

**Unit 1: PythonによるTI-Innovator™ Hubのスタート**

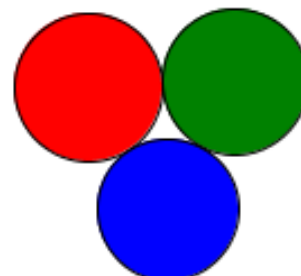
**Skill Builder 2: 色とHubプロジェクト**

このレッスンでは、Pythonプログラムテンプレートを使ってTI-Innovator HubのカラーLEDを制御する方法について学習します。

**目標**

- Pythonプログラムテンプレートの使用
- TI-Innovator HubのカラーLEDの制御

カラーLEDには、赤、緑、青の3つの色のチャンネルがあります。これはしばしばRGB LEDと呼ばれます。コンピュータ画面、スマートフォン画面、テレビ画面はすべてこれらのLEDを使って画像を作成します。



ユニークな色を得るには、適切な量の赤、緑、青を混ぜます。これら3つの原色の光を適切に組み合わせることで、多くの色が可能になります。

**Teacher Tip:** このレッスンでは、PythonのHub Projectプログラミングテンプレートを紹介합니다。このTexas Instrumentsが提供する機能は、TI-Innovator Hubを使うプロジェクトで使用される多くの異なるモジュールと特別な機能をインポートします。このレッスンでは、これらすべてのツールについて説明している訳ではありませんが、このテンプレートを今後のTI-Innovator Hubレッスンで取り入れ、コース全体でほとんどの機能について説明します。Pythonを初めて使う場合、役に立ちます。

テンプレートには、次の3種類のインポートステートメントが表示されます。

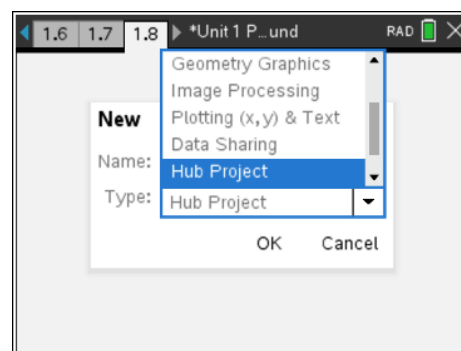
```
from ti_system import *  ti_hubモジュールからすべてをインポートします。
import ti_plotlib as plt  ti_plotlibモジュールからすべてをインポートします。
                           これらの関数の前にはpltという名前を付ける必要があります。
from time import sleep   timeモジュールからのみsleep関数をインポートします。
```

1. **home > New**(ホーム>新規)を使って新規のTI-Nspire™ ドキュメントを開始するか、**ctrl+doc** (+ page)を使って現在のドキュメントに新規のページを挿入します。

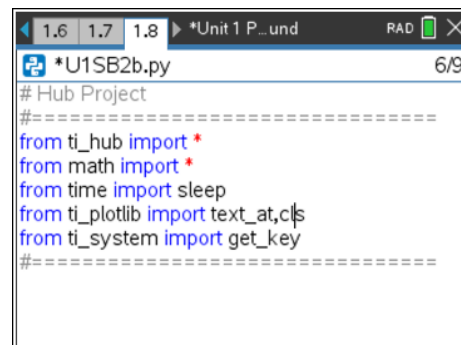
**Add Python > New....** (Python追加>新規...)を選択します。

Pythonプログラムの名前を入力します(U1SB2としました)。

まだenterを押さないでください! **Type:** (タイプ:)ドロップダウンリスト(フィールドの右側にある矢印をクリック)から、**Hub Project**を選択します。**enter**を押します。



2. Hub Projectテンプレートは、Pythonエディタ上にいくつかの**import**(インポート)ステートメントを提供します。#commentステートメントもあります。#記号(number(ナンバー), hash(ハッシュ), pound(ポンド), octothorpe(オクトソープ))はコメントの開始を示し、コメントはプログラムの実行時に無視されます。コメントは、プログラムに個人的なメモを追加するため使われます。





3. **import**ステートメントは、TI-Innovator Hubを操作するとき役立つツールを提供します。このレッスンではこれらすべてのツールについては説明しませんが、後のレッスンで使います。最初のレッスンでは、さまざまなメニューからこれらのインポートステートメントを選択して、`ti_hub`モジュールと`time`モジュール(`sleep()`関数用)を使いました。テンプレートには、いくつかの便利なツールが集められています。TI-Nspire CX IIIには、さまざまなタイプのPythonプログラミングプロジェクトで利用できるテンプレートがあります。

4. カーソルをプログラムの一番下(一番下の`#=====`コメント行の下)に置きます。

**menu > TI Hub > Hub Built-in Devices > Color Output** (メニュー > TI Hub > Hub内蔵デバイス > カラー出力)を押します。

**rgb(red,green,blue)**を選択すると、プログラムに次のステートメントが置かれます。

### color.rgb(red, green, blue)

3つのインラインプロンプト(行内入力要請)red, green, blueは、ホバリング(空中停止)しているツールチップが示すように、それぞれ0~255の数字に置き換える必要があります。**tab**または**右矢印**を押してフィールド間を移動することにより、3つの数字を選択します。

**Teacher Tip:** メニューからステートメントを選択すると、インラインプロンプトとホバリングツールチップが表示されます。ステートメントが手動で入力された場合は表示されず、入力された情報が削除された場合は返されません。

何色が可能ですか。  $256 \times 3 = 256^3 = 16,777,216$

**color.rgb()**について : `color`はクラスのインスタンスであり、`.rgb()`はそのクラスのメソッドです。`.rgb()`関数には3つの引数が必要です。

5. プログラムを実行します。シェルアプリが画面に表示され、TI-Innovator HubのカラーLEDが点灯します。つぎに、ドキュメントのページをPythonエディタに戻し(**ctrl+左矢印**)、3つのカラーチャネルに対して他の数値を試してください。

```

Python Shell
>>>#Running U1SB2.py
>>>from U1SB2 import *
>>>

```

6. **Color Output**メニューの他の2つの関数は、次のとおりです(右図参照)。

### blink(frequency, time) off().

1 Color Output	1 rgb(red, green, blue)
2 Light Output	2 blink(frequency, time)
3 Sound Output	3 off()

これらが何するものか予測できますか。プログラムに追加して、テストしてみましょう。

作業内容を保存することを忘れないでください。

**Teacher Tip:** シェルでプログラムを実行した後、シェルは`color.rgb()`関数を認識します。コマンドプロンプト(>>>)で、`color.`(ピリオドを含む)という単語を入力すると、選択リストに3つの方法が表示されます。コマンドを完了して`enter`を押すと、シェルから直接カラーLEDを制御できます。

```
Python Shell 5/5
>>>#Running U1SB2.py
>>>from U1SB2 import *
>>>#Running U1SB2.py
>>>from U1SB2 import *
>>>c
  rgb(red,green,blue)
  blink(freq,time)
  off()
```