

Lektion 6: Zeichnen

Übung 1: Grundformen

In dieser Übung wirst du lernen, wie man die Möglichkeiten von **Zeichnen** in der Programmierumgebung des **TI-Nspire™ CX II-T** nützt.

Lernziele:

- **Zeichnen** (*Draw*) einiger Formen
- Umgehen mit dem Zeichenschirm
- Ein Programm mit **Zeichnen**-Befehlen starten und beenden
- Ändern der Zeichenfarbe

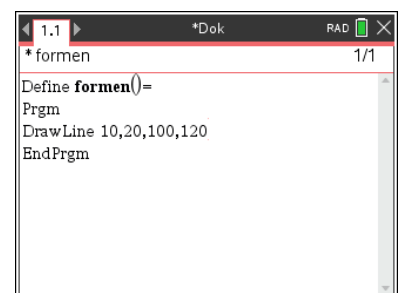
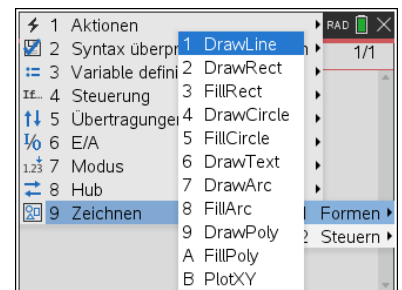
Hinweis: Lektion 6 erfordert den TI-Nspire CX II-T mit OS 5.0 oder höher. Programmiersprache und Betriebssystem haben **Zeichnen**- (*Draw*-) Werkzeuge eingebaut.

- Der Programmeditor enthält ein Zeichnen-Menü. (*Beachte: Diese Lektion ist nicht geeignet für TI-Nspire oder TI-Nspire CX Taschenrechner.*) Die **Zeichnen**-Werkzeuge und andere erweiterte Eigenschaften des OS 5.0 sind auch in der **TI-Nspire CX Premium Computer Software** verfügbar.
- Ein Programm, das **Zeichnen**-Befehle beinhaltet, kann nur über die *Calculator App* gestartet werden. Es läuft nicht in einer *Math Box* in einer *Notes App*.
- **Zeichnen** erfolgt in einem besonderen neuen Schirm, der zur Darstellung der Zeichnung über den *Calculator* gelegt wird. Das ist keine App aber sie überdeckt den ganzen Schirm. Man kann weder auf einer Graphs- noch auf einer Geometry-Seite zeichnen. Wenn das Programm „Fertig“ ist, führt ein beliebiger Tastendruck zurück zum *Calculator*.

Zeichnen einer Geraden

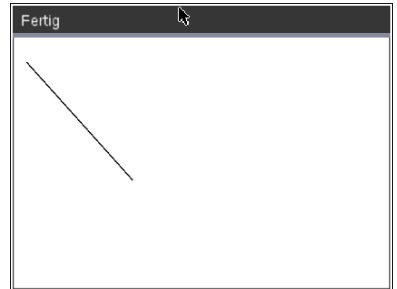
1. Beginn ein neues Dokument und öffne den Programmeditor. Gib dem Programm den Namen **formen**.
2. Drück **[menu]** und wähl **Zeichnen > Formen**. Du siehst die verfügbaren Formen.
3. Wähl **DrawLine**.
4. Du musst die Endpunkte der Geraden, (x_1, y_1) und (x_2, y_2) , angeben.
5. Das Eingabeformat des Befehls ist:
DrawLine x_1, y_1, x_2, y_2
6. Beginn mit den folgenden Werten:
DrawLine 10, 20, 100, 120

Beachte: Die **Zeichnen**-Befehle benötigen keine Klammern. Ihre Benützung führt zu einer Fehlermeldung.



- Drück `ctrl` `R` um das Programm aufzurufen und drück dann `enter` in der *Calculator App*, um es auszuführen.

Du siehst den Schirm wie rechts abgebildet. Das ist ein eigenes „Zeichenfenster“ zur Ausgabe von **Zeichnen**-Befehlen. (10,20) und (100,120) sind linker oberer und rechter unterer Endpunkt der Strecke. Die Titelleiste zeigt mit „Fertig“ an, dass das Programm beendet wurde. Der Rechner zeigt die Zeichnung weiterhin an. **Drück eine beliebige Taste, um das Zeichenfenster zu schließen und zum Calculator zurückzukehren.**



Hinweis: Die Standardmaßeinheit für die Dimensionen des Schirms ist *Pixel*.

Wenn du Werte außerhalb dieser Werte nimmst, dann gibt es keine Fehlermeldung, aber du wirst nichts sehen, da das Objekt außerhalb des sichtbaren Bereichs gezeichnet wird. Während das Zeichen-Fenster aktiv ist, sind E/A-Befehle wie **Request** und **Disp** nicht verfügbar. Die `getKey()`-Funktion kann eingesetzt werden. Das wird später in Übung 3 behandelt werden.

DrawLine erzeugt eigentlich nur eine Strecke. Da ergibt sich eine gute Gelegenheit, Algebra anzuwenden: Schreibe ein Programm, das zwei Punkte annimmt (4 Argumente) und dann eine Gerade *durch* die beiden Punkte quer über den ganzen Schirm zeichnet. Dazu muss man die Steigung und die Punkt-Richtungsform der Geraden kennen.

Zeichnen eines Rechtecks

- Geh zum Programmeditor zurück und füg unter dem **DrawLine**-Befehl eine Programmzeile für das Zeichnen eines Rechtecks hinzu; wähl dazu die Anweisung **DrawRect** über *menu > Zeichnen > Formen*

DrawRect 10,20,100,120

Nimm die gleichen Werte, die du für den **DrawLine**-Befehl verwendet hast.

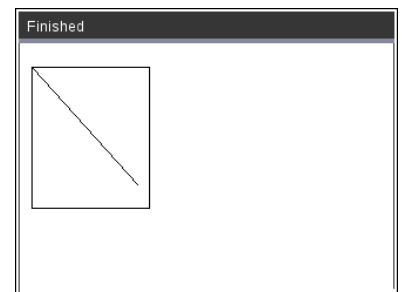
Kannst du dir vorstellen, was geschehen wird?

- Drück `ctrl` `R` und starte das Programm nochmals. Warum reicht die Gerade nicht bis in die untere rechte Ecke des Rechtecks?

Der **DrawRect**-Befehl hat das Eingabeformat:

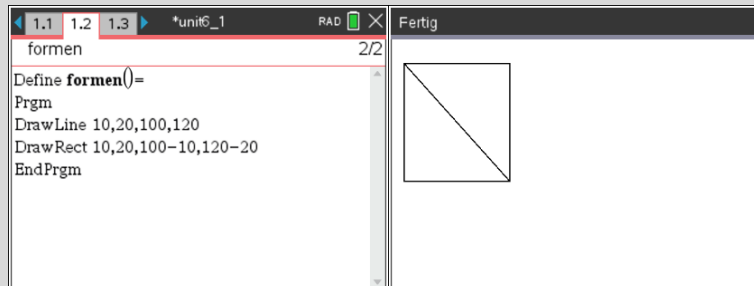
DrawRect x, y, Breite, Höhe

Welche *Breite* und *Höhe* müsste das Rechteck haben, dass die Gerade zur Diagonalen des Rechtecks wird?



Hinweis: Antwort: **DrawLine 10,20, 90, 100**

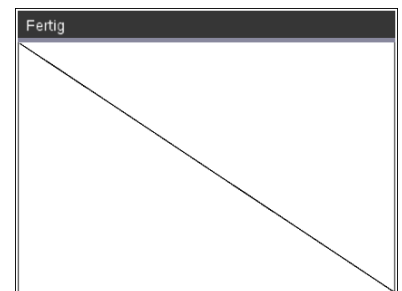
Man muss die Anfangskordinaten von Breite und Höhe abziehen.



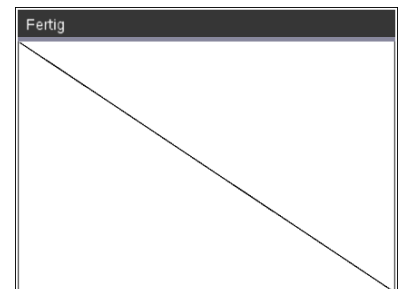
Sicherlich wirst du jetzt die Abmessungen des Schirms wissen wollen? Versuch vorher, mit einer Geraden eine Diagonale des Schirms zu zeichnen.

Die Abmessungen des Bildschirms

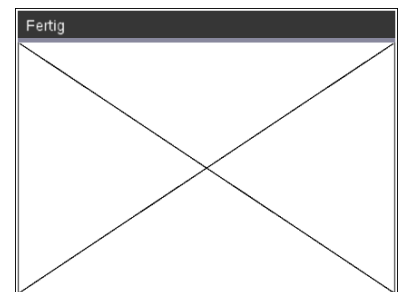
- Die Zeichenfläche hat die Abmessungen 318 x 212 (Pixel = Bildpunkte).
- Die linke obere Ecke des Zeichenfensters hat die Koordinaten (0,0).
- Die rechte untere Ecke hat die Koordinaten (317,211).



10. Der Befehl **DrawLine 0, 0, 317, 211** zeichnet eine Bildschirmdiagonale.



11. Schreib den Befehl zum Zeichnen der anderen Diagonale.



Hinweis: Nimm für die andere Diagonale den Befehl:

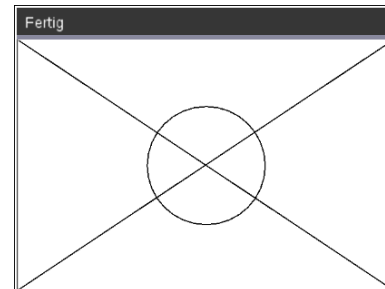
DrawLine 0, 211, 317, 0

Weitere Formen

DrawCircle x, y, r zeichnet einen Kreis mit dem Mittelpunkt in (x,y) und dem Radius r .

Beispiel:

- DrawCircle 159, 106, 50**



FillRect und **FillCircle** sind ähnlich; sie füllen die Formen mit der aktuellen Farbe.

Hinweis: Wie würdest du ein Quadrat zeichnen? Die Werte für Breite und Höhe sind gleich. Denk daran, dass wir jetzt nur von Pixel sprechen.

Mit **FillRect** kann die „Hintergrundfarbe“ weiß verändert werden.

Beispiel:

```
SetColor 255,255,0  gelb
FillRect 0,0,317,211  Standardfenster
```

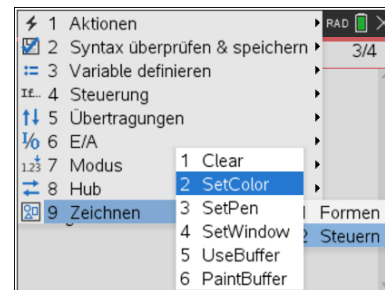
erzeugt einen gelben Schirm fürs Zeichnen.

Beim Einsatz von **SetWindow** werden die **FillRect**-Werte entsprechend angepasst.

Ändern der Zeichenfarbe

- Füg im Programmeditor zwischen zwei Zeilen den **SetColor**-Befehl ein:

menu > Zeichnen > Steuern > SetColor



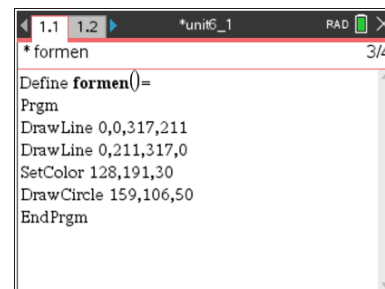
R, G, B

Die Zeichenfarbe wird durch drei Werte festgelegt: die Anteile von **Rot**, **Grün** und **Blau** werden gemischt. Jeder dieser Werte muss im Bereich zwischen 0 und 255 liegen.

SetColor Rotwert, Grünwert, Blauwert

Beispiel:

```
SetColor 128, 191, 30
```



Hinweis: Gewisse Farben wie Gelb, Cyan und Magenta haben spezielle RGB-Werte:

Gelb ist eine Mischung von Rot mit Grün

Cyan ist eine Mischung von Grün mit Blau

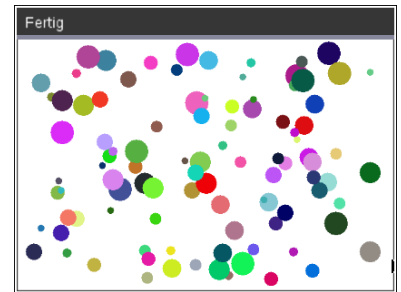
Magenta ist eine Mischung von Rot mit Blau

Je näher die Zahlenwerte bei 255 liegen, desto heller werden die Farben.

14. Setz den **SetColor**-Befehl *vor* das Zeichnen des Objekts.
15. Versuch, ein Programm zu schreiben, das 100 zufällige Kreise in verschiedenen Farben zeichnen lässt. Verwende dazu eine **For-EndFor**-Schleife.

Wenn du die Kreislinie und die Kreisfläche in unterschiedlichen Farben zeichnen willst, dann musst du zwei Befehle verwenden:

SetColor 255, 0, 0	<i>Farbe Rot</i>
FillCircle 100, 100, 50	<i>die Kreisfläche</i>
SetColor 0, 0, 0	<i>Farbe Schwarz</i>
DrawCircle 100, 100, 50	<i>die Kreislinie</i>



... Das zeichnet einen schwarzen Kreis, der rot gefüllt ist. Aber, wenn du *zuerst* den Kreis zeichnest (*draw*) und *nachher* füllst (*fill*), wirst du den schwarzen Kreis nicht sehen, da er von der roten Fläche überdeckt wird. Versuch das!

Hinweis: Lösungsvorschlag zum Zeichnen von 100 Zufallskreisen

```

Local i, r, g, b, x, y, ra
For i, 1, 100
  r := randInt(0, 255)
  g := randInt(0, 255)
  b := randInt(0, 255)
  © alle Kreise sollen vollständig am Schirm erscheinen...
  x := randInt(10, 308)
  y := randInt(10, 202)
  ra := randInt(3, 10)  © Variable r ist schon für den Farbwert für Rot in Verwendung
  SetColor r, g, b
  FillCircle x, y, ra
EndFor
    
```

Versuch eine andere Herausforderung: Zeichne nur mit den in dieser Übungseinheit eingeführten Anweisungen ein Bild.

