

Unit 5 : Att använda ti\_image-modulen

Övning 3: Arbeta med bilder

I den här lektionen introduceras vi till ti\_image-modulen som kan användas för att redigera bilder.

**Mål:**

- Importera bilder
- Bildbehandling

Modulen ti\_image gör det möjligt att importera och redigera bilder med hjälp av Python.

Vi skall använda bilden på ballonger som du kan se till höger. Bilden finns med i Nspire-programvaran, men du kan också kopiera bilden från detta dokument.

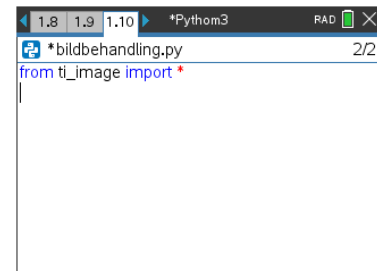


Vi ska skapa ett program som omvandlar den här bilden till en bild i gråskala.

Denna uppgift måste helt eller delvis utföras i datorversionen av TI-Nspire.

Öppna ett nytt Python-program och importera ti\_image-modulen.

(Detta kan till också göras genom att välja programtypen "Bildbehandling" – men då får du med lite mer som vi inte behöver just nu.)

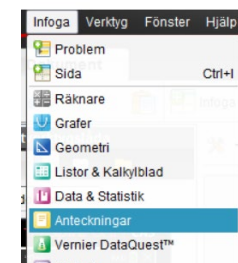


```
1.8 1.9 1.10 *Python3 RAD X  
*bildbehandling.py 2/2  
from ti_image import *
```

För att redigera en bild i Python måste den bilden vara på en anteckningssida.

För att lägga till en bild på en anteckningssida så måste vi använda datorprogramvaran. Det fungerar inte på handenheten.

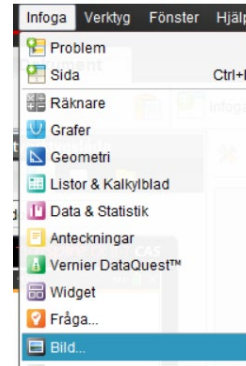
I programvaran lägger du till en anteckningssida i det aktuella dokumentet.



Sedan kan du importera en bild till den här sidan med menyalternativet "infoga - bild".

(Du kan använda en valfri bild eller kopiera bilden från det här dokumentet.)

Du kan nu ladda över dokumentet till handenheten eller fortsätta på datorn.



Nu när bilden finns på anteckningssidan måste du ge den ett namn.

Flytta markören över bilden och använd ctrl-meny (eller ctrl-klicka eller högerklicka) för att få en popup-meny. Välj sedan 5: Namn på bild och ge bilden ett namn (till exempel Ballong)

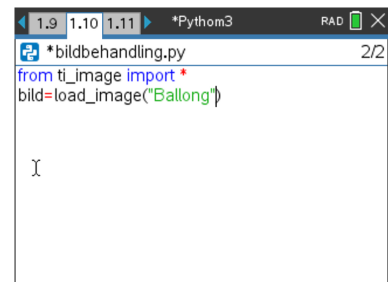


Gå tillbaka till Python-sidan för att fortsätta skriva programmet.

Vi kan lagra bilden i en variabel med kommandot **load\_image()**.

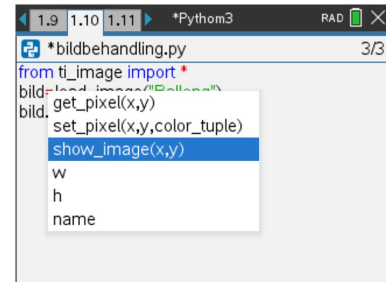
Du hittar detta i Meny>Fler moduler >TI-image> load\_image().

I det här exemplet kallar vi variabeln där vi lagrar bilden "bild".



Om du nu skriver bildens variabelnamn följt av en punkt så dyker det upp en meny med ett antal val.

En av dessa val är **show\_image()**.



```
*bildbehandling.py 3/3
from ti_image import *
bild=load_image("Ballong")
bild.get_pixel(x,y)
bild.set_pixel(x,y,color_tuple)
show_image(x,y)
w
h
name
```

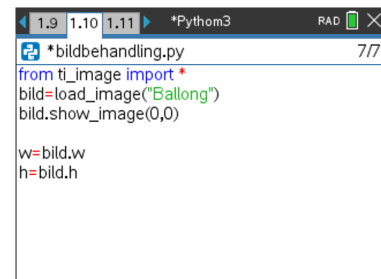
Vi väljer det alternativet och anger koordinaterna (0,0).

Detta betyder att bildens övre vänstra hörn hamnar 0 pixlar från vänster och 0 pixlar nedåt.

Om du nu kör programmet så visas bilden.

För att redigera bilden behöver vi bildens dimensioner i pixlar.

Du kan finna dessa med bild.w för bredden och bild.h för höjden.



```
*bildbehandling.py 7/7
from ti_image import *
bild=load_image("Ballong")
bild.show_image(0,0)

w=bild.w
h=bild.h
```

Vi gör nu två loopar i varandra som tillsammans går igenom alla pixlar och beräknar motsvarande gråskalevärde för varje pixel, varefter vi skriver in de nya värdena in i bilden.

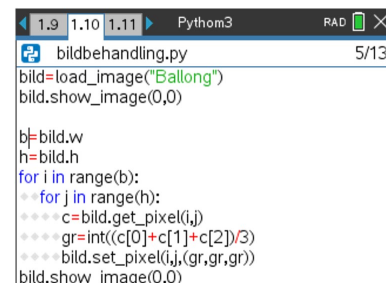
Med kommandot **get\_pixel (i, j)** får du färgerna från motsvarande pixel givet som en slags lista (en så kallad **tuple**) som består av tre element, (röd, grön, blå).

För varje pixel beräknar vi medelvärdet av dessa tre tal och så ger vi alla tre färgerna detta värde. (Tre lika färgvärden ger en grå nyans)

Slutligen skriver vi ut bilden på skärmen igen.

Programmet tar en stund att köra. Så länge det körs så står det Körs... över ritfönstret.

(Ur grafisk synvinkel vore det bättre att istället beräkna snittet med  $gr=int(\sqrt{(c[0]**2+c[1]**2+c[2]**2)/3})$ , det så kallade kvadratiske medelvärdet. Detta gör så att ljusstyrkan på bilden bättre överensstämmer med originalet. Tänk på att du behöver importera matematikmodulen om du vill prova detta.)



```
*bildbehandling.py 5/13
bild=load_image("Ballong")
bild.show_image(0,0)

b=bild.w
h=bild.h
for i in range(b):
    for j in range(h):
        c=bild.get_pixel(i,j)
        gr=int((c[0]+c[1]+c[2])/3)
        bild.set_pixel(i,j,(gr,gr,gr))
bild.show_image(0,0)
```