



Unit 2: TI-Innovator™ Hubのforループ

Application: コンピュータ音楽

この応用では、ランダムなコンピュータ音楽を生成する方法を学びます。

目標

- forループを使って音符の数を制御
- 乱数ジェネレーターを使ってランダムな音符を作成

ユニット2では、forループを使って光、色、音を制御しました。この応用では、コンピュータで生成されたランダムなサウンドまたは音楽を再生するプログラムを作成します。この課題には、3つのアプローチがあります。

- a) 純粋にランダムなトーン(周波数)を再生する。
- b) 特別な周波数を使ってランダムな音を再生する。
- c) 音符(リストにある)を使ってランダムな音を再生する。

また、トーン/音符にランダムな持続時間(タイミング)を使います。そして、さらに花を添えるため、音符にカラーLEDを使って異なる色を点灯させることができます。

1. 新規のPython Hub Projectを作成します。

乱数を生成する関数が必要になります。このツールは標準Pythonコマンドの一部ですが、Hub Projectテンプレートがインポートしない別のモジュールにあります。

menu > Random(メニュー>ランダム)を押して、次のステートメントを追加します。

from random import *

コード上部にあるインポートステートメントの集まりに追加します。

2. escを使ってプログラムを終了します。

menu > TI Hub > Commands(メニュー>TI Hub>コマンド)を押して、次のステートメントを選択します。

while get_key() != "esc":

残りのステートメントは必ずインデント(字下げ)してください。このプログラムを実行するときは、escを押してプログラムを終了します。

3. menu > Random(メニュー>ランダム)にあるrandint()関数を使います。

r = randint(min, max)

後で使うために整数の乱数を変数rに格納します。minとmaxは数字に置き換えられます。ただし、これらの数値を入力する前に、次の手順はどうなるかを検討してください。

```

1.7 1.8 1.9 *Unit 2 Py...ops RAD 9/19
# Unit 2 Application
#=====
from ti_hub import *
from math import *
from time import sleep
from tiplotlib import text_at,cls
from ti_system import get_key

from random import *
#=====

```

```

1.7 1.8 1.9 *Unit 2 Py...ops RAD 13/21
# Unit 2 Application
#=====
from ti_hub import *
from math import *
from time import sleep
from tiplotlib import text_at,cls
from ti_system import get_key

from random import *
#=====
while get_key() != "esc":
    block

```

1	Actions	2 P...ops	RAD	9/19
2	Run			13/23
3	Edit			
4	Built-ins			
5	Math			
6	Random	1	from random import *	
		2	random()	
		3	uniform(min, max)	
		4	randint(min, max)	
		5	choice(sequence)	
		6	randrange(start, stop, step)	
		7	seed()	
7	TI PlotLib			
8	TI Hub			
9	TI Rover			
A	More Mod			
var	B Variables			



4. r は音の周波数を表します。周波数はすべて聞こえる訳ではありません。非常に小さい周波数や非常に大きい周波数は、私たちの聴力範囲外にあるため、避ける必要があります。

前のレッスンで音楽を操作するときは、数百の周波数を使ったことを思い出してください。最小値と最大値を選択するときはそれを念頭に置きます。

別の確率変数 t (時間)を追加し、`menu > Random > uniform(min, max)`(メニュー>ランダム>一様(最小, 最大)にある`uniform()`乱数ジェネレーター)を使います。これにより、最小値と最大値の間のランダムな10進数が返されるため、一部の音符は1秒の一部で再生されます。ここでの最小値と最大値の選択は、各音符をどれだけ長く持続させるかによって異なります。

5. `sound.tone()`を使ってサウンドを作成し、プロンプトの`frequency`(周波数)と`time`(時間)をそれぞれ変数 r と t に置き換えます。

トーンの再生中にコンピュータを一時停止するには、`sleep()`関数を加えることを忘れないでください。プログラムはどのくらい待機する必要がありますか。

6. カラーLEDはどうでしょう。赤、緑、青の3つのカラーチャンネルがあり、値は0~255に制限されています。各チャンネルの`randint(0,255)`関数を使って、LEDを純粋にランダムな色で点灯させることができます。

`red = randint(0,255)`

または、周波数 r または時間 t 、あるいはその両方に色を依存させることができます。範囲外になることに注意します。たとえば、255を超える。

右図は完全なプログラムではありません。

```

1.6 1.7 1.8 *Unit 2 P...ops RAD 13/23
*u2app.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
from random import *
#=====
while get_key() != "esc":
    r=randint(min,max)
    t=uniform(min,max)

```

```

1.6 1.7 1.8 *Unit 2 P...ops RAD 13/22
*u2app.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
from random import *
#=====
while get_key() != "esc":
    r=randint(min,max)
    t=uniform(min,max)
    sound.tone(frequency,time)

```

```

1.6 1.7 1.8 *Unit 2 P...ops RAD 14/19
*u2app.py
from ti_plotlib import text_at,cls
from ti_system import get_key
from random import *
#=====
while get_key() != "esc":
    r=randint(min,max)
    t=uniform(min,max)
    sound.tone(frequency,time)
    color.rgb(red,green,blue)
    sleep()

```

```

1.6 1.7 1.8 Unit 2 Pyt...ops RAD 19/23
*u2app.py
from random import *
#=====
while get_key() != "esc":
    r=randint(min,max)
    t=uniform(min,max)
    sound.tone(frequency,time)
    red=randint(
    color.rgb(red,green,blue)
    sleep()

```