



#### Unit 4: ドライブ機能

#### Application: Custom Polygons

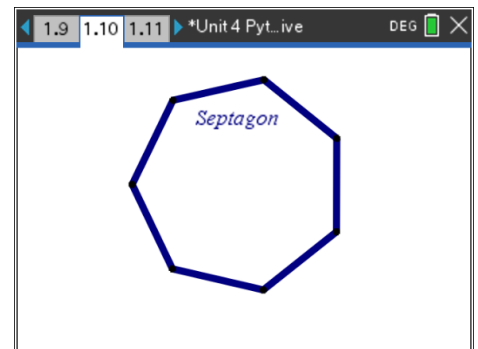
この応用レッスンでは、正多角形メーカーを設計することで、Roverの運転免許証を取得します。

#### 目標

- `input()`ステートメントを使って正多角形の頂点の数と各辺の長さのデータを入力
- 側面と角に沿ってライトを表示

正方形と五角形の経路を正常にドライブ(運転)したので、Roverの運転免許試験を受ける準備が整いました。

正多角形の頂点の数と各辺の長さを入力し、そのルートをドライブし、途中でまばゆいばかりの色で空を照らすプログラムを作成します。単にRoverを動かすだけでなく、プログラムはユーザーからの入力を求め、各辺に適切な距離を移動し、各頂点で回転する適切な角度を計算します。



**Teacher Tip:** 多角形の外角の和が $360^\circ$ であることから回転角は $360/(\text{頂点の数})$ 度になります。

1. 前レッスンのpentagon(正五角形)プログラムのコピーを作成します。コードは右図のようにになっているはずですが、このコードにいくつかの追加と変更を加えます。

```

1.9 1.10 1.11 *Unit4 Pyt...ive DEG
# u4app.py 26/26
#=====  

# colorful pentagon:  

for i in range(5):  

  rv.color_rgb(0, 10 + 60 * i, 0) # green sides  

  rv.forward(1)  

  rv.wait_until_done()  

  rv.color_rgb(10 + 60 * i, 0, 0) # red vertices  

  rv.left(72)  

  rv.wait_until_done()  

rv.color_rgb(0,0,0)

```

2. `for`ループの前に、頂点の数(n)と各辺の長さ(s)を入力する2つの入力ステートメントを記述します。

**n** = .... (入力ステートメント, 頂点の数)

**s** = .... (入力ステートメント, 辺の長さ)

頂点や辺の長さなど、より分かりやすい変数名を使うことをお勧めしますが、Pythonの予約語(`and`, `import`, `try`など、ユーザーが変数名や関数名として使用できない文字列)は使わないよう注意してください。

```

1.9 1.10 1.11 *Unit4 Pyt...ive RAD
# u4app.py 11/21
#=====  

n = .....  

s = .....  

# colorful polygon:  

for i in range(5):  

  rv.color_rgb(0, 10 + 60 * i, 0) # green sides  

  rv.forward(1)  

  rv.wait_until_done()  

  rv.color_rgb(10 + 60 * i, 0, 0) # red vertices  

  rv.left(72)

```



3. ループステートメントには、変更する必要がある3つの値があります。

`range( ? ), forward( ? ), left( ? )`

これらの3つの引数を編集した後、プログラムを実行し、入力ステートメントの値を入力してプログラムをテストします。

Roverを注意深く観察するか、マーカーホルダーにマーカーを挿入して紙に多角形を描きます。

両側に沿ったLEDの明るさの式を使った場合、頂点の数の変化を考慮して、それらのLEDステートメントも調整する必要があります。赤や緑だけでなく、さまざまな色を使うことをお勧めします。ランダムな色を試してみましょう。

```

1.9 1.10 1.11 *Unit 4 Pyt...ive RAD 21/21
*u4app.py
S= .....
# colorful polygon:
for i in range(5):
  rv.color_rgb(0, 10 + 60 * i, 0) # green sides
  rv.forward(1)
  rv.wait_until_done()
  rv.color_rgb(10 + 60 * i, 0, 0) # red vertices
  rv.left(72)
  rv.wait_until_done()
rv.color_rgb(0,0,0)

```

#### Teacher Tip: 考えられる解

```

1.9 1.10 1.11 *Unit 4 Pyt...ive RAD 21/21
*u4app.py
=====
n= int(input("Vertices? "))
s= int(input("Length of sides? "))
# colorful polygon:
for i in range(n):
  rv.color_rgb(0, 10 + 245 / n * i, 0) # green
  rv.forward(s)
  rv.wait_until_done()
  rv.color_rgb(10 + 245 / n * i, 0, 0) # red
  rv.left(360 / n)
  rv.wait_until_done()
rv.color_rgb(0,0,0)

```

LEDには注意が必要です。範囲外になることは避けてください。

挑戦が必要なら、次のルートプログラミングしてみてください。

