

**Kapitel 4: Använda biblioteksmodulen ti-plotlib**
**Övning 1: Komma igång med grafik**

I den här första övningen i kapitel 4 får du lära dig att skriva och använda en instruktion för att göra grafiska representationer i Python. Du kommer också att lära dig att rita ett diagram och konfigurera skärmen på räknaren.

**Syfte:**

- Upptäcka ti-**plotlib**-modulen
- Representation av punkt och linje
- Göra en grafisk presentation

**1: Biblioteket eller modulen ti-plotlib**

Om du vill utföra en grafisk representation när du kör ett skript måste skriptet kunna förstå de grafiska instruktionerna. Det är därför nödvändigt att "lägga på" de grafiska funktionerna inom ett bibliotek **ti-plotlib**.

Starta ett nytt skript genom att namnge det K4OVN1 och inkludera ti-plotlib modulen. Du hittar den under f1 (Fns...) sedan Modul och alternativ 5. Välj sedan **1: import ti\_plotlib as plt**.

För den här första delen ska du skriva ett skript som visar en punkt vars koordinater är kända. Sedan ska du ändra skriptet så att det hittar din punkt i en markör och dessutom ändrar dess färg.

För att slutföra den här första övningen, ska du också namnge koordinatsystemets axlar och ge en titel till "grafan".

Definiera nu en funktion med koordinaterna för punkten som argument och lägg sedan till en instruktion så att punkten visas.

- Rensa först din skärm med hjälp av instruktionen **plt.cls ()** som du hittar i modulen **ti\_plotlib** i menyn.
- För att rita punkten, välj instruktion 6: Plot belägen i ritmenyn (**Draw**) för ti\_plotlib-modulen.
- Välj också önskad markering för punkten.

**Lärarkommentar:** Representationsformen 4:pixel ska du använda om du ska plotta ett stort antal punkter.

```

EDITOR: KAP4OVN1
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...

Esc Help
    
```

```

EDITOR: KAP4OVN1
ti_plotlib module
Setup Draw Properties
1:import ti_plotlib as plt
2:cls()          clear screen
3:grid(xsc1,ysc1,"style")
4>window(xmin,xmax,ymin,ymax)
5:auto_window(xlist,ylist)
6:axes("mode")
7:labels("xlabel","ylabel",x,y)
8:title("title")
9:show_plot()  display>[clear]

Esc Modul
    
```

```

EDITOR: KAP4OVN1
PROGRAM LINE 0005
import ti_plotlib as plt
plt.cls()
def punkt(x,y):
    plt.plot(x,y,"o")
...
...

Fns... a A # Tools Run Files
    
```

```

EDITOR: KAP4OVN1
ti_plotlib point marks
> plot(x,y,"mark")
1:o          circle
2:+          plus
3:x          cross
4:.          pixel

Esc
    
```

Fortsätt skriptet genom att begära visning av punkten genom att välja `plt.show_plot()` uttalande (instruktion 9 i ti-plotlib-modulen, ställa in menyn)..

- Be nu om körning av skriptet (tryck f4). Tryck sedan på `vars`- för att komma åt punktfunktionen `punkt()`.
- Ge koordinaterna för en punkt och observera sedan skärmen.
- Tryck på `clear` för att komma ur den grafiska skärmen, sedan på `vars` för att hitta listan över variabler i skriptet.
- Ändra koordinaterna för din punkt (t.ex. till `punkt(10,10)`) och upptäck att punkten inte längre syns på skärmen.

**Lärarkommentar:** När du skriver ett skript som använder grafiska funktioner är det nödvändigt att specificera parametrarna för grafikfönstret och om möjligt visa en markör, ett rutnät, namn på axlarna mm.

## 2: Förfina din representation

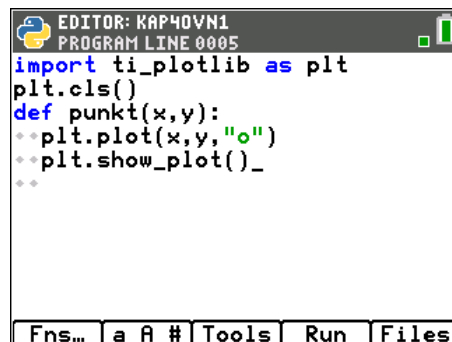
Inkludera `plt.cls()`-satsen under funktionsdefinitionen för att undvika att menyraden längst ner läggs ovanpå din grafiska representation.

**Lärarkommentar:** För att klippa ut, kopiera och klistra in en rad så använd verktyget Tools (f3) i editorn.

Från de olika alternativen i inställningsmenyn för modulen `tiplotlib` så ska du lägga till i skriptet de instruktioner som gör det möjligt att:

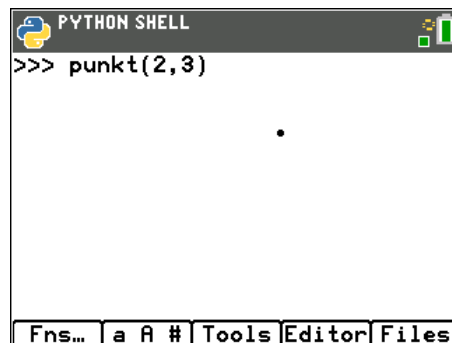
- Ställa in grafikfönstret så här:  
Xmin -10; Xmax - 10; Ymin - -10 och Ymax - 10  
Du väljer då alternativ 4:Window)
- Visa ett rutnät. Välj alternativ 3: grid. Typ av rutnät väljer du själv.
- Visa axlar (alternativ 6:axlar)
- Ställa in färgen på punkten. Välj Draw i menyn längst upp och sedan väljer du 1: color (r,g,b).

**Lärarkommentar:** Färgen hos en punkt eller ett spår specificeras med en kod, r, g, b (red, green, blue), där varje parameter kan ha värden i intervallet [0; 255]. Färgerna är kodade i 8 bitar eller 28 ==256 möjligheter, eftersom 0 också räknas med. 0 i koden betyder ingen färgkomponent. Det är också möjligt att få färg på rutnätet genom att komplettera satsen `grid(xscl, yscl, "style," (r,g,b))`.

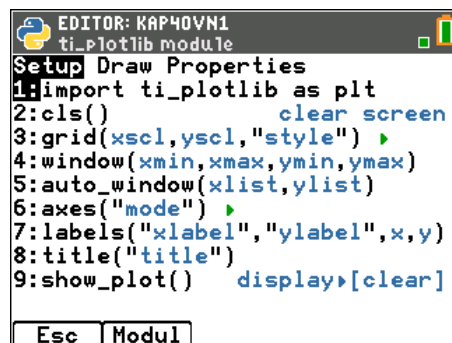


```
EDITOR: KAP40VM1
PROGRAM LINE 0005
import tiplotlib as plt
plt.cls()
def punkt(x,y):
    plt.plot(x,y,"o")
    plt.show_plot()_

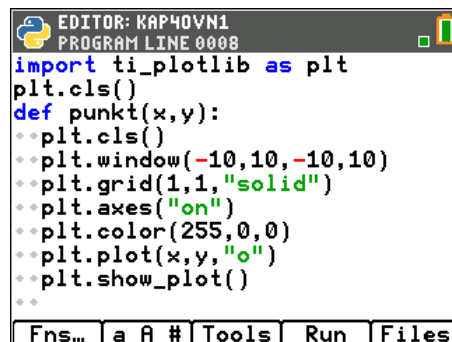
```



```
PYTHON SHELL
>>> punkt(2,3)
.
```



```
EDITOR: KAP40VM1
tiplotlib module
Setup Draw Properties
1:import tiplotlib as plt
2:cls() clear screen
3:grid(xscl,yscl,"style")
4>window(xmin,xmax,ymin,ymax)
5:auto_window(xlist,ylist)
6:axes("mode")
7:labels("xlabel","ylabel",x,y)
8:title("title")
9:show_plot() display>[clear]
Esc Modul
```



```
EDITOR: KAP40VM1
PROGRAM LINE 0008
import tiplotlib as plt
plt.cls()
def punkt(x,y):
    plt.cls()
    plt.window(-10,10,-10,10)
    plt.grid(1,1,"solid")
    plt.axes("on")
    plt.color(255,0,0)
    plt.plot(x,y,"o")
    plt.show_plot()

```

Kör skriptet och observera ändringarna. Du bör få en skärm identisk med den till höger.

Ändra skriptet igen för att ge axlarna ett namn. Till exempel ("x-axel" och "y-axel").

Det gör du genom att inkludera en rad, `plt.labels`, i skriptet (valfri plats). Den finns som alternativ 7: `Labels()` på menyn `Setup` i `tiplot`-modulen.

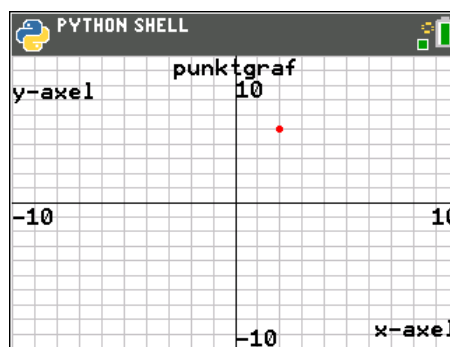
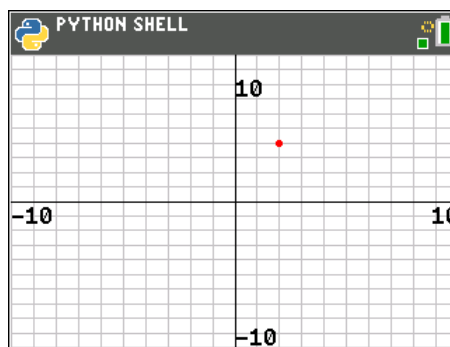
**Lärarkommentar:** Instruktionen `labels` ("`xlabel`", "`ylabel`", `x`, `y`) namnger axlarna genom att placera etiketterna för `x` och `y` på positionen 12 och 2 för `x`- resp `y`-axeln. Du kan själv ändra dessa värden.

Man kan också lägga till en titel med instruktionen `plt.title`.

Ska man ha med tecken å, ä eller ö får man gå till teckenpaletten genom att trycka på `f2` (a `A #`).

Du kan också komplettera ditt skript ytterligare genom att t.ex. skriva en instruktion som ritar ett linjesegment. Se skriptet till höger.

**Lärarkommentar:** Om du lägger till tecknet `#` framför en instruktion så kommer den inte att utföras. Ett snabbval för att nå detta tecken är `2nd``3`. Det finns även andra sätt att nå detta tecken, t.ex. genom teckenpalettsmenyn `f3` (a `A #`).



```

EDITOR: KAP4OVN1
PROGRAM LINE 0019
♦♦ plt.color(255,0,0)
♦♦ plt.plot(x,y,"o")
♦♦ plt.show_plot()
#rita ett linjesegment
def linje(x0,y0,x1,y1):
♦♦ plt.cls()
♦♦ plt.pen("medium","solid")
♦♦ plt.line(x0,y0,x1,y1," ")
♦♦ plt.show_plot()
-
Fns... a A # Tools Run Files
    
```