Unit 6: Rover座標

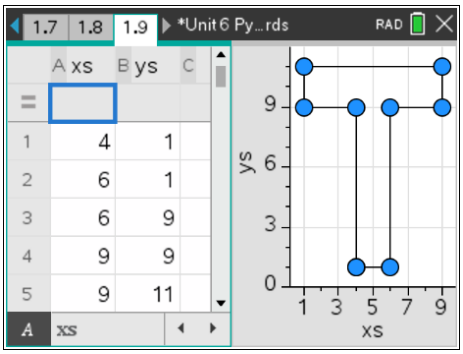
Skill Builder 3: 形を作る

このレッスンでは，事前にデザインされた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に紙に図形を描画(または単にドライブ)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，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字型のすべての座標は，右図のとおりです。2つのリスト名は**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。

   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)を使って，値をテストできます。

	A xs	B ys
=		
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		

Teacher Tip: Lists & Spreadsheetアプリに慣れていない場合は，Pythonプログラムで2つのリストを作成できます。

xs = […]
ys = […]

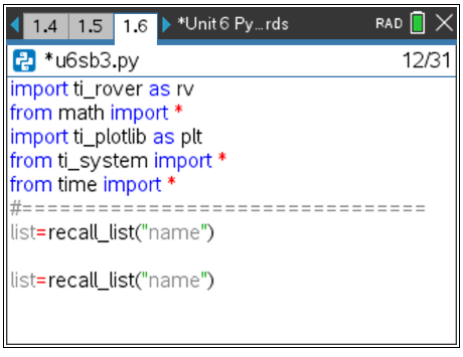
ただし，このレッスンで使う方法は，PythonとTI-Nspire CX IIの間で変数を転送する機能です。

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

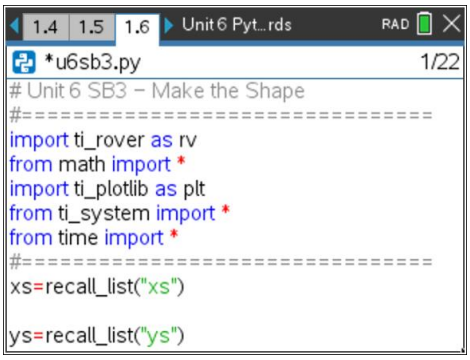
   最初の2つの新しいステートメントは，TI-Nspire変数からリストを取得し，それらを2つのPython変数に格納します。同じ変数名を使ってもかまいませんが，実際には同じ変数ではありません。
   menu > More modules > TI System(メニュー>その他のモジュール>TIシステム)から，次のステートメントを選択します。

   recall_list(“name”)

   これらのステートメントの2つが必要なので，ステートメントを選択してコピーして貼り付けるだけです(メニューからもう一度取得してもかまいません)。



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。



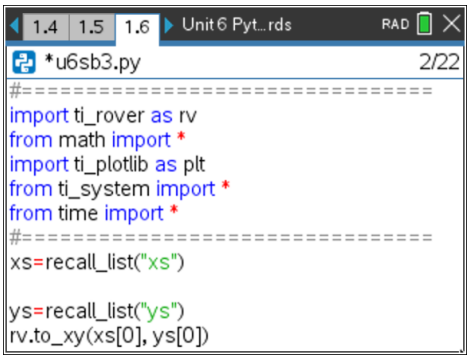
これらの2つのステートメントは，TI-Nspire CX IIからリスト**xs**と**ys**を取得し，それらをそれぞれPython変数**xs**と**ys**(＝記号の左側)に格納します。

Teacher Tip: Python変数とTI-NspireCX II変数は同じ名前の必要はありませんが，これは理にかなっています。
TIシステムメニューのステートメント**store_var(list, “name”)**は，その逆を行います。つまり，PythonリストからTI-Nspire CXII変数“name”に値を転送します。ツールチップは，どれがどれであるかを明確に示しているため，このステートメントで特に役立ちます。

4. これで，Roverのルートをプログラムする準備が整いました。Roverは点(0, 0)から始ま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])なので，次のステートメントを使います。

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

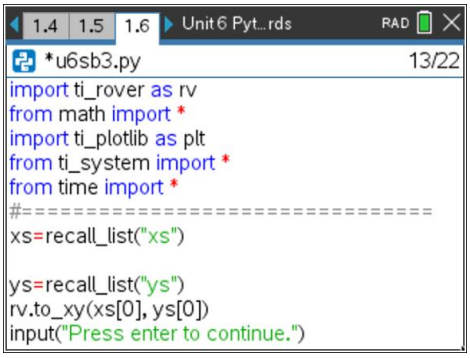
[ ]は，0キーの左側にある左括弧キーにあります。



Teacher Tip: 電卓で選択すると両方の角かっこが表示されますが，どちらか一方だけが必要な場合はどちらかを削除します。
Pythonには，2つのリストを繰り返す方法がいくつかあります。ここで紹介するものは初心者に適しています。

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。

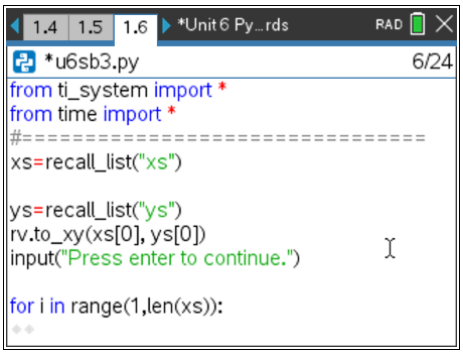
input(“Press enter to continue.”)



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(サイズ)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。

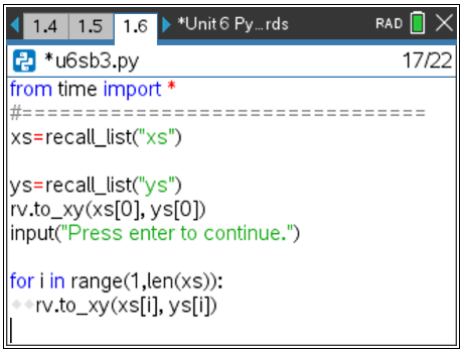


Teacher Tip: xsとysは同じサイズである必要があります。

7. 次のステートメントを使って，ループの**block**を完了します。

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

ループの後(インデント解除)，再び一時停止してマーカーを削除し，Roverを東向きの原点(0, 0)に戻します。



8. 課題：**ti_plotlib**ツールを使って，Roverと同期して画面上に形状の座標もプロットします。**plt.auto_window(xs,ys)**を使って，点のウィンドウを設定します。最後に，関数**plt.plot(xs,ys,"mark")**を使って点が結ばれた散布図を描画できます。