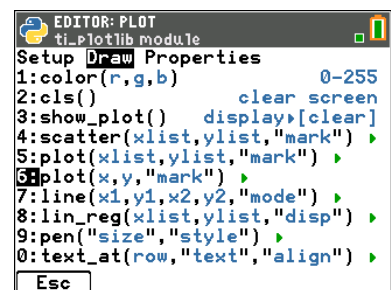
Als laatste moeten we nog een opdracht toevoegen: show\_plot().

(Weer te vinden in het plot-menu)

Deze laatste opdracht zorgt ervoor dat alle plot opdrachten worden uitgevoerd in een grafisch scherm.

Als het goed is zie je, nadat je het programma hebt uitgevoerd (met Run), een stip op het scherm.

Je kunt het grafisch scherm weer verlaten met de ` toets.





We passen het programma aan zodat het scherm er wat mooier uitziet.

We voegen rooster en een assenstelsel toe.

Zie hiernaast:

De meeste functies kun je vinden in het plot-menu onder het tabblad Setup.

De functie `plt.color()` staat onder het tabblad Draw.

De drie getallen staan voor rood, groen en blauw.

De volgorde is belangrijk: elke volgende tekenopdracht tekent over de vorige opdracht heen.

```
EDITOR: PLOT
PROGRAM LINE 0009
import ti_plotlib as plt
plt.cls()
plt.window(-10,10,-10,10)
plt.grid(1,1,"dot")
plt.axes("on")
plt.color("on")
plt.color(0,0,255)
plt.plot(2,3,"o")
plt.show_plot()
```

**Tip voor de docent:** De kleur van een punt of een lijn kun je wijzigen met de functie `color`.

Deze functie vraagt drie argumenten. Dit zijn de RGB-waardes van de kleur (Rood, Groen en Blauw).

Deze waarden moeten liggen in het interval [0-255].

Bij de functie `grid()` kan nog een extra argument voor de kleur worden meegegeven: `plt.grid(xsc1,ysc1,"stijl",(r,g,b))`.

Hiernaast zie je dat er nog enkele regels aan het programma zijn toegevoegd.

Bedenk voordat je het programma uitvoert wat het effect zal zijn.

```
EDITOR: PLOT
PROGRAM LINE 0011
import ti_plotlib as plt
plt.cls()
plt.window(-10,10,-10,10)
plt.grid(1,1,"dot")
plt.axes("on")
plt.color(0,0,255)
plt.plot(2,3,"o")
plt.color(255,0,0)
plt.line(2,3,8,6,"")
plt.show_plot()
```

Voer het programma uit.

Je krijgt dan de figuur hiernaast.

