Module 5: Plaatjes — **Oefenblad 3: Lijnen, Tekst en Kleuren**

In deze derde les van module 5 leer je over het tekenen van lijnen, tekst en het verrijken met kleur.

Doelen:

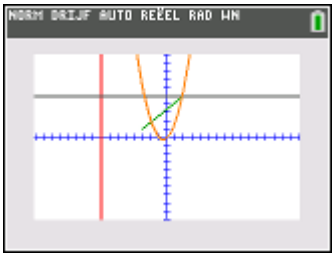
- De lijn, functie en tekst tekenopdrachten gebruiken.
- Kleur gebruiken in grafische opdrachten.
- Formules maken om gebruik te maken van plaatjes in programma's.

Lijnen en krommen tekenen

Lijn(X,Y,W,Z) tekent een *lijnstuk* tussen de punten (X,Y) en (W,Z). Zie Help van de catalogus voor de optionele kenmerken.
Verticaal A tekent de verticale lijn X=A.
Horizontaal B tekent de horizontale lijn Y=B.
TekenF X^2+X tekent de functie. Dit is anders dan het tekenen van de grafiek van de functie.

Zie de voorbeelden rechts. Merk op dat de optionele kleueren te vinden zijn in het menu kleur van PRGM. Kleur is niet beschikbaar op de TI-84 Plus.

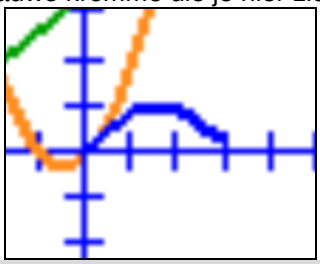
Dit programma doet dit:



Tip: om een deel van de functie te tekenen deel je deze in stukken met behulp van het gewenste interval:

TekenF sin(X)/(X≥0 en X≤π)

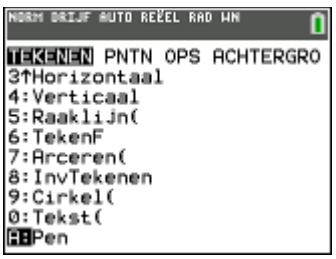
Tekent de blauwe kromme die je hier ziet (waarom?)



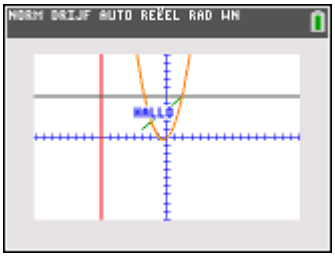
Docenten Tip: De **TekenF** functie is niet hetzelfde als het tekenen van de grafiek van een functie met de! en %. Een functie die is getekend kan niet worden gevolgd, ook kan er niet mee worden 'gerekend' (è). De instructie **TekenF** tekent alleen de pixels die de functie bepaalt. Het opnieuw tekenen van het scherm zal elke getekende functie wissen. Deze zal niet opnieuw worden getekend tenzij de instructie opnieuw gegeven wordt.
Mark op dat de opdracht **Line( )** alleen een lijnstuk tekent. Maar **Verticaal** en **Horizontaal** tekenen volledige lijnen van de ene rand naar de andere rand van het scherm. Dit si een belangrijk moment om iets uit te leggen! Er volgt een project waarin de leerlingen een volledige lijn programmeren en niet alleen een lijnstuk.

Tekst tekenen

De tekenopdracht **Tekst(** is uniek omdat het de pixelwaarden gebruikt en niet de vensterwaarden (punten) voor het positioneren van de tekst. Er bestaat ook een aparte opdracht **TekstKleur(** die de kleur instelt voor de volgende tekst die wordt getekend.



Tekst(50,100,"HALLO") geeft HALLO weer op dezelfde plek op het scherm ongeacht de vensterinstellingen. Rij 50, kolom 100 van de pixels stellen de *linkerbovenhoek* voor van de tekst die getekend wordt.



Opmerking: Denk aan de pixel-afmetingen van jouw scherm: TI-84 Plus: 96 kolommen x 64 rijen en TI-84 Plus C/CE: 265 kolommen x 165 rijen.

Docenten Tip: Dit is een verraderlijke opdracht omdat het de enige van de tekeninstructies die *pixel* coördinaten gebruikt in plaats van venstercoördinaten. Hieronder wordt een interessant project besproken..

Programmeren met Lijn(en algebra

Deze programmeeractiviteit versterkt de opdracht **Lijn(**.
De opdracht **Lijn(** tekent slechts een *lijnstuk* tussen twee punten. Wij zouden gaar een *lijn* zien door de twee punten die helemaal doorloopt over het scherm ongeacht welke twee punten zijn geselecteerd. In deze activiteit wordt gebruik gemaakt van begrippen uit e algebra, dus wees voorbereid!

1. Start een programma. We zullen het **LIJN** noemen.
2. Voeg de gebruikelijke opdrachten om de grafiek in te stellen toe aan het begin van het programma.
3. Gebruik twee **Input**-opdrachten *zonder variabelen* om de coördinaten van twee punten op het scherm te krijgen. Input bepaalt de waarden van **X** en **Y** dus we moeten de eerste twee waarden opslaan in andere variabelen **A** en **B** zodat we het tweede paar coördinaten kunnen krijgen in **X** en **Y**.
4. Bereken de *helling* **M** (rico) van deze lijn en sla die op. `:(Y-B)/(X-A)→M`
5. Nu hebben we de twee punten nodig aan de linker- en rechterkant van het scherm voor de lijn-opdracht. De *x*-coördinaten van deze punten zijn **Xmin** en **Xmax**.
6. We moeten nu de *y*-coördinaten berekenen.
7. De vergelijking van de lijn is ***y* = M*(*x* - A) + B**.

Jouw opdracht ……

1. Vul **Xmin** en **Xmax** (de *namen* niet de waarden!) in de vergelijking in voor *x* en sla het resultaat op in de variabelen **Q** en **R** die de *y*-coördinaten voorstellen.

   Antwoord: M*(Xmin-A)+B → Q
   M*(Xmax-A)+B → R

2. Gebruik de opdracht **Lijn(** om een lijn te tekenen tussen de linker- en rechterkant van het scherm. `:Line(Xmin,Q,Xmax,R)`

Uitbreidingen

1. Voeg een lus toe in het programma om het mogelijk te maken veel lijnen te tekenen zonder het programma opnieuw te hoeven uitvoeren, wat het scherm leeg maakt.
2. Dit programma werkt niet wanneer de lijn verticaal loopt. Waarom? Neem een **If**-structuur op om dit speciale geval af te handelen.

Docenten Tip: Omdat de helling (rico) van de lijn niet gedefinieerd is. Het programma probeert om door nul te delen, wat een fout veroorzaakt.

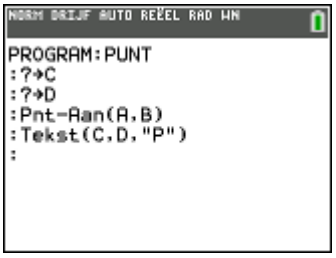
Punten naar pixels programmeren

Stel je voor: je gebruikt de opdracht **Pnt-Aan(** om een punt (**A**,**B**) te tekenen op het grafiekscherm. Je wilt dit punt nu een *label* geven met tekst. Waar moet he deze tekst tekenen?

Schrijf twee formules (een voor **C** en een voor **D**) die venstercoördinaten omzetten in pixelcoördinaten voor de opdracht **Tekst(**. Deze tabel (TI-84 C/CE waarden) kan misschien helpen:

VENSTER	pixel
Xmin	0
Xmax	264
A	**?**
Ymax	0
Ymin	164
B	**?**

Maak het **?** *af zodat het punt gelabeld wordt met* P:



Opmerking: denk eraan dat in de instructie ***Tekst(*** *het eerste argument het RIJ-nummer is, dat correspondeert met de y-coördinaat van dit punt!*

Docenten Tip: Dit is een ander voorbeeld over het gebruik van een lineaire verband. Denk aan (Xmin, 0) en (Xmax, 264) als twee punten op een lijn.

De helling (rico) van de lijn is dan

(264-0)/(Xmax-Xmin)

dus de lineaire transformatie voor A is

264/(Xmax-Xmin)*(A-Xmin) → D *(vergelijking van de lijn*)*

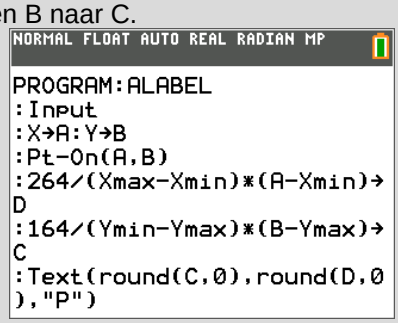
Op dezelfde manier krijgen we voor B de helling (rico)

(164-0)/Ymin-Ymax)

dus de transformatie voor B is

164/(Ymin-Ymax)*(B-Ymax) →C *(vergelijking van de lijn*)*

* bedenk dat Pixel-georiënteerde instructies het format (kolom#, rij#, tekst) gebruiken. Dit is de reden dat A naar D gaat en B naar C.



round() is nodig omdat de instructie alleen gehele getallen in het juiste interval kan gebruiken. Laat de leerlingen dit ontdekken!

Voor een TI-84 Plus gebruik je 95 in plaats van of 264 en 63 in plaats van164.