Unit 5: Roverセンサ

Skill Builder 2: Off the Wall (壁を離れて)

このレッスンでは，TI-Innovator Roverの超音波Rangerを使って障害物を回避します。

目標

- Rover前の物体までの距離読み取り
- 距離が小さいとき，向きを変えて運転続行

Teacher Tip: このプロジェクトでは，Roverを使うとき，かなり一般的な問題に遭遇します。TI-Nspire CX IIは，Roverよりはるかに高速に動作します。前ユニットのLEDレッスンでは，TI-Nspire CX IIはRoverが順方向または方向転換コマンドを終了するまで待機してから，コード内の次の命令を出す必要がありました。
ここでも同じ問題が発生しますが，その影響(エラー)はより劇的です。Roverは，移動，方向転換，停止の間で苦労しています。(適切な場所に)別の待機を追加すると，問題は解決します。

Roverがドライブしているとき，プログラムは超音波Rangerを使って前方の障害物を検出し，Roverに割り込んで別のルートをたどることができます。このプロジェクトでは，Roverは2つの向かい合った壁の間を行ったり来たりします。

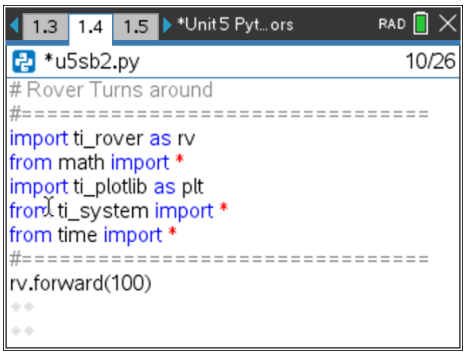
Teacher Tip: このプロジェクトは，ハードカバーの本を並べて使えばうまく機能します。本が垂直方向と水平方向の両方でRoverに対して垂直であることを確認してください。そうしないと，(サイレント)トーンがセンサに直接跳ね返る(エコーする)ことがないため，Rangerは障害物を認識しません。

1. Rover Codingテンプレートを使って新規のPythonプログラムを開始します。

   次のコマンドを使って，Roverを動かします。

   rv.forward (100)

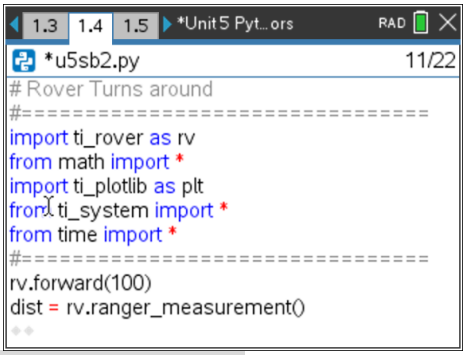
   それは10m(約39フィート)ですが，途中で障害物があります。



2. Roverの前の距離を読みます。

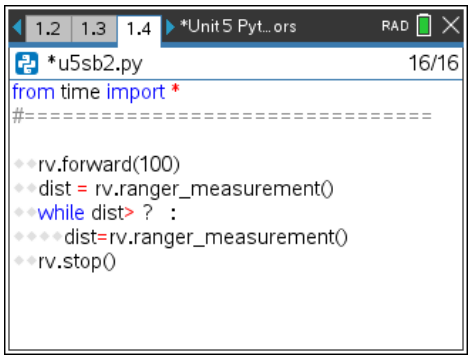
   dist = rv.ranger_measurement()

   続行できる限り，続行します。しかし，Roverが障害物に近すぎる場合は，向きを変えてください。「近すぎる」とはどのくらいでしょう。それはあなたが決定してください。



Teacher Tip: Roverは向きを変えるのに約20cm必要です。

3. **while**ループを追加して，障害物が遠くにある限り，距離の監視を続けます。



while dist > ? :

(疑問符(?)を小さい数字に置き換えます。)

dist = rv.ranger_measurement()

今のところ，ループが終了したらRoverを停止します。

rv.stop()

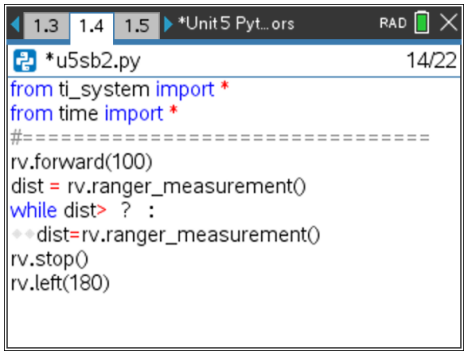
rv.stop()は，**menu > TI Rover > Drive(**メニュー>TI Rover>ドライブ)にあります。

Roverを壁(40フィート(約12.19m)未満)に向けてプログラムをテストし，プログラムを実行します。Roverは壁にぶつかる前に停止する必要があります。

Teacher Tip: 手も障害物としてうまく機能します。
rv.stop()はRoverの運転を即座に停止し，ドライブキュー内のすべてのコマンドをクリアします。

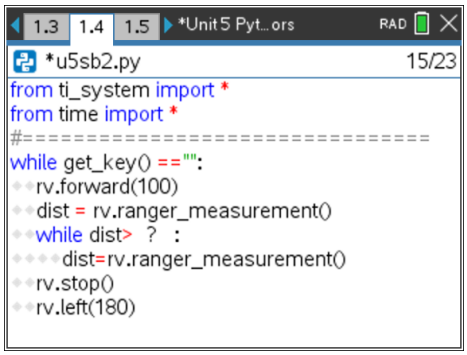
上記の原則：Roverを動かしてから，距離をモニター(監視)します。driveコマンドはキューに入れられますが，**measurement()**はキューに入れられません。

4. Roverが障害物に遭遇すると，Roverは向きを変える必要があります。Roverが停止したら，次を使ってRoverを回転させます。



rv.left(180) または **rv.right(180)**

5. 向きを変えた後，キーが押されるまで反対方向(**forward**(前方))に進みます。Roverが動いていることを忘れないでください。そのため，**esc**を押すのは難しいかもしれません。
次のステートメントを置きます。



while get_key() != “esc”:

これは，コードの先頭にあります。
ステートメントを編集して，任意のキーを使ってプログラムを停止できるようにします。

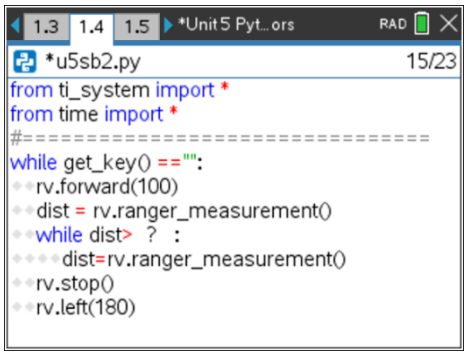
while get_key() == “”:

(==は等しいを表し，“”は何もないことを表します)

このwhileステートメントの下にあるすべてのステートメントをインデント(字下げ)して，**while block**にします。ステートメントのグループをインデントする最も簡単な方法は，**shift+**下矢印を使ってそれらのステートメントをすべて選択し，**tab**を押すことです。

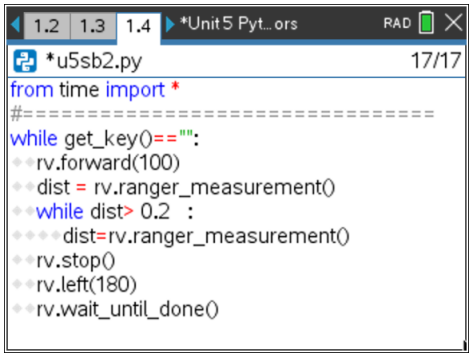
Teacher Tip: これは，コードが正しく機能しない点です。**left(180)**コマンドが処理されると，制御はすぐにループの先頭に戻り，Roverはまだ壁に近すぎて，Roverが指示に従うよりも速くプログラムが実行されるため，実際には移動しません。モーターが苦労しているのが聞こえます。これは，ランタイムエラーの良い例です。コンピュータに関する限りすべてのコードは正しいですが，物理的な影響は望ましくありません(間違っています)。

6. プログラムを実行します。Roverは壁に近づき，立ち止まり，向きを変えます(十分なスペースがある場合)。待ってください－何かがおかしいです。Roverが動けなくなりました!



向きを変えるにはRoverは1，2秒かかりますが，プログラムはすぐにループの先頭に戻り，Roverに前進するように指示してから，壁を再び検出して停止します。Roverは向きを変え，前進し，停止します。コマンドがほぼ同時に入ってくるので，Roverは圧倒されます。モーターが苦労しているのが聞こえますが，Roverは動きません。すべてが速すぎて，何かが行われなければRoverは神経衰弱を起こします…まもなく!

7. 任意のキーを押してプログラムを終了します。
Roverはまだしばらく苦労するかもしれません。その場合は，電卓からRoverのプラグを抜き，電源の電源をオフにしてから，再接続して再度オンにします。



プログラムの問題を修正するには，次のステートメントを追加して，Roverが向きを変え終わるまでTI-Nspire CX IIを待機させます。

rv.wait_until_done()

このステートメントを**rv.left(180)**ステートメントのすぐ下に置きます。

プログラムを再び実行すると，Roverは2つの向かい合った壁の間を楽しく移動します。

Roverは向きを変えるのに約20cm必要です。Roverの後部は船首よりも長いです。**while dist > 0.2:** これを適切に処理します。