

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

Skill Builder 1: ライトアップ

このレッスンでは、Pythonプログラムの作成・実行と、TI-Innovator Hubのlight(赤色LED)を使った基本事項を学習します。

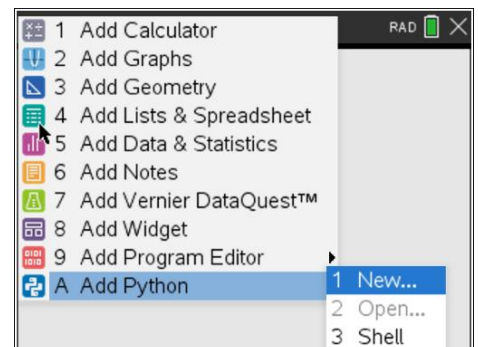
目標

- Pythonプログラムの作成・実行
- TI-Innovator Hubのlightを制御

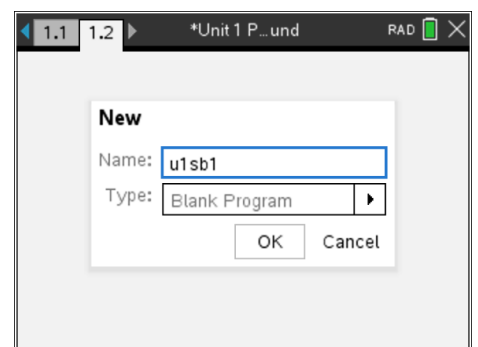
TI-Nspire CX II でPythonを使ったTI-Innovator Hubプログラミングの世界へようこそ!  
最初のプログラムは、TI-Innovator Hub回路基板の赤色LEDを操作します。ボード上では見づらいですが、電源を入れると分かります。



1. 新規TI-Nspire™ドキュメントを開きます。利用可能なアプリケーションが一覧表示されます。**Add Python**を選択してから、**New...**を選択します。



2. Pythonプログラム名として**u1sb1**と入力し、**enter**を押します。  
2番目の欄のTypeについては、後で説明します。



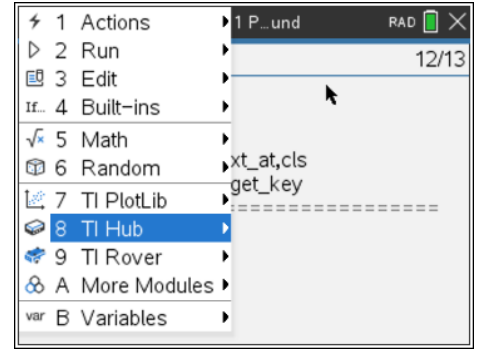


3. これでPythonエディタが表示されます。menuを押します。各メニュー項目には、関連するPythonプログラミングツールが表示されています。今のところ、私たちの主な関心はTI Hubメニューです。そのメニューから一番上の項目を選択します。

```
from ti_hub import *
```

このPythonコマンドは、TI-Innovator Hubでデバイス进行操作する(または接続する)ため必要なツール(コマンド)を提供します。

また、このステートメントはTI-Innovator Hubが接続されているかどうかを確認します。そうでない場合、プログラムは実行されません。



4. 使用するステートメントは次のものです。

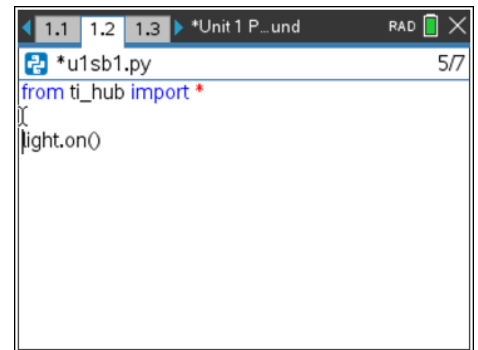
```
light.on()
```

これが何をするのか予測できますか。

このステートメントは、次のところにあります。

```
menu > TI Hub > Hub Built-in Devices > Light Output > on()
(メニュー>TI Hub>Hub内蔵デバイス>光出力>オン())
```

Hub関係の関数は、すべてTI Hubメニューにあります。

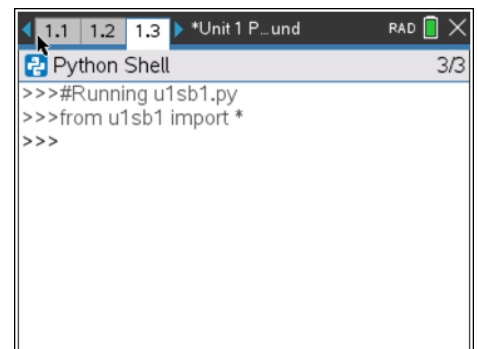


5. これで、この素晴らしいプログラムを実行する準備が整いました。使うことができます。

```
menu > Run > Run (Ctrl+R)
```

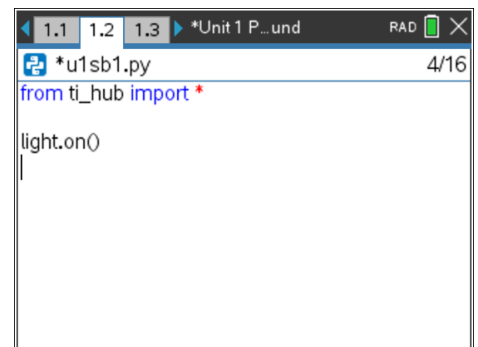
```
(メニュー>実行>実行(Ctrl+R))
```

実行は、ショーカットキーctrl+Rと押しもかまいません。TI-Nspire CX IIの画面は右図のようになります。ctrl+Rを押すとドキュメントにページが追加され、Python Shell(シェル)アプリが開きます。右図はPythonシェル画面です。シェルはTI-Nspireの電卓アプリケーションに似ています。これは、Pythonプログラムが実行される場所です(実際には、唯一の場所です)。



記号>>>はPythonコマンドプロンプトです。次のコマンドを待っています。しかし、TI-Innovator Hubを見ると、赤色LEDが点灯しているのが分かります。これは、プログラムのlight.on()ステートメントの結果です。

6. Pythonプログラムは、シェルページの前のページにあります。プログラム編集に戻るには、ctrl+左矢印と押します(右図)。TI-Innovator Hubで、赤色LEDがまだ点灯していることに注意します。

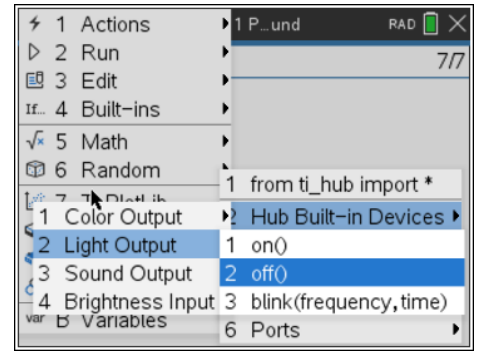




7. どのコマンドがライトをオフにするか推測できますか。それは次のところから見つけることができます。

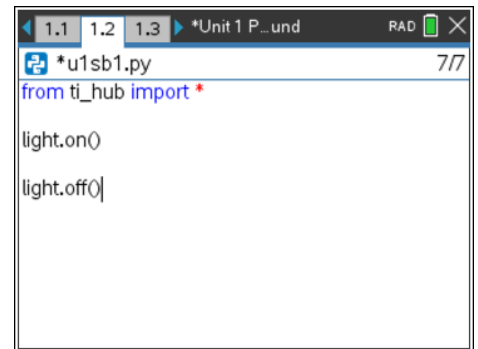
**menu > TI Hub > Hub Built-in Devices > Light Output > ...**

(メニュー>TI Hub>Hub内蔵デバイス>光出力>...)



8. **light.on()**コマンドの後に**light.off()**コマンドを追加します。行を空けてもかまいません。これらは実行に影響を与えず、コードを読みやすくします。プログラムを再度実行してください。LEDが速く点滅するのがわかりますか。速すぎる？

次のいくつかのステップでは、LEDが点灯している時間を制御する機能を追加します。



9. **from ti\_hub import \***の下の行にカーソルを置きます。

**menu > More Modules > Time** (メニュー>その他のモジュール>時間) と押して、次を選択します。

**from time import \***

**light.on()**と**light.off()**の間に、次のステートメントを追加します。

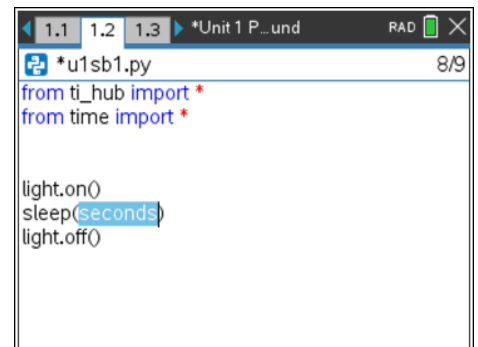
**sleep(seconds)**

これは、**Time**メニューと同じところにあります。

seconds(秒)はプレースホルダー(仮に置いた文字列)です。

2や3などの数字に置き換えます。

sleep()関数は、プログラムの次のステートメントに進む前に、その秒数の間、待機または一時停止するようコンピュータに指示します。



ここでプログラムを実行すると(**ctrl+R**を押す), LEDはオフになる前に選択した秒数の間、点灯したままになります。



## 10 Minutes of Code - Python

TI-NSPIRE™ CX II WITH THE TI-INNOVATOR™ HUB

UNIT 1: SKILL BUILDER 1

STUDENT ACTIVITY

10. ライトを点滅させるには、プログラム内の一連のステートメントを繰り返すか、または…。

Light Output(光出力)メニューには、LEDを点滅させる**blink()**関数もあります。**light.blink()**には、frequency(周波数, 頻度)とtime(時間)の2つのパラメータがあります。両方を数値に置き換えて、パターンを理解してください。ポップアップツールチップに注意します!

```
1.1 1.2 1.3 *Unit 1 P...und RAD [X]  
*u1sb1.py 9/11  
from ti_hub import *  
from time import *  
  
light.on()  
sleep(seconds)  
light.off()  
  
light.blink(frequency,time)
```