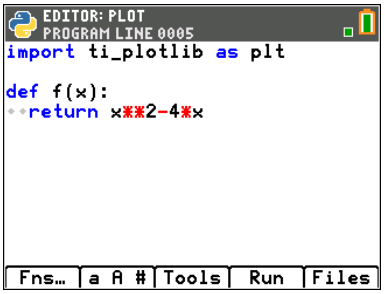
Unit 4 : Het gebruik van de ti_plotlib module | **Oefenblad 2: Grafieken tekenen**

In deze unit maken we kennis met de ti_plotlib module waarmee je grafische voorstellingen kan maken

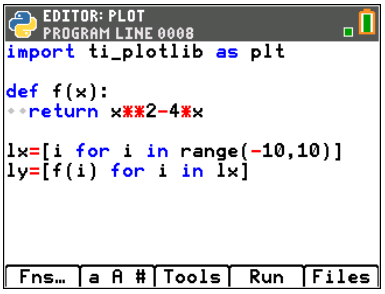
Doelen :

- De ti_plotlib module gebruiken.
- De grafiek van een functie tekenen.

In het vorige oefenblad gebruikten we de functie plot(x,y,"...") om een enkel punt te tekenen waarbij x en y de coördinaten zijn van het punt.
Als je meerdere punten wilt tekenen (voor bijvoorbeeld een grafiek van een functie) kun je voor de x en de y ook een **lijst** met coördinaten invullen.
Als lx de lijst met x-coördinaten is en ly de lijst met de bijbehorende y-coördinaten, dan kun je met de functie plot(lx,ly,"...") deze punten tekenen waarbij de punten onderling verbonden worden.

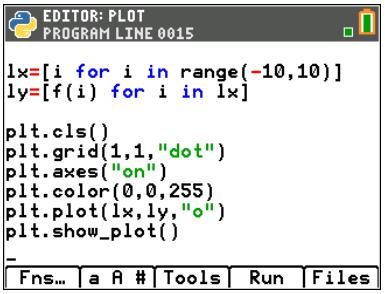


We gaan de functie f(x)=x^2 – 4x plotten.
Start een nieuw Python programma en voeg de ti_plotlib module in.
We definiëren eerst de functie f.
Vervolgens maken we een lijst met x-coördinaten (lx).
En daarna een lijst met de bijbehorende y-coördinaten (ly).

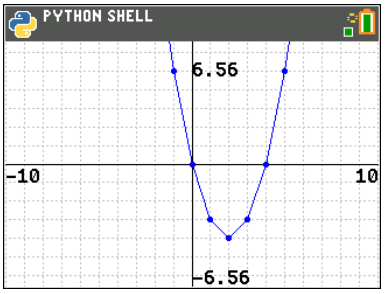


Tip voor de docent: De lijsten worden gemaakt m.b.v. een **for**-lus.
Je kunt deze vinden met F1(Fns...) en dan **Ctl**.
Bij de tweede lijst wordt de optie gebruikt waarbij de waarde van i niet steeds wordt opgehoogd, maar i doorloopt de waarden in de lijst lx.

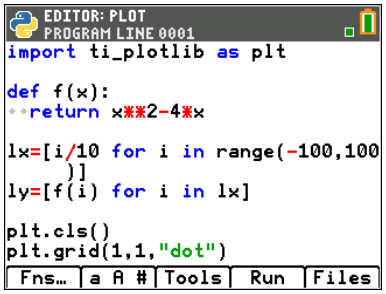
We maken het programma af met dezelfde opdrachten als in het vorige voorbeeld.
Merk op dat er nu bij de plot() opdracht twee lijsten zijn ingevuld.
Als je het programma uitvoert zie je een grafiek van de functie f.
Het ziet er hoekig uit, maar dat komt omdat we voor x alleen hele waarden hebben gebruikt.



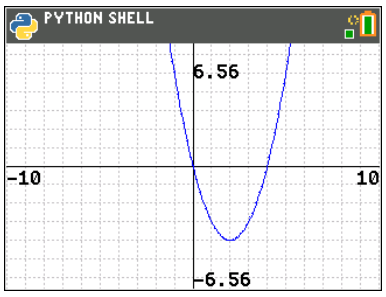
We gaan het aanpassen zodat het er wat vloeiender gaat uitzien.



We willen voor de x-coördinaten niet alleen hele getallen gebruiken.
Daarom passen we die lijst aan.
Stel bijvoorbeeld dat we in plaats van hele stappen stapjes willen nemen van 0.1.
Dan kunnen we de lijst aanpassen zoals hiernaast.
Let er wel op dat ook de range() getallen moeten worden aangepast.
Als i bij -100 begint willen voor de eerste x-coördinaat er -10 uit hebben.



Als je ook nog de plot() opdracht aanpast met een andere puntgrootte krijg je de grafiek zoals hiernaast.



Tip voor de docent: De plot opdracht ziet er nu als volgt uit: **plt.plt(lx,ly,".")**
In plaats van **i/10** in de lx-lijst kun je ook nemen **i/2** met **range(-20,20)**.
De grafiek ziet er dan aardig uit terwijl de tekensnelheid hoger is.