



Unit 2: TI-Innovator™ Hubのforループ

Skill Builder 1: 光のループ

このレッスンでは、TI-Innovator Hubのコンテキストでのforループの概念を紹介します。

目標

- range()関数を使ったforループ作成
- float()を使って10進数値を入力
- カスタム点滅ライトの作成

コードの集まりを、指定された回数ループするときは、range()関数でforループを使えます。

range()関数は、既定値で0から始まり、1(既定値)ずつ増加し、指定された数で終わる一連の数値を返します。

```
for index in range(size):
    block
```

関数light.blink()では、点滅サイクルを十分に制御できません。ライトはどのくらい点灯し、どれくらい消灯させるか。10秒間に3回点滅させることはできるだろうか。

このレッスンでは、この制御と、点滅の総数を直接制御できるプログラムを開発します。

**Teacher Tip:** Pythonのforループは、シーケンス(つまり、リスト、タプル(tuple, 順序付けられた複数の要素で構成された組)、辞書、セット、文字列)の反復にも使われます。これについては、後のレッスンで説明します。

1. 新規のPython Hub Projectテンプレートから始めます。

3つの入力ステートメントを記述します。

- 点滅の総数に対して1つ
- onTime(LEDが点灯する時間)用に1つ
- offTime(LEDが消灯する時間)用に1つ

**Note:** 整数ではない点滅時間を設定したい場合があります。input()関数の前にint()を使う代わりに、float()を使います。これにより、小数で数値を入力できます。

float()とint()は、menu > Built-ins > Typeの中(メニュー>組み込み>タイプの中)にあります。

```
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key

blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
```

2. ステートメントを挿入します。

**for index in range(size):**  
**block**

これは、menu > Built-ins > Control(メニュー>組み込み>制御)から選択します。

つぎに、インラインプロンプト(行内入力要請)index, size, blockを置き換える必要があります。

```
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key

blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for index in range(size):
    block
```

**Teacher Tip:** indexは任意の変数にすることができます。sizeは0より大きい整数である必要があります。blockはループの各反復で処理されるコードのブロックです。ブロック内の各ステートメントはインデント(字下げ)されます。



3. **index**を任意の変数に置き換えます(*i*が一般的に使われます)。**tab**を押して、次のフィールドに移動します。

**size**を、最初の入力ステートメントで使われた変数**blinks**に置き換えます。もう一度**tab**を押して、**block**を強調表示します。

```

1.2 1.3 1.4 *Unit2 Py...ops RAD 13/27
*u2sb1.py
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#=====
blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for i in range(blinks):
  **block

```

4. forループの**block**は、ライト(TI-Innovator Hubの赤色LED)を操作する一連のステートメントです。**light.on()**、**light.off()**、**sleep()**ステートメントを使います。やってみましょう。inputステートメントで使ったタイミングの変数を使うことを忘れないでください。



すべての入力ステートメントに10進数の値を使えますが、点滅回数は整数に変換されます。まばたきを1/2にすることはできません!

次のステップは、完成したプログラムを示しています。

5. プログラムは次のようになります。

```

blinks = int( input("Number of blinks? ") )
onTime = float( input("ontime?") )
offTime = float( input("offtime") )
for i in range(blinks):
  light.on()
  sleep(onTime)
  light.off()
  sleep(offTime)

```

```

1.2 1.3 1.4 *Unit2 Py...ops RAD 17/31
*u2sb1.py
from ti_system import get_key
#=====
blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for i in range(blinks):
  **light.on()
  **sleep(onTime)
  **light.off()
  **sleep(offTime)
  **

```

6. プログラムを実行するときは、値を入力してからLEDを確認してください。プログラムが終了すると、ライトはオンまたはオフになりますか。

LEDは点滅しますが、TI-Nspire CX II画面では何も起こりません。forループ内にprintステートメントを追加して、現在の点滅番号を表示します。

### print(i)

**print()**はmenu > Built-ins > I/O(メニュー>組み込み>I/O)にあります。

プログラムを実行すると、LEDが命令どおりに点滅するため、シェル画面にインデックス変数*i*の値が表示されます。画面上の数字に何か変わったことに気づきましたか。

\*点滅番号は「点滅回数-1」です。

```

1.2 1.3 1.4 *Unit2 Py...ops RAD 13/32
*u2sb1.py
from ti_system import get_key
#=====
blinks = int( input("Number of blinks? ") )
onTime = float( #complete this...
offTime = float( #complete this...
for i in range(blinks):
  **print(i)
  **light.on()
  **sleep(onTime)
  **light.off()
  **sleep(offTime)
  **

```



**Teacher Tip:** `print()`も便利なデバッグツールです。

何が変わっていますか。 `range(size)`を使うPythonループは、0で始まり、 `size-1`で終わります。したがって、 `range(5)`では、処理される5つの数値は0, 1, 2, 3, 4です。

`range(start,stop)`を使ったPythonループは、開始値で開始し、 `stop-1`で停止します。

例：

```

doc.py 6/12 Python Shell 7/7
for i in range(1,5):
  print(i)
1
2
3
4
>>>
  
```

- 標準のPython `print()`関数の代わりに、Hub Projectテンプレートがインポートする、特別にTexas Instrumentsが開発した`text_at()`関数を使えます。プログラムの上部にあるimportステートメントを参照してください。

**menu > TI Hub > Commands**(メニュー>TI Hub>コマンド)から、`text_at(...)`を選択します。

### `text_at( row , "text" , "align" )`

`row`は、何行目に`text`を表示するかで、ここでは6行目にします。

“`text`”の代わりに、次の式にある`str(i)`を使います。

**Menu > Built-ins > Type**(メニュー>組み込み>タイプ)と`i`の整数値を文字列に変換します。

“`align`”は文字列を表示する横方向の位置を、3つのポップアップの選択肢の`left`(左)、`center`(中央)、`right`(右)のいずれかを選択します。

`print(i)`ステートメントは、`ctrl+t`を押してコメントに変換します。

次のようになります。

### `# print(i)`

### `text_at(6, str(i), "center")`

- プログラムを再度、実行します。ライトを10秒間に3回点滅させるには、どのような値を入力する必要がありますか。次を試してみましょう。

```

blinks = 3
onTime = 2
offTime = 2
  
```

3回目の点滅は開始10秒後に終了しますが、プログラムはさらに2秒間終了しません。他に選択肢はありますか。

作業内容を保存することを忘れないでください。

```

*u2sb1.py 13/27
from tiplotlib import text_at,cls
from ti_system import get_key
#=====
blinks = int( input("Number of blinks? ") )
onTime = float( input("on time? ") )
offTime = float( input("off time?") )
for i in range(blinks):
# print(i)
  text_at(6, str(i), "center")
  light.on()
  sleep(onTime)
  
```

