

Unit 6: Rover座標

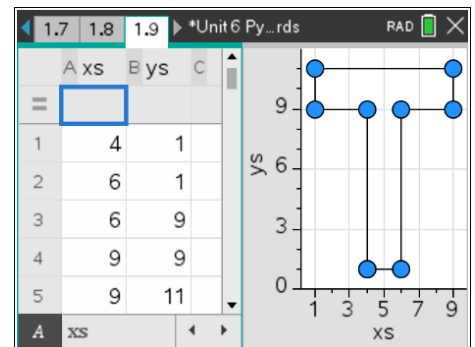
Skill Builder 3: 形を作る

このレッスンでは、デザインされた2次元の図形を作成するプログラムを作成します。リストを使ってプログラミングし、ループを使ってRoverに図形を紙に描画(または単にドラッグ)させます。

目標

- Lists & Spreadsheetアプリでリストを作成
- Pythonでrecall_listを使用
- forループを使ってリスト内の要素を処理
- pauseステートメントを使って処理を制御

このプロジェクトでは、独自のデザインの形の頂点のx座標, y座標を表す、2つのリストを作成する必要があります。このレッスンでは、右に示す文字Tのデザインを使います。目標は、Roverにマーカーを使ってこのデザインを作成させることです(あるいは、マーカーが利用できない場合はルートをたどります)。



1. プログラムを作成する前に、TI-Nspire CX IIのLists & Spreadsheet(リストとスプレッドシート)アプリの2つのリストに形の座標を入力します。T字形のすべての座標は、右図のとおりです。リスト名はxsとysです。これらの名前はプログラムにとって重要です。

A	xs	B	ys	C
1	4	1		
2	6	1		
3	6	9		
4	9	9		
5	9	11		
6	1	11		
7	1	9		
8	4	9		
9	4	1		
10				

TI-Nspire CX IIのGraphsアプリで散布図を設定するか、Data & Statisticsアプリ(Lists & Spreadsheetアプリのクイックグラフ)を使えば、値をテストできます。

2. 新規のPython Rover Codingプロジェクトを開始します。

最初の2つの新しいステートメントは、TI-Nspire変数からリストを取得し、それらを2つのPython変数に格納します。同じ変数名を使ってもかまいませんが、実際には同じ変数ではありません。

menu > More modules > TI System(メニュー>その他のモジュール>TIシステム)から、次のステートメントを選択します。

recall_list("name")

これらのステートメントは2つ必要なので、ステートメントを選択してコピーして貼り付けます(メニューからもう一度取得してもかまいません)。

```

import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *

#=====
list=recall_list("name")

list=recall_list("name")
    
```



3. これらのステートメントの1つはリストと名前に`xs`を使い、もう1つのステートメントでは`ys`を使います。

これらの2つのステートメントは、TI-Nspire CX II からリスト`xs`と`ys`を取得し、それらをそれぞれPython変数`xs`と`ys`(=記号の左側)に格納します。

```

Unit 6 Pyt...rds
RAD
* u6sb3.py 1/22
# Unit 6 SB3 - Make the Shape
#=====
import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *
#=====
xs=recall_list("xs")

ys=recall_list("ys")

```

4. これで、Roverのルートプログラムする準備が整いました。Roverは点(0, 0)からスタートすることを忘れないでください。最初の点が原点ではない可能性があるため、マーカーの挿入を一時停止する前に、Roverを最初の点に移動させます。最初の点は`xs[0]`, `ys[0]`です。次のステートメントを使います。

`rv.to_xy(xs[0], ys[0])`

ブラケット`[]`は、0キーの左側にある左括弧キーにあります。

```

Unit 6 Pyt...rds
RAD
* u6sb3.py 2/22
#=====
import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *
#=====
xs=recall_list("xs")

ys=recall_list("ys")
rv.to_xy(xs[0], ys[0])

```

5. マーカーを挿入している間(マーカーがある場合)、処理を一時停止するステートメントを追加します。

`input("Press enter to continue.")`

`input(「続行するにはenterを押してください。」)`

```

Unit 6 Pyt...rds
RAD
* u6sb3.py 13/22
import ti_rover as rv
from math import *
import ti_plotlib as plt
from ti_system import *
from time import *
#=====
xs=recall_list("xs")

ys=recall_list("ys")
rv.to_xy(xs[0], ys[0])
input("Press enter to continue.")

```

6. **for**ループを作成して、残りの点に移動します。

**`for i in range(1, len(xs)):`
`block`**

`len(xs)`はリスト`xs`の長さ(要素数)です。長さが12のとき、つぎにループはリストの最後の要素である値`i = 11`で終了します。

```

Unit 6 Py...rds
RAD
* u6sb3.py 6/24
from ti_system import *
from time import *
#=====
xs=recall_list("xs")

ys=recall_list("ys")
rv.to_xy(xs[0], ys[0])
input("Press enter to continue.")

for i in range(1,len(xs)):

```

