



#### Unit 3: 明るさ, TI-Innovator™ Hub でのifとwhile

#### Skill Builder 1: 光の測定

このレッスンでは、TI-Innovator Hubの明るさセンサ (brightness sensor)を制御し、その情報を利用する方法を学習します。

#### 目標

- 明るさセンサを読む
- 明るさセンサの範囲設定
- 明るさセンサの監視
- 明るさセンサでライトを制御

TI-Innovator Hubのlight(光), color(色), sound(音)機能とは異なり, brightness sensor(明るさセンサ)は出力デバイスではなく入力デバイスです。プログラムは、明るさセンサから情報を取得し、その数値に基づいて動きをとることができます。既定値の明るさの値を使うか、指定した **brightness.range()**関数を使って値の範囲を設定するかのどちらかができます。

明るさセンサは、TI-Innovator Hubの一方の端にラベルではっきりと表示されています。



1. 新規のPython Hub Projectを開始します。次の3つのステートメントから始めます。

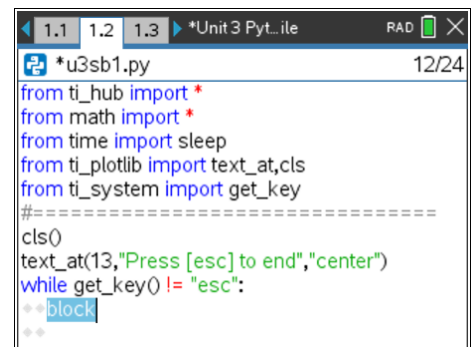
```
cls()
text_at(13,"Press [esc] to end","center")
while get_key() != "esc":
    block
```

これらはすべて **menu > TI Hub > Commands**(メニューTI Hub>コマンド)にあります。

**cls()**は画面をクリアします。

**text\_at(13, ...)**は画面中央下にメッセージを表示します。

プログラムを実行したときは、**esc**を押して終了します。



2. **while**ブロックで、2つのステートメントを使います。1つは明るさを読み取るもの、もう1つは値を表示するものです。

明るさの測定値を読み取るため、パラメータ**b**に代入します。

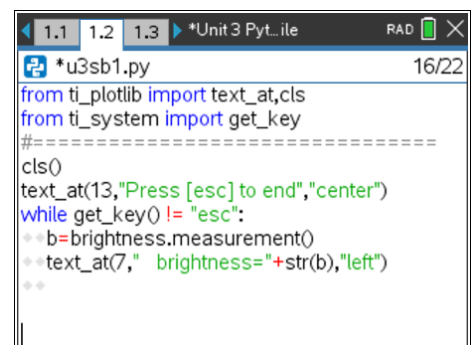
```
b = brightness.measurement()
```

これは、**menu > TI Hub > Hub Built-in Devices > Brightness Input** (メニュー>TI Hub>Hub内蔵デバイス>明るさ入力)にあります。

加えて

```
text_at(7, "brightness = " + str(b), "left")
```

**text\_at(7...**は画面の垂直方向の中央です。





#### 3. "brightness = " + str(b)について

**str(b)**はmenu > Built-ins > Type(メニュー>組み込み>タイプ)にあります。これは、**b**の数値を文字列に変換します。**text\_at()**が表示できるのはテキスト(文字)のみで、数値変数の値は表示できないためです。

**+** 記号は、"brightness ="という単語と文字**b**の値を組み合わせたものです。このタイプの文字列の追加は、コンカチネーション(concatenation, 連結)と呼ばれます。

この場合、位置合わせの**left**は**center**よりも優れています。**center**を使う場合、線の長さが異なるため新しいデータで古いデータを完全に消去できない場合があります。" brightness = "の前(引用符の中)にスペースを追加すれば、テキストを中央に近づけることができます。

#### 4. プログラムを実行して、画面を確認します。右図のように表示され、明るさの値が変化します。

#### 5. 表示を少し遅くします。**text\_at()**ステートメントの直後に**sleep()**ステートメントを追加します。**while**ブロック内の他のステートメントと一致するようインデント(字下げ)されていることを確認します。

プログラムを再度実行し、センサに当たる光の強度を変更します。センサが現在表示している値から最小値と最大値を決定します。

#### 6. 明るさセンサが提供する値の範囲は、次のステートメントを使って設定できます。

### brightness.range(min, max)

これは、 menu > TI Hub > Hub Built-in Devices > Brightness Input (メニュー>TI Hub>Hub内蔵デバイス>明るさ入力)にあります。

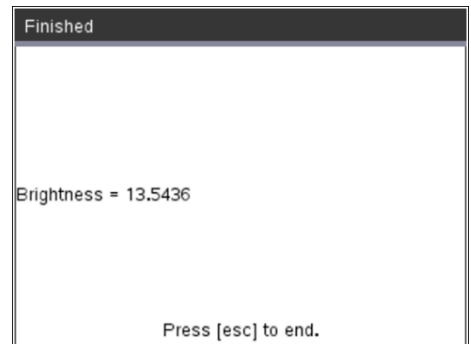
このステートメントは、必ずwhileループの前に置いてください。**min**と**max**には任意の値を指定できますが、**min < max**であることを確認します。

範囲を変更し、プログラムを再度実行して、生成された値を確認します。これでカスタムのデジタル光度計ができました。

これはなぜ大切ですか? 次のレッスンで確認しましょう。

```
# b=brightness.measurement()
text_at(7,"Brightness =" + str(b),"left")
```

Brightness = 13.5436



```
1.1 1.2 1.3 *Unit 3 Pyt...ile RAD 8/17
*u3sb1.py
#=====
cls()
text_at(13,"Press [esc] to end","center")
while get_key() != "esc":
  *b=brightness.measurement()
  *cls()
  *text_at(7,"brightness =" + str(b),"left")
  *sleep(.25)
```

```
1.1 1.2 1.3 *Unit 3 Pyt...ile RAD 17/20
*u3sb1.py
from ti_system import get_key
#=====
cls()
text_at(13,"Press [esc] to end","center")
brightness.range(min,max)
while get_key() != "esc":
  *b=brightness.measurement()
  *cls()
  *text_at(7,"brightness =" + str(b),"left")
```