



Unit 6: Het gebruik van de ti_hub module

Oefenblad 1: Built-in devices en de hub

In deze les gebruiken we de TI-Innovator Hub in een Python programma.

Doelen :

- Het verkennen van de ti_hub module.
- Het toepassen van enkele basisvaardigheden.

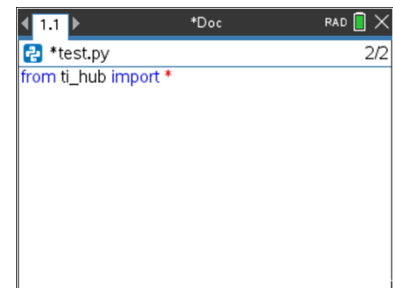
In de ti_hub module vind je de opdrachten om vanuit Python met de TI-Innovator Hub te communiceren.

We gaan een programma maken dat de gekleurde led in de hub laat branden en dat de ingebouwde luidspreker gebruikt.

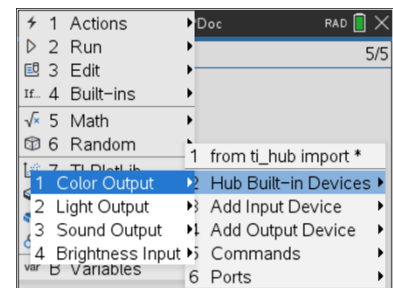


Sluit de hub aan op de rekenmachine of op de computer.

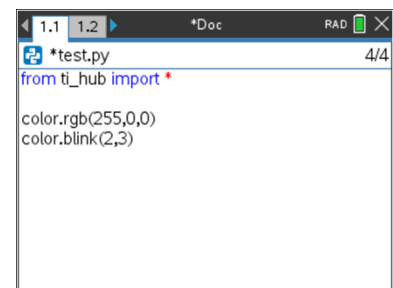
Open een nieuw Python programma en kies voor "Blank Program".
 Om de Innovator Hub te kunnen gebruiken heb je de Hub module nodig.
 Kies uit het menu TI Hub en dan optie 1: **from ti_hub import ***.
 Je hebt dan de beschikking over alle hub-functies.



In het TI Hub menu staat bij Hub Built-in Devices de Color Output.
 Kies hierin voor rgb() om een kleur te kiezen, en daarna in hetzelfde menu voor blink om het lampje te laten knipperen.



In het voorbeeld hiernaast is gekozen voor de kleur rood en knippert het lampje 3 seconden lang met een frequentie van twee keer per seconde.



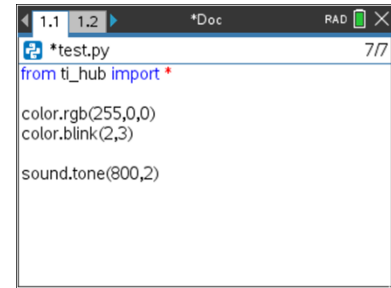


We gaan ook geluid toevoegen.

Ga weer naar het ti_hub menu en daarin weer de Built-in devices en kies nu voor Sound Output.

Kies hieruit **tone(freq, time)**.

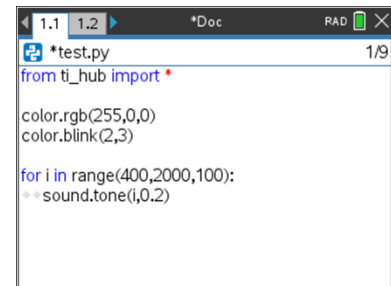
Laat nu 2 seconden een toon van 800 Hz horen.



```
1.1 1.2 *Doc RAD X  
*test.py 7/7  
from ti_hub import *  
  
color.rgb(255,0,0)  
color.blink(2,3)  
  
sound.tone(800,2)
```

Pas het programma aan door de toonhoogte te variëren.

Maak een loop die de frequentie laat lopen van 400 tot 2000 Hz met stappen van 100 Hz waarbij elke toon 0.2 seconde duurt.

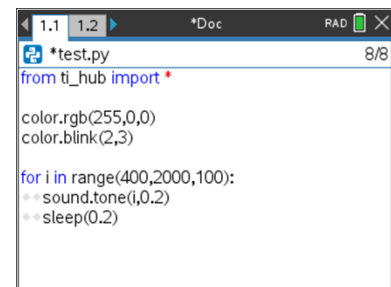


```
1.1 1.2 *Doc RAD X  
*test.py 1/9  
from ti_hub import *  
  
color.rgb(255,0,0)  
color.blink(2,3)  
  
for i in range(400,2000,100):  
    sound.tone(i,0.2)
```

Als je dit programma uitvoert merk je dat niet elke toon 0.2 seconde duurt.

Belangrijk is dat de tijd tussen twee opeenvolgende tonen ook minstens 0.2 seconde is.

Je kunt hiervoor de opdracht **sleep()** gebruiken.



```
1.1 1.2 *Doc RAD X  
*test.py 8/8  
from ti_hub import *  
  
color.rgb(255,0,0)  
color.blink(2,3)  
  
for i in range(400,2000,100):  
    sound.tone(i,0.2)  
    sleep(0.2)
```