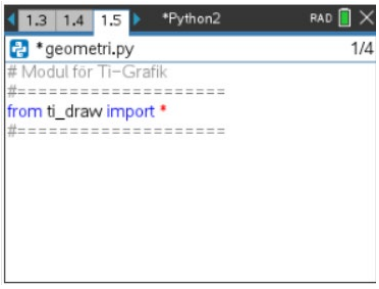
Kapitel 4: Grafik, att använda ti_draw modulen	Övning 1: Grafikskärmen
I det här kapitlet skall vi titta ti_draw-modulen som gör så att vi kan rita figurer med Python på TI-Nspire.	**Mål:** • Att använda ti_draw-modulen • Konfigurera ritytan och rita figurer

Du kan använda ti_draw-modulen för att rita grafiska figurer med hjälp av Python. I den här modulen hittar du kommandon för att rita punkter, linjer, cirklar, rektanglar och så vidare.

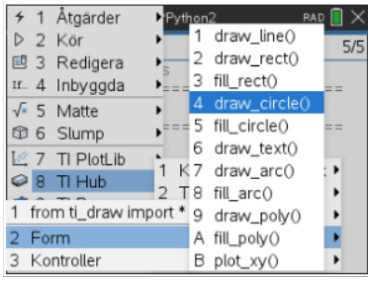
Starta ett nytt Python-program och välj typ: Geometri Grafik.
Du kan också ladda den här modulen efteråt med Meny> Fler moduler> TI Draw.



Som standard är skärmens (ritytans dimensioner) 318 pixlar på längden gånger 212 pixlar på höjden med koordinaterna för det övre vänstra hörnet (0,0).Så “x”-axeln går år höger och “y”-axeln går nedåt.

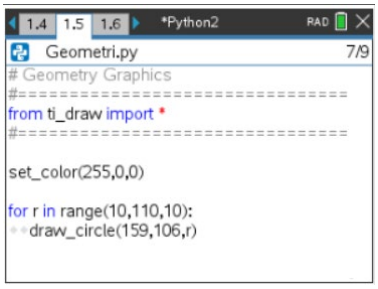
Skapa ett program som ritar en cirkel centrerad i mitten av skärmen och har en radie på 100.Till höger kan du se hur du hittar kommandot för att rita en cirkel i menyn. Funktionen för detta är: **draw_circle**(159,106,100).

För att komma ut ur grafläget så att du kan ändra kod trycker du på ESC.



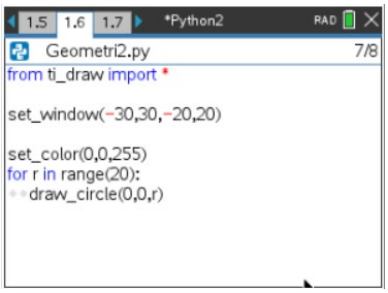
Ändra programmet så att 10 cirklar ritas med samma centrum, men vars radie varierar från 10 till 100.

I exemplet till höger har ett extra kommando lagts till. Detta kommando ställer in ritningsfärgen och du hittar den i TI Draw-menyn >Kontroller.
VI har också använt range(från, till, steg) där värdet på loopvariabeln ökas med steg-storleken.



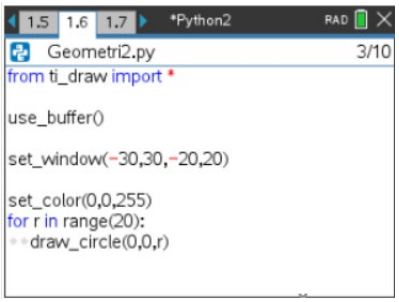
Tips: Du kan ändra färgen på en punkt eller en linje med funktionen set_color (). Denna funktion har tre argument. Dessa är RGB-värdena för färgen (röd, grön och blå). Dessa värden måste vara heltal (typ int) och ligga i intervallet [0-255]. Här ger (0,0,0) svart och (255, 255, 255) ger vitt.

Ett av kommandona i kontrollmenyn i ti-draw-modulen är **set_window()**.
Detta gör att du kan ställa in start- och slut-värden för koordinataxlarna som skall användas i ritfönstret. Vi har valt värden så att origo blir i mitten på ritytan. Det är också viktigt att bredden är en och en halv gånger så stor som höjden för att bibehålla proportionerna.



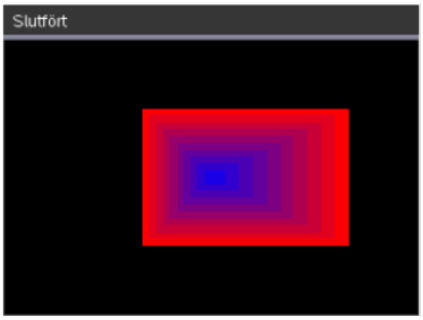
Rita nu 20 cirklar med mitten i mitten av skärmen och vars radie varierar från 0 till 20.

När du kör det här programmet tar det lite tid att rita.
Det beror på att ritmodulen måste kallas om och om igen.
Det finns ett kommando, **use_buffer()**, som gör så att grafikkommandon inte utförs direkt utan istället lagras, "buffras", tills allt ritande är klart och programmet avslutas, eller tills kommandot **paint_buffer()** körs. Du hittar dessa kommandon
Dessa kommandon har lagts till i figuren till höger.



Tips: Use_buffer() placeras vanligtvis i början av programmet innan några grafikkommandon har utförts.

Försök göra ett program som skapar figuren här bredvid – eller något liknande.



Tips: Använd fill_rect() för att göra hela grafikfönstret svart och för att skapa en serie rektanglar inne i varandra. Använd en for-loop med negativ steglängd för att skapa allt mindre rektanglar inne i varandra.

En möjlig lösning finns på nästa sida.

Ett förlag till lösning visas till höger.