プログラミング教室のテクノロ

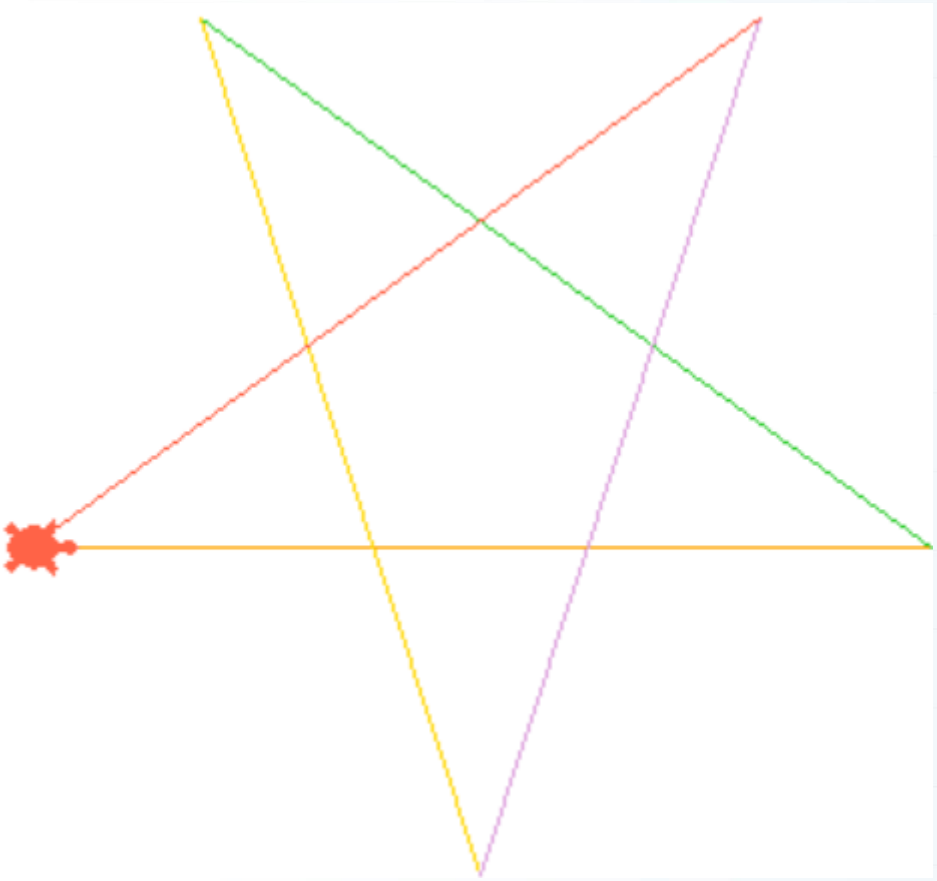
Pythonの道

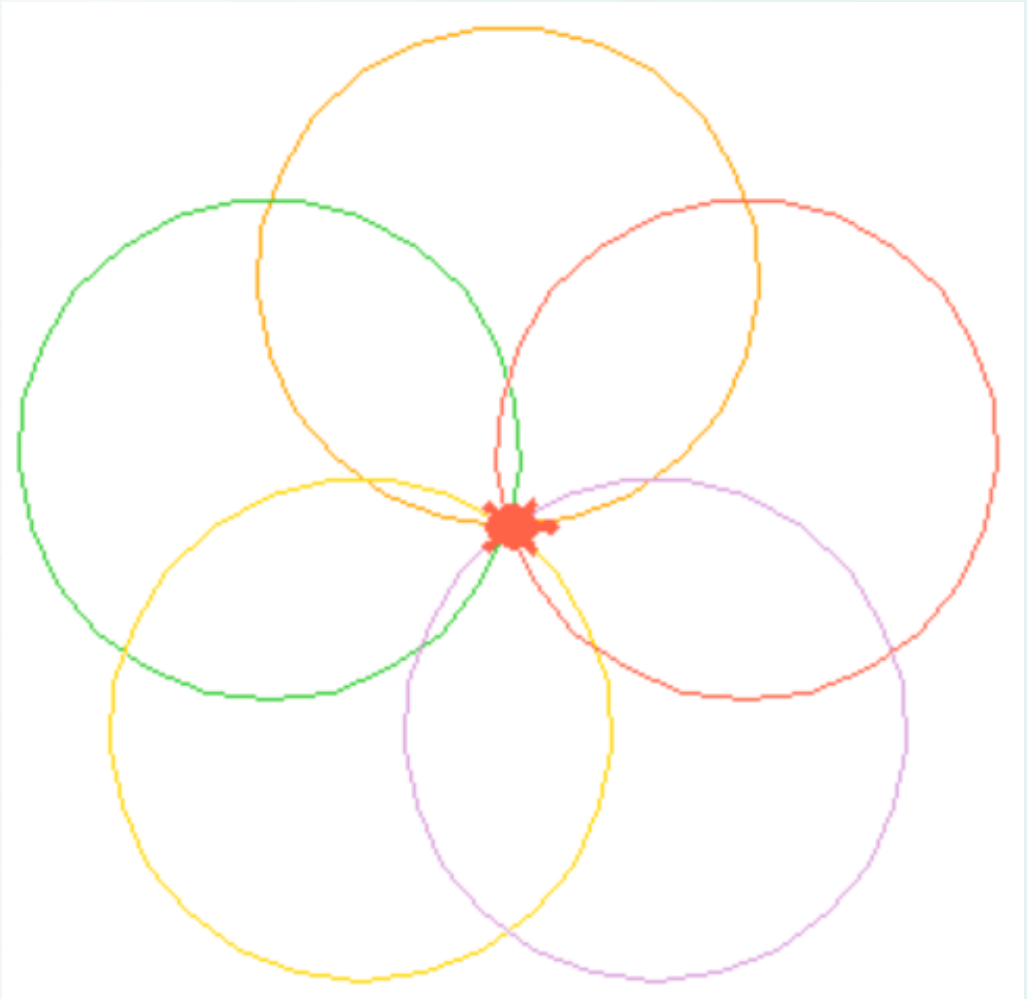
タートルグラフィックス①

もくじ

タートルグラフィックスを知る

タートルグラフィックスとはカメを動かして絵をかくツールだよ。
タートルグラフィックスを使っていろんな図形を書いてみよう。





直線を引く

まずはタートルグラフィックスで直線を引いてみよう。

線を引くプログラム

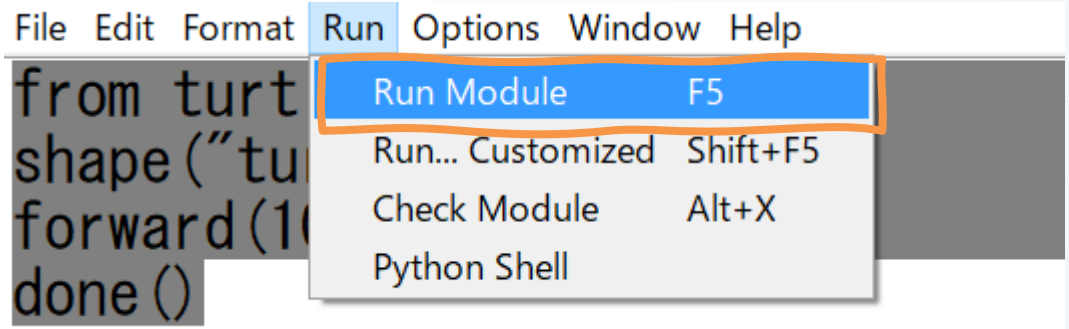
```
from turtle import *　・・・・タートルグラフィックスの呼出し
shape("turtle")　　・・・・カメを選択
forward(100)　　・・・・前に100進む
done()　　・・・・終了
```

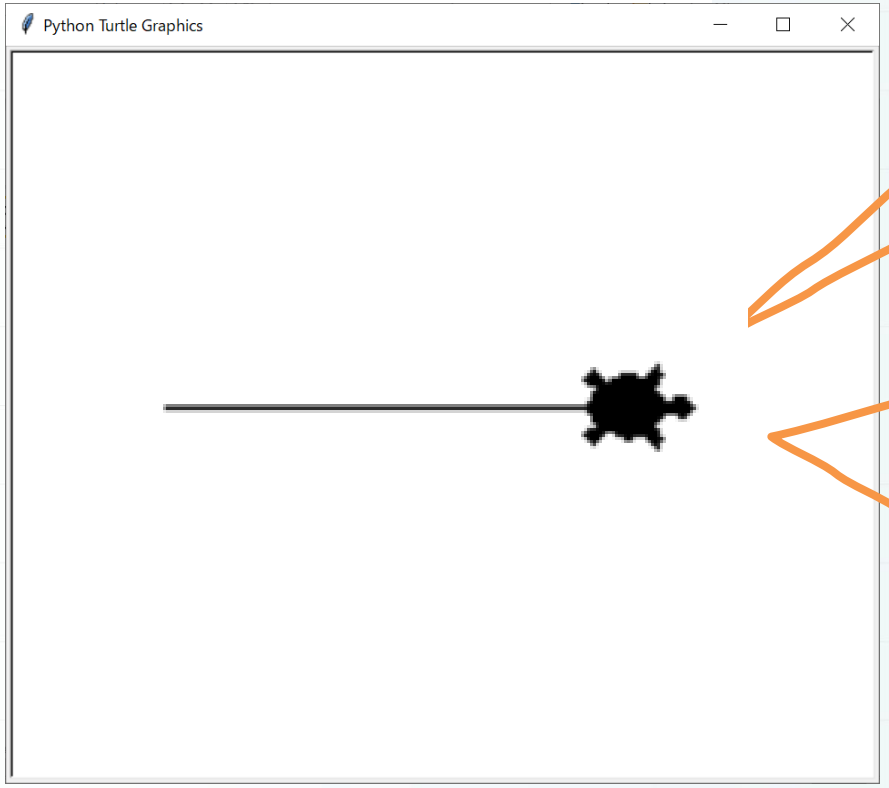
単語帳　～英語も一緒に覚えてしまおう！～

■from:～から
■turtle:カメ
■import:取り込む。輸入する。
■shape:形　姿　輪郭
■forward:前
■done:～した　※doの過去分詞形

直線を引く

「Run Module」またはF5キーを押してプログラムを実行する。





カメが前に向かって100進むことを確認する。

カメ以外にも
arrow
circle
square
triangle
classic
のアイコンも用意されている

from文を理解する

from文を使えば拡張機能を取り込むことができる。つまり、Pythonにない機能を使うことができるようになる。

拡張機能の取り込みはscratchの「拡張機能を選ぶ」ボタンと同じ

コードの書き方

```
from 拡張機能 import ＊
```

fromの後ろに呼び出したい拡張機能の名前を書く。
importの後ろの＊（アスタリスク）は拡張機能がもつすべての機能を呼び出すという意味。

正三角形を書く　その１

次に正三角形を書いてみよう。

正三角形を描くプログラム

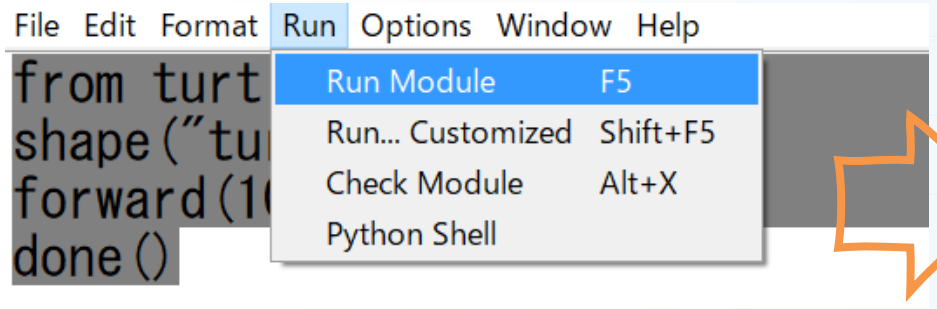
```
from turtle import *      ・・・・タートルグラフィックスの呼出し
shape("turtle")           ・・・・カメを選択
forward(100)              ・・・・まっすぐ100進む
left(120)                 ・・・・120度曲がる
forward(100)              ・・・・まっすぐ100進む
left(120)                 ・・・・120度曲がる
forward(100)              ・・・・まっすぐ100進む
left(120)                 ・・・・120度曲がる
done()                    ・・・・終了
```

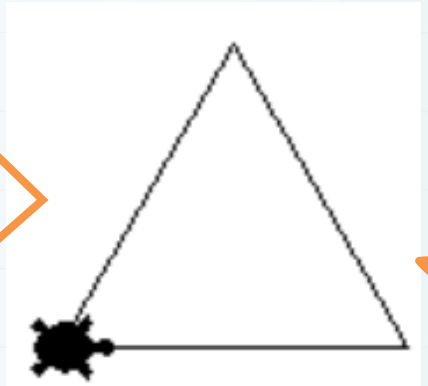
正三角形を書く　その１

単語帳　～英語も一緒に覚えてしまおう！～

■from:～から
■turtle:カメ
■import:取り込む　輸入する。
■shape:形　姿　輪郭
■forward:前
■left:左
■done:～した　※doの過去分詞形

「Run Module」またはF5キーを押してプログラムを実行する。





カメが前に100進む。
120度向きを変える。
前に100進む。
120度向きを変える。
前に100進む。

正三角形を書く　その２

もっとシンプルなプログラムで正三角形を書いてみよう。

正三角形を描くプログラム

```
from turtle import *          ・・・・タートルグラフィックスの呼出し
shape("turtle")               ・・・・カメを選択
for i in range(3):            ・・・・以下の2行を3回くり返す
    forward(100)              ・・・・まっすぐ100進む
    left(120)                 ・・・・120度曲がる
done()                        ・・・・終了
```

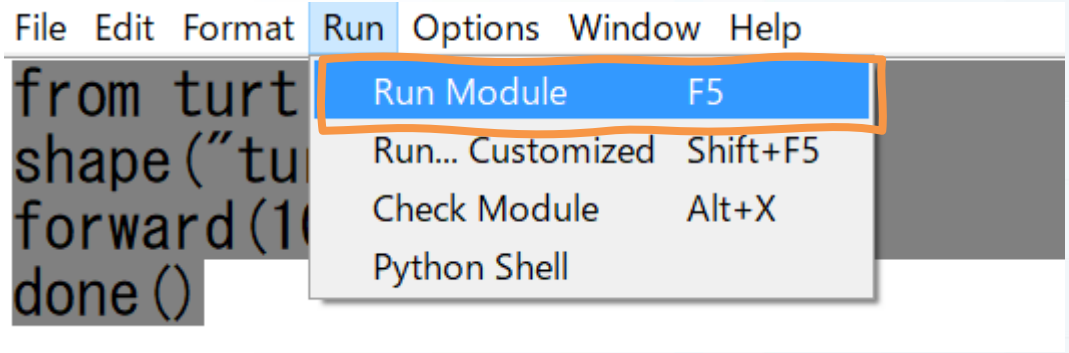
3回くり返す

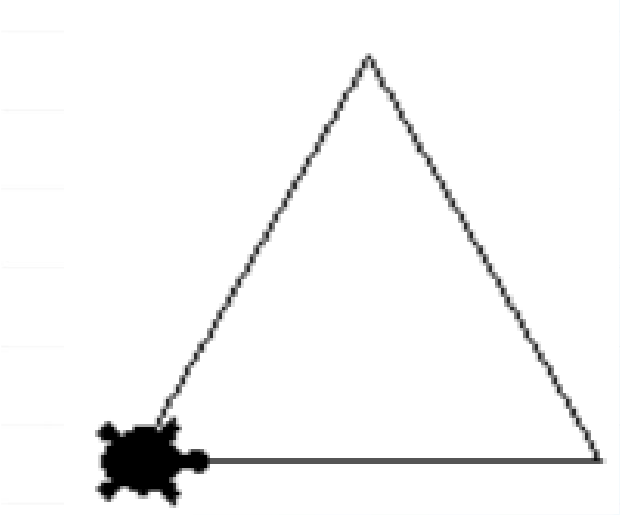
単語帳　～英語も一緒に覚えてしまおう！～

■from:～から　■turtle:カメ　■import:取り込む。輸入する。
■shape:形　姿　輪郭　■for:～のために　■in:～の中に
■range:範囲　■forward:前　■left:左
■done:～した　※doの過去分詞形

正三角形を書く　その２

「Run Module」またはF5キーを押してプログラムを実行する。





カメが前に100進む。
120度向きを変える。
前に100進む。
120度向きを変える。
前に100進む。

for文を理解する

For文を使えば繰り返し処理を実行することができる。

繰り返し処理とはscratchの「〇回繰り返す」ブロックと同じ

5回繰り返す処理を書くコード（変数を使わない場合）

```
for 変数 in range(5):
    forward(100)
    left(90)
```

・・・・関数を使うときは最後に「：」をつける

・・・・繰り返しの中で行う処理

※forとinの間には変数が入る。for文で使用する変数のことをカウンタ変数という。カウンタ変数を使用しない場合でもforとinの間に変数を書く必要がある。

5回繰り返す処理を書くコード（変数を使う場合）

```
for 変数 in range(5):
    print(変数)
```

```
0
1
2
3
4
>>>
```

正方形を書く　その１

次に正方形を書いてみよう。

正方形を描くプログラム

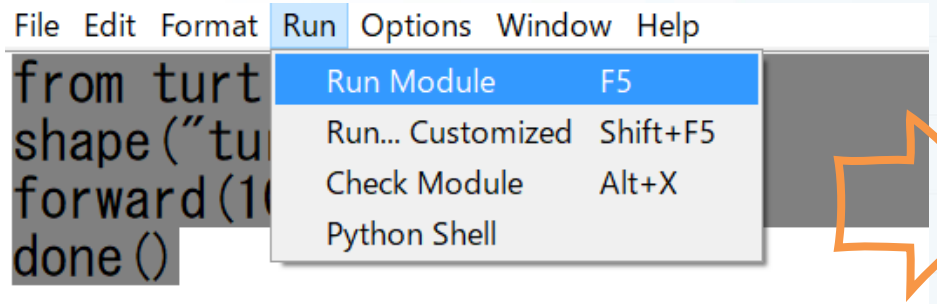
```
from turtle import *        ・・・・タートルグラフィックスの呼出し
shape("turtle")             ・・・・カメを選択
forward(100)                ・・・・まっすぐ100進む
left(90)                    ・・・・90度曲がる
forward(100)                ・・・・まっすぐ100進む
left(90)                    ・・・・90度曲がる
forward(100)                ・・・・まっすぐ100進む
left(90)                    ・・・・90度曲がる
forward(100)                ・・・・まっすぐ100進む
left(90)                    ・・・・90度曲がる
done()                      ・・・・終了
```

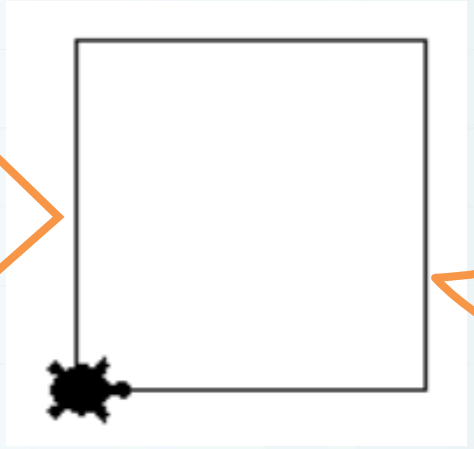
正方形を書く　その１

単語帳　～英語も一緒に覚えてしまおう！～

■from:～から
■turtle:カメ
■import:取り込む。輸入する。
■shape:形　姿　輪郭
■forward:前
■left:左
■done:～した　※doの過去分詞形

「Run Module」またはF5キーを押してプログラムを実行する。





カメが前に100進む。
90度向きを変える。
前に100進む。
90度向きを変える。
前に100進む。
90度向きを変える。
前に100進む。

正方形を書く　その２

もっとシンプルなプログラムで正方形を書いてみよう。

正方形を描くプログラム

```
from turtle import *          ・・・・タートルグラフィックスの呼出し
shape("turtle")               ・・・・カメを選択
for i in range(4):            ・・・・以下の2行を4回くり返す
    forward(100)              ・・・・まっすぐ100進む
    left(90)                  ・・・・90度曲がる
done()                        ・・・・終了
```

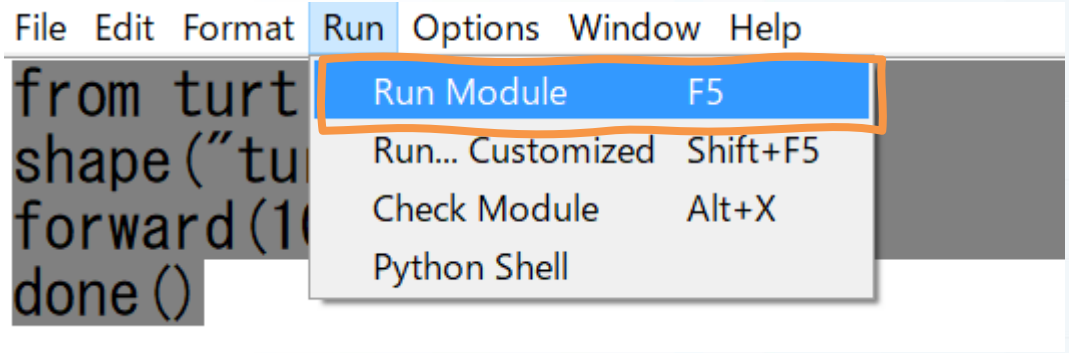
4回くり返す

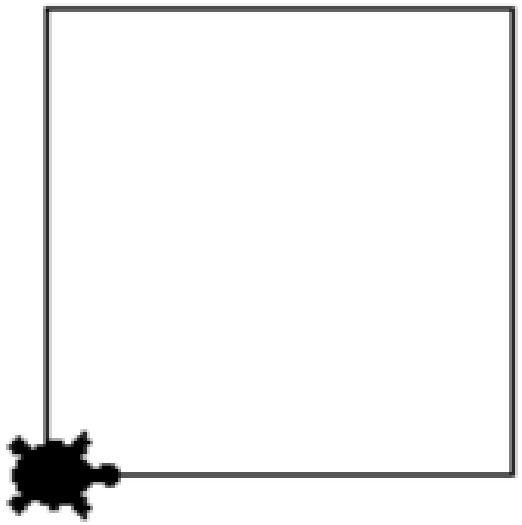
単語帳　～英語も一緒に覚えてしまおう！～

■from:～から　■turtle:カメ　■import:取り込む。輸入する。
■shape:形　姿　輪郭　■for:～のために　■in:～の中に
■range:範囲　■forward:前　■left:左
■done:～した　※doの過去分詞形

正方形を書く　その２

「Run Module」またはF5キーを押してプログラムを実行する。





カメが前に100進む。
90度向きを変える。
前に100進む。
90度向きを変える。
前に100進む。
90度向きを変える。
前に100進む。

チャレンジ(正五角形、正六角形を書く)

正五角形、正六角形にもチャレンジしよう。

正五角形を描くプログラム

```
from turtle import *          ・・・・タートルグラフィックスの呼出し
shape("turtle")               ・・・・カメを選択
for i in range(5):            ・・・・以下の2行を5回くり返す
    forward(100)              ・・・・まっすぐ100進む
    left(72)                  ・・・・72度曲がる
done()                        ・・・・終了
```

5回くり返す

正六角形を描くプログラム

```
from turtle import *          ・・・・タートルグラフィックスの呼出し
shape("turtle")               ・・・・カメを選択
for i in range(6):            ・・・・以下の2行を6回くり返す
    forward(100)              ・・・・まっすぐ100進む
    left(60)                  ・・・・60度曲がる
done()                        ・・・・終了
```

6回くり返す

発展学習（正多角形を書く）

正多角形を描く３つのステップを理解しよう。

①内角の和を求める。

〇角形の内角の和の公式
（〇-2）× 180 ＝ 内角の和
三角形の場合
（3-2）× 180 ＝ 180度
四角形の場合
（4-2）× 180 ＝ 360度

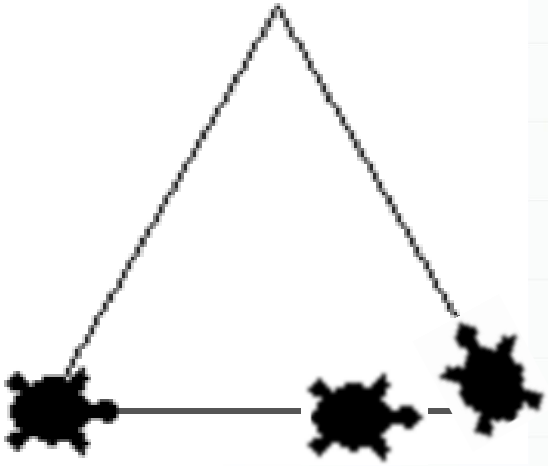
②1つの頂点の角度を求める。

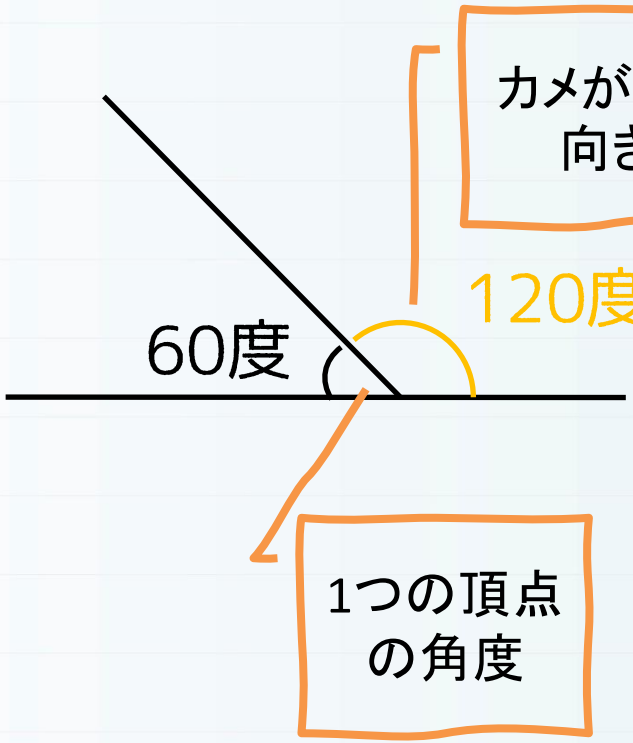
内角の和 ÷ 頂点の数 ＝ １つの頂点の角度
三角形の場合
180 ÷ 3 ＝ 60度
四角形の場合
360 ÷ 4 ＝ 90度

発展学習（正多角形を書く）

③カメの動く向きを求める。

カメの動く向きは
180度 ー １つの頂点の角度 ＝ カメが動く向き






```
for i in range(3):
    forward(100)
    left(120)
```

発展学習（正多角形を書く）

正多角形を描くプログラム

```
from turtle import *        ・・・・タートルグラフィックスの呼出し
shape("turtle")             ・・・・カメを選択
for i in range(〇):          ・・・・多角形の頂点の数
    forward(100)            ・・・・まっすぐ100進む
    left(③)                 ・・・・③で求めた角度
done()          〇回くり返す  ・・・・終了
```

正八角形の場合

頂点の数：8個　・・・・for i in range(8)

カメの向き：180 － 135 ＝ 45度 ・・・・left(45)

内角の和

(8-2) × 180 ＝ 1080度

頂点の角度

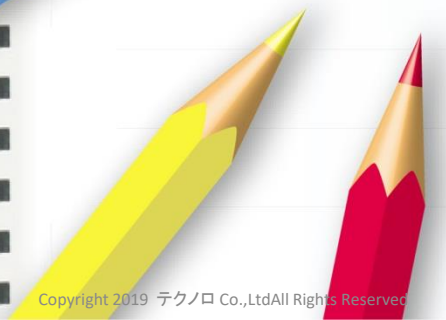
1080 ÷ 8 ＝ 135度

復習＆チャレンジ

ここまで習ったことをScratchでもできるかチャレンジしてみよう。
その過程でScratchでできること、Pythonでないとできないことを整理してみよう。
例えば、データの型という概念はscratchにはあっただろうか？

メモ

プログラミング教室のテクノロ



なまえ：