Unit 6: micro:bit with python

Skill Builder 1: ディスプレイ

このレッスンでは，micro:bitのディスプレイ(ボード前面にある5×5 LED)をさまざまな方法で制御する最初のPythonプログラムを作成します。このレッスンには2つのパートがあります。

Part1：エイリアンとの出会い
Part2：画像の表示

目標

- **.show( )**, **.scroll( )**, **.show(**image)を使ったmicro:bitのディスプレイ制御
- **sleep(***ms***)**を使ったディスプレイ速度制御

---

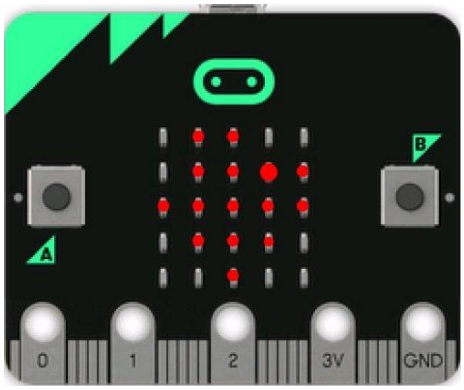
1. 始める前に，次のことを確認してください。
   - TI-Nspire CXⅡのOSは5.3以降ですか。
   - Pythonプログラミングに慣れていますか。それとも，ユニット1～5を完了しましたか。
   - **micro:bit**はTI-Nspire CXⅡに接続されていますか。
   - Micro:bit **Getting Started Guide**のセットアップ手順とファイル転送に従いましたか。

     https://education.ti.com/en/teachers/microbit

   このセットアップは1回だけ行う必要があります。アップグレードについては定期的に通知されます。

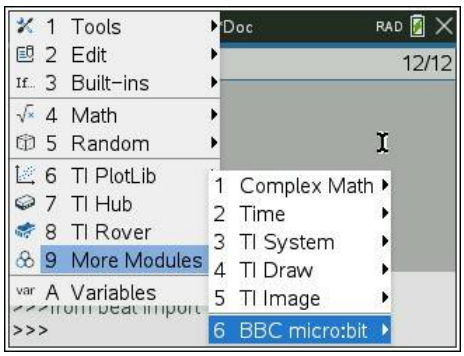
2. すべて正常にセットアップが行われ，TI-Nspireからmicro:bitに電力が供給されると，micro:bitは右のようになります。

   micro:bitのディスプレイにはTIロゴが表示されます。TIロゴは**Texas Instruments, Inc.**(テキサスインスツルメント社)の根拠地である州都ダラスがあるテキサス州の形です。



3. そして最後に…

   micro:bitモジュールがPythonライブラリにインストールされます。Pythonエディタで，**[menu] > More Modules** (メニュー＞その他のモジュール)を押すと，TIモジュールの下に**BBC micro:bit**が表示されます。



Note: OS 5.3以降では，デバイスの**Pylib**フォルダーに保存されているPythonモジュールが，ti_モジュールに加えてこのメニューに表示されます。リストは，ここの表示とは異なる場合があります。モジュールはファイル名のアルファベット順にリストされているため，**BBC micro:bit**はBではなく，リストのmの中にリストされています。

Teacher Tip: micro:bit入門
このユニットは，micro:bitとTI-Nspire電卓(またはコンピュータソフトウェア)が準備完了であることを前提としています。あなた(またはあなたの学生)が個々の電卓とmicro:bitのセットアップを完了していることを確認してください。TI Education Webサイトのダウンロードフォルダに含まれているGetting Started Guideのセットアップを必ず読んで実行してください。

https://education.ti.com/en/teachers/microbit.

すべての新しいソフトウェアとハードウェア同様，アップデートについてつねに情報を入手してください。

micro:bitが接続され，正しく機能していることを確認する，**my first program.tns**であるTI-Nspireの紹介ドキュメントもあります。

BBC micro:bit module (microbit.tns)はTexas Instruments社が提供する特別なモジュールです。各電卓のPylibフォルダーとTI-Nspire CXⅡコンピューターソフトウェアにインストールされます。それにより，**[menu] > More Modules > BBC micro:bit** (メニュー>その他のモジュール>BBC micro:bit)で表示されます。

Pythonの最初の5つのユニット：10 Minutes of Code(10分間のコード)カリキュラムは，このユニットの前に完了する必要があります。
ここで紹介するコーディングは複雑ではありませんが，TI-Nspire CXⅡでのPythonコーディングには申し分のないレベルです。

このユニットは，micro:bitバージョン1, 2で動作するよう設計されていますが，バージョン2に固有な機能には対応していません。
BBC micro:bitモジュールはこれらの機能をサポートしています。

オンラインで見つかるすべてのmicro:bitプロジェクト(micro:bitバージョン2プロジェクトを含む)は，Pythonと付属のmicro:bitを使ってTI-Nspire CXⅡで開発できます。しかし，2つの重要な違いがあります。

①micro:bitにバッテリーが接続されていても，プログラム実行中，micro:bitを電卓から切り離すことはできません。micro:bit runtime(.hexファイル)は，電卓の指示に従うように作成されています。電卓はmicro:bitに電力を供給して制御しています。

②オンラインで見つかるデモプログラムの多くは，While True:を使って無限ループを作成します。このループは，ダウンロードされた.hexファイルを使ってプログラムが別のプログラムに置き換えられるまで，micro:bitボード上で直接実行されます。TI-Nspireを使う場合，電卓はmicro:bitを制御するため，最も一般的に使用されるループは次のものです。

While get_key() != 'esc':

これにより，ユーザーは**[esc]**を押してプログラムを終了することができます。

重要：「micro:bit not connected (micro:bitが接続されていません)」というメッセージが表示された場合には，micro:bitを取り外してもう一度接続してください(リセット)。

4. **Part 1: alien encounter** (エイリアンとの出会い)**:** すべてのプログラミングのスタート同様，まずはmicro:bitディスプレイにメッセージを表示することから始めます。

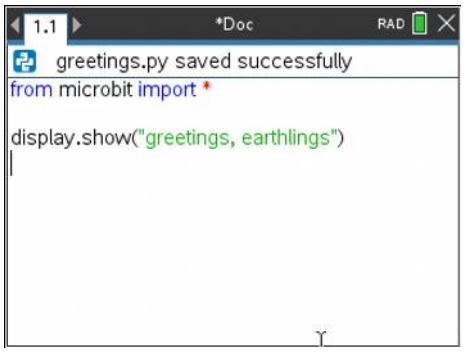
新規のTI-Nspireドキュメントで，**Add Python > New** (Pythonを追加>新規)を選択して，新規プログラムにgreetingsという名前を入力して開始します。Pythonエディタで，**[menu] > More Modules > BBC micro:bit** (メニュー>その他のモジュール>BBC micro:bit)を使って，メニュー上部の**import**ステートメントを選択します。

from microbit import *

Tip:‘micro:bit not connected (micro:bitが接続されていません)’というメッセージが表示された場合には，micro:bitを取り外してもう一度接続してください(リセット)。

Teacher Tip: micro:bitが取り付けられていない場合，プログラムの実行時，**import**ステートメント自体がエラーを報告します。Pythonはモジュラー言語(モジュラーは規格化された部品（モジュール）を組合せて機能を連結させる，という意味)であるため，必要に応じて補足機能が追加されます。このimportステートメントは，TINspire CXⅡを使ってmicro:bitを操作するため必要な関数やメソッドをロードします。また，他の標準モジュールやti_モジュールからいくつかの便利なメソッドをロードします。

5. micro:bitディスプレイにテキストメッセージを表示するには，次のステートメントを使います。



display.show(image or text**)**

このステートメントは次の場所にあります。

[menu] > More Modules > BBC micro:bit > Display > Methods
(メニュー>その他のモジュール>BBC micro:bit>表示>メソッド)

ステートメントは**display.show(**image or text**)**として挿入されますが，image or textは単なるプレースホルダー(実際の内容を後から挿入するため，とりあえず仮に確保した場所)であり，何かに置き換える必要があります。括弧内にメッセージ文字列を引用符で囲んでimage or textに置き換えます。

“greetings, earthlings” (こんにちは，地球の皆さん)

[ctrl] [R]を押してこのプログラムを実行すると，メッセージの文字が一度に1文字ずつディスプレイに表示されます。小文字‘e’は2回表示されますが，2つを区別することはできません。

間違えたら… 1.1ページに戻ってプログラムを修正してから，プログラムを再実行してください。表示したいテキストの前後の引用符を忘れていませんか。

6. メッセージを表示する良い方法は，次のとおりです。

display.scroll(“greetings, earthlings”)

これは次のところにあります。

[menu] > More Modules > BBC micro:bit > Display > Methods
(メニュー>その他のモジュール>BBC micro:bit>表示>メソッド)

.scroll()ステートメントを完成するには，**.show**ステートメントから文字列をコピーして貼り付けます。

前の**.show()**ステートメントを#commentにして(カーソルをその行に置き，**[ctrl] [T]**を押す)，無効にしてから，プログラムを再実行します。

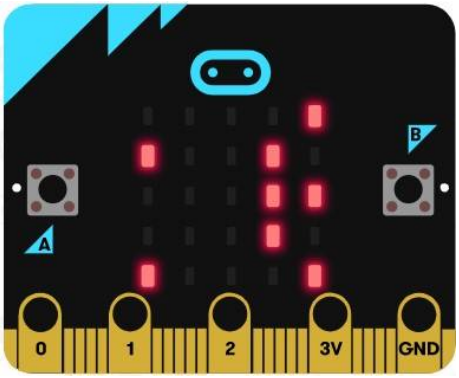
.showは**.scroll**に入力して変更することもできます。

7. **.scroll()**メソッドを使うと，メッセージが右から左にスクロール(移動)し，読みやすくなります。

delay =パラメータを追加すれば，スクロールの速度を制御できます。

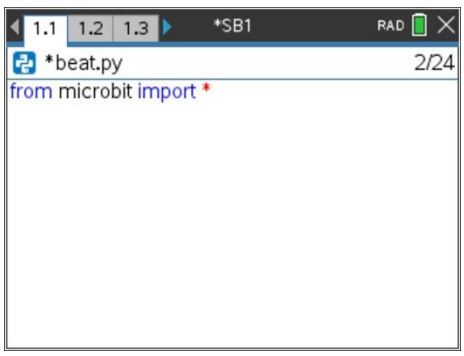
display.scroll(“greetings, earthlings”, delay = 200)

これは，スクロールを200ミリ秒(＝0.2秒)遅くします。他の値も試してください。



<greetings, earthlings.gif>

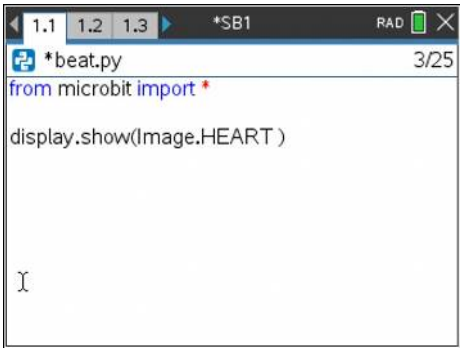
8. **Part 2: Be.Still.My.Beating.Heart** (まだ私の鼓動する心)**:** **[ctrl] [doc]**と押してページを挿入し，**Add Python > New** (Pythonを追加>新規)を選択して，新規Pythonプログラムを追加します(名前は**beat**です)。



Pythonエディタで，**[menu] > More Modules > BBC micro:bit** (メニュー>その他のモジュール>BBC micro:bit)を使って，リストのトップにある**import**ステートメントを選択します。

from microbit import *

9. micro:bitディスプレイにHEARTマークを表示します。次のステートメントを使います。



display.show( )

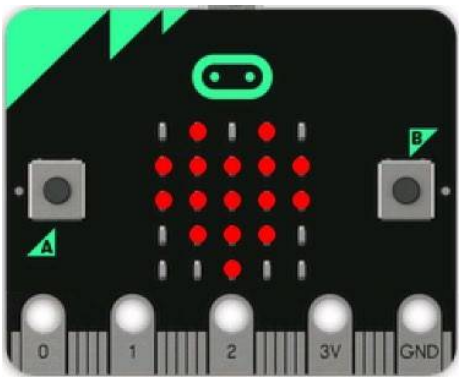
このステートメントは次のところにあります。
[menu] > More Modules > BBC micro:bit > Display > Methods
(メニュー>その他のモジュール>BBC micro:bit>表示>メソッド)

括弧内は，次を選択してプロンプトを置き換えます。

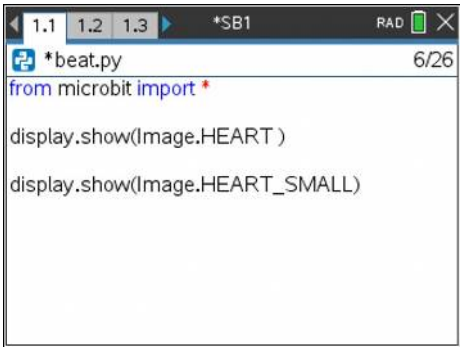
Image.HEART

このステートメントは次のところにあります。
[menu] > More Modules > BBC micro:bit > Display > Images > Set 1> Heart (メニュー>その他のモジュール>BBC micro:bit>表示>画像>セット1>Heart)

10. プログラムを実行(**[ctrl] [R]**)して，micro:bitの5×5 LEDグリッドに表示されるHEARTアイコンを確認します(右図)。この表示は，プログラムが完了した後も，何かが発生するまで残ります。



11. 前のページのプログラムエディタに戻り，小さなハートを表示する別のdisplayステートメントを追加します。

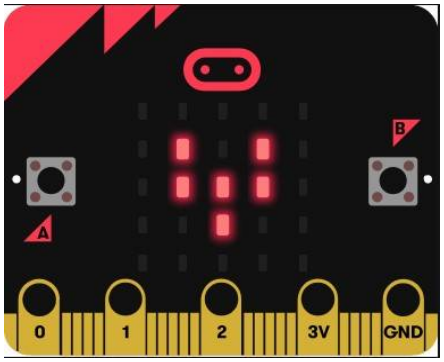


display.show(Image.HEART_SMALL)

この画像は，同じ**Images**メニューにあります。
[menu] > More Modules > BBC micro:bit > Display > Images > Set 1 (メニュー>その他のモジュール>BBC micro:bit>表示>画像>セット1)

Tip: 最初のdisplayステートメントをコピーして貼り付け，編集することもできます。**_SMALL**と入力します。_はアンダースコア(**[ctrl] [space]**と押す)で，文字は大文字(**[shift]+**英字)です。

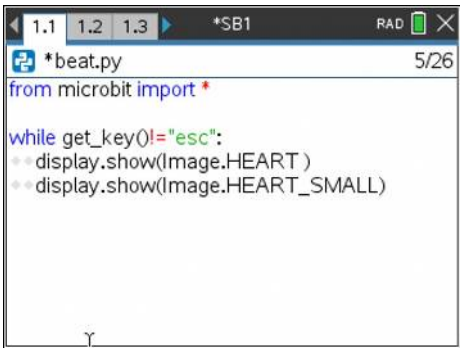
12. プログラムを再度，実行します。大きなハートを一瞬表示してから，小さなハートを表示します(右図)。



13. **ループ作成**: 2つのハートを点滅('beat')させるには，2つのdisplayステートメントをloop(ループ)で囲みます。2つのdisplayステートメントの前に次を挿入します。

while get_key() != "esc":

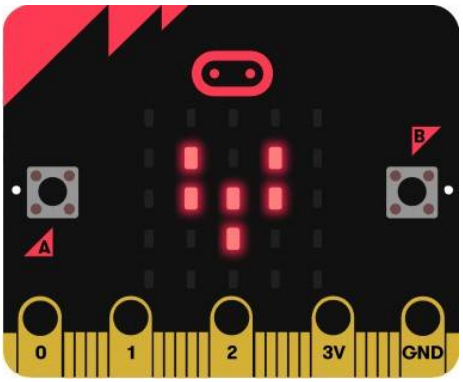
これは**[menu] > More Modules > BBC micro:bit > Commands** (メニュー>その他のモジュール>BBC micro:bit>コマンド)で選択します。そして，2つのdisplayステートメントをインデント(字下げ)して，loop本体を形成します。



重要なヒント：Pythonプログラムではインデント(字下げ)が重要です。これは，Pythonがloopブロックとifブロックを解釈する方法です。2つのdisplayステートメントが同じ数のスペースでインデントされていない場合，構文エラーが表示されます。**[space]**または**[tab]**を使って，2行を同じ量だけインデントします。インデントは，エディタでは薄い灰色のひし形(♦♦)で表示されます。

14. プログラムをもう一度実行してBeating Heart (鼓動する心)を見てください。**[esc]**を押してプログラムを終了します。

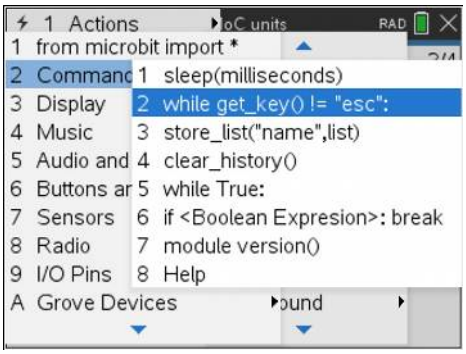
Tip: プログラムが無限ループに陥った場合，TI-Nspireの**[home/on]**を押したままにして，プログラムを中断してください。これは，**while True:**をCommandsメニューから不適切に使った場合に発生することがあります。このレッスンは，そのタイプの構造を回避します。



<beating_heart.gif>

15. micro:bitの**Commands**メニューには，他のメニューにもある便利なpythonコマンドがいくつか含まれています。micro:bitモジュールは，これらのPythonコマンドをインポートします。

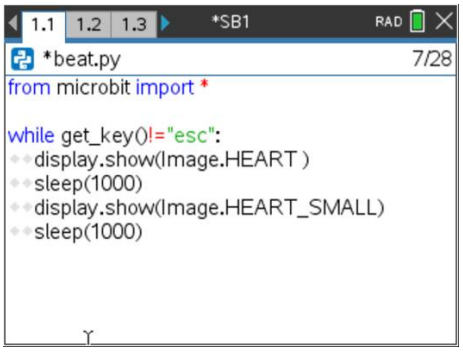
    他のPythonコマンドは他のメニューからも実行できます。BBC micro:bitメニューを使うだけでなく，適切なimportコマンドを提供する必要があることもあります。



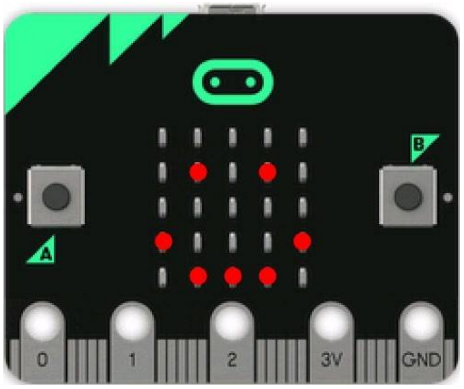
16. 心が鼓動する率(心拍数)を制御するには，各**display**ステートメントの後に2つの**sleep()**ステートメントを追加します。

    sleep(1000)は1000ミリ秒(＝1秒)の遅延を意味します。それは，**[menu] > More Modules > BBC micro:bit > Commands** (メニュー>その他のモジュール>BBC micro:bit>コマンド)で選択します。

    Tip: インデントに注意してください。



17. **類題 Making Faces:** 顔を表示してみましょう。**Beating Heart**(鼓動する心)同様のプログラム構造を使いますが，代わりにSet 1にある顔の画像を使います。



Teacher Tip: オンラインのmicro:bitレッスンでは，通常，Pythonでは**While True:** ループを使い，MakeCodeではforever: ループを使います。これらの無限ループは，micro:bitがオフになるか，別のプログラム(.hexファイル)に置き換えられるまで，micro:bit上で直接実行されます。TI-Nspireを使ってmicro:bitを制御する場合，電卓が完全に制御されるため無限ループは必要ありません。While True: はCommandsメニューにありますが，無限ループから抜け出すには，if(ブール式):breakと組み合わせて使う必要があります。

学生が電卓で無限ループに陥った場合，プログラムを中断するには**[on]**を押し続けます。コンピュータでPythonプログラムを中断する方法については，TIのmicro:bitソフトウェアに付属のGetting Started Guide(PDF)を参照してください。

Commandsメニューには，micro:bitのコーディング時に役立つ，よく使用される一般的なPythonコマンドが含まれています。micro:bitモジュールは，必要に応じて他のモジュールからこれらのコマンドをインポートします。ただし，すべてのpythonコマンドは他のメニューから利用でき，

micro:bitプロジェクトで使用できます。とはいうものの，適切なimportステートメントを使うことは重要です。

オプション：プログラムの最後にディスプレイをTI状態(テキサスのTIロゴ)にするには，次のステートメントを追加します。

display.show(ti) 小文字tiを入力します。
Whileループの終了後に置きます(字下げを解除(dedent))。

これは必須ではありません。電卓画面と同様，プログラムがmicro:bitディスプレイで終了したことを示すのに役立ちます。

micro:bitプログラムの開始時(whileループが始まる前)に，print(“running…(実行中...)”)のようなものを電卓画面に表示すると役立つ場合があります。

sleep( )について: micro:bit用に大幅に変更されたPython関数の1つは，timeモジュールにあるsleep()ステートメントです。通常，引数の値は秒を表しますが，micro:bitモジュールは値がミリ秒(1000分の1秒)を表すようにsleepメソッドを変更します。micro:bitモジュールをインポートした後，sleep(1000)は1秒の遅延を表します。sleep()はmicro:bit自体ではなく，電卓プログラムの速度を制御することに注意します。このため，from microbit import *は，インポート時間(または時間をインポートする可能性のあるモジュール)の後に必ず実行します。

.show()メソッドには，**delay =**と**wait =**の2つのオプションパラメータがあります。

display.show(value, delay=xxx , wait=False/Ture)

delay value(遅延値)はミリ秒単位であり，表示が遅くなります。

wait=tureはアニメーションが終了するまで処理をブロックします。終了しない場合，アニメーションはバックグラウンドで実行されます。

以下に示したものは画像(image)の一覧です。参考にしてください。

Set 1

1:HEART
2:HEART_SMALL
3:HAPPY
4:SMILE
5:SAD
6:CONFUSED
7:ANGRY
8:ASLEEP
9:SURPRISED
A:SILLY
B:FABULOUS
C:MEH

Set 2

1:YES
2:NO
3:TRIANGLE
4:TRIANGLE_LEFT
5:CHESSBOARD
6:DIAMOND
7:DIAMOND_SMALL
8:SQUARE
9:SQUARE_SMALL
A:RABBIT
B:COW
C:TI LOGO

Set 3

1:MUSIC_CROTCHET
2:MUSIC_QUAVER
3:MUSIC_QUAVERS
4:PITCHFORK
5:XMAS
6:PACMAN
7:TARGET
8:TSHIRT
9:ROLLERSKATE
A:DUCK
B:HOUSE
C:TORTOISE

Set 4

1:BUTTERFLY
2:STICKFIGURE
3:GHOST
4:SWORD
5:GIRAFFE
6:SKULL
7:UMBRELLA
8:SNAKE

Beating Heartと同じプログラム構造を使う，よく似たプロジェクトはMaking Facesですが，**Set 1**の顔アイコンを使います。