

**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)の機能とは異なり, 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。



**Teacher Tip:** この最初のプロジェクトは、明るさの値を読み取って画面に表示します。

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。

```
1.1 1.2 1.3 *Unit 3 Pyt...ile RAD 12/24
*u3sb1.py
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#-----
cls()
text_at(13,"Press [esc] to end","center")
while get_key() != "esc":
    block
```

```
1.1 1.2 1.3 *Unit 3 Pyt...ile RAD 16/22
*u3sb1.py
from ti_plotlib import text_at,cls
from ti_system import get_key
#-----
cls()
text_at(13,"Press [esc] to end","center")
while get_key() != "esc":
    b=brightness.measurement()
    text_at(7," brightness="+str(b),"left")
```



3. "brightness = " + str(b)について

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

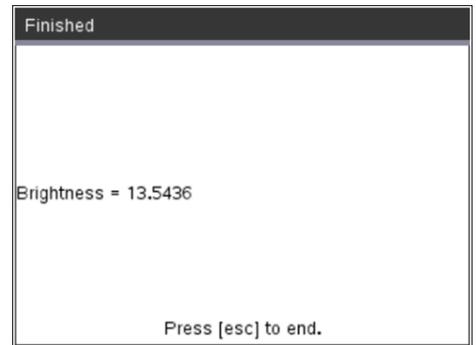
**+ 記号**は、"brightness ="という単語と文字**b**の値を組み合わせたものです。このタイプの文字列の追加は、コンカチネーション (concatenation, 連結)と呼ばれます。

この場合、位置合わせの**left**は**center**よりも優れています。**center**を使う場合は、線の長さが異なるため、新しいデータで古いデータが完全に消去され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。

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

Brightness = 13.5436

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てください。

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。

```
1.1 1.2 1.3 *Unit 3 Pyt...ile RAD 8/17
#=====
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)
```

**Teacher Tip:** 明るさの既定値範囲は[0,100]です。スマートフォンの懐中電灯は優れた光源になります。明るさの値を0にするのは難しい場合があります。

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てください。

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? 次のいくつかのレッスンでチェックしましょう。

```
1.1 1.2 1.3 *Unit 3 Pyt...ile RAD 17/20
#=====
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")
```



**Teacher Tip:** 範囲を設定すると、音や色などの他のTI-Innovator Hubデバイスでの作業がはるかに簡単になります。あるスケールから別のスケールに変換する必要がないからです。これについては、次のレッスンで説明します。

プログラム内で明るさの範囲を再定義することは可能ですが、これが必要になることはめったにありません。