



#### Unit 5: Het gebruik van de ti\_image module

#### Oefenblad 3: Werken met afbeeldingen

In deze les maken we kennis met de ti\_image module waarmee afbeeldingen kunnen worden bewerkt.

#### Doelen :

- Importeren van afbeeldingen.
- Bewerken van afbeeldingen.

De ti\_image module maakt het mogelijk afbeeldingen te bewerken.

**(Je hebt de software nodig om een afbeelding in je document te importeren)**

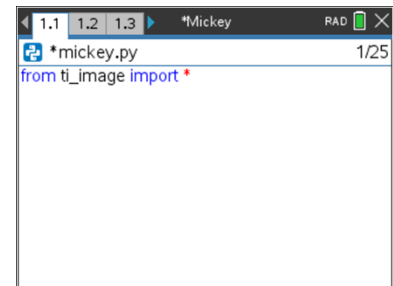
Hiernaast staat een afbeelding van Mickey Mouse.

We maken een programma waarmee we deze afbeelding veranderen in een afbeelding met alleen grijstinten.



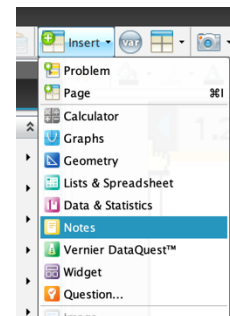
Open een nieuw Python programma en importeer de ti\_image module.

(Dat kan bijvoorbeeld door het type "Image Processing" te kiezen)

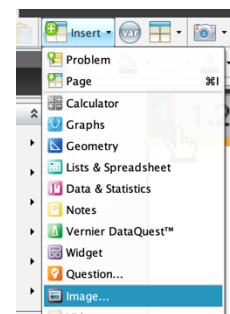


Om in Python een afbeelding te bewerken moet deze afbeelding in een Notes-pagina staan.

Voeg in het huidige document een Notes-pagina toe.

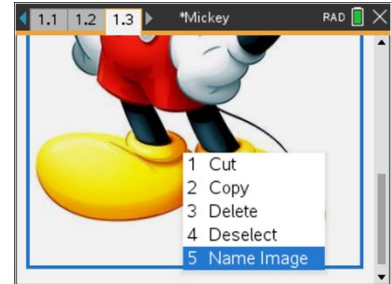


Daarna kun je in deze pagina een afbeelding importeren met de menu optie "insert - image".

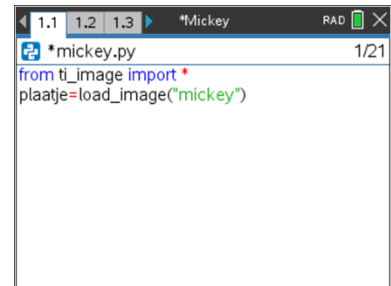




Als de afbeelding in de Notes pagina staat, moet je deze een naam geven.  
Ga met de cursor op de afbeelding staan en gebruik Ctrl\_menu (of Ctrl-muisklik) om een pop-up menu te krijgen. Kies dan 5:Name Image en geef de afbeelding een naam (bijvoorbeeld mickey)



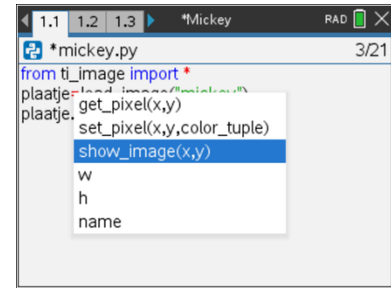
Ga terug naar de Python pagina om het programma te schrijven.  
Hiernaast zie hoe je de afbeelding in Python kunt krijgen.  
(De opdracht load\_image() kun je vinden in Menu --> More Modules --> TI Image --> load\_image()).  
We noemen de variabele waarin we de afbeelding opslaan in dit voorbeeld "plaatje".



Als je nu de variabele-naam van de afbeelding intypt gevolgd door een punt krijg je een menu in beeld waarin een aantal keuzes staan.

Een ervan is `show_image()`

We kiezen deze en geven aan waar de afbeelding moet worden geplaatst.



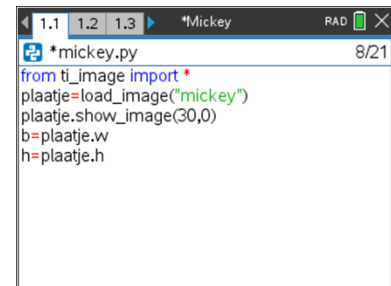
```
1.1 1.2 1.3 *Mickey RAD 3/21
*mickey.py
from ti_image import *
plaatje=load_image("mickey")
plaatje.get_pixel(x,y)
plaatje.set_pixel(x,y,color_tuple)
show_image(x,y)
w
h
name
```

(30,0) betekent: 30 pixels vanaf links en 0 pixels omlaag en is de positie van de linkerbovenhoek van de afbeelding.

Als je nu het programma uitvoert wordt de afbeelding afgedrukt.

Om de afbeelding te bewerken hebben we de afmetingen (in pixels) nodig.

Deze kun je opvragen met `plaatje.w` (breedte) en `plaatje.h` (hoogte).



```
1.1 1.2 1.3 *Mickey RAD 8/21
*mickey.py
from ti_image import *
plaatje=load_image("mickey")
plaatje.show_image(30,0)
b=plaatje.w
h=plaatje.h
```

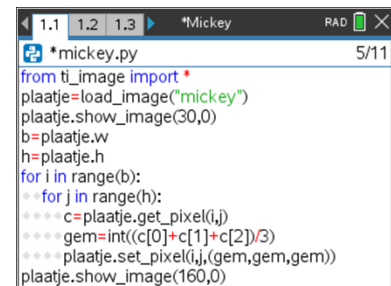
We maken nu twee lussen die samen alle pixels doorlopen en rekenen voor elke pixel de grijswaardes uit waarna we die nieuwe waardes weer in de afbeelding "terugzetten".

Met de opdracht `get_pixel(i,j)` krijg je van pixel `i,j` de kleuren `(r,g,b)`.

Dit is een tuple (een soort lijst) die bestaat uit drie elementen.

Voor elke pixel rekenen we het gemiddelde van deze drie getallen uit en geven elke kleur deze waarde. (Drie gelijke kleurwaarden geven een grijs tint)

Als laatste drukken we de afbeelding weer af op het scherm.



```
1.1 1.2 1.3 *Mickey RAD 5/11
*mickey.py
from ti_image import *
plaatje=load_image("mickey")
plaatje.show_image(30,0)
b=plaatje.w
h=plaatje.h
for i in range(b):
    for j in range(h):
        c=plaatje.get_pixel(i,j)
        gem=int((c[0]+c[1]+c[2])/3)
        plaatje.set_pixel(i,j,(gem,gem,gem))
plaatje.show_image(160,0)
```