

Lesson 4

学習日： _____

ピクさんも考える

ピクさんに考えてもらおう。

これまで、あなたがピクさんに命令をしてきました。しかし、ピクさんも考えることができます。今回はピクさんに考えさせるための方法を知りましょう。まずはそのための準備！ピクさんの位置を知りましょう。

1 ピクさんの位置

□ ピクトグラミングの座標

中心が $(0, 0)$ でこれはピクさんの初期位置です。

右上が $(320, -320)$ 、

左上が $(-320, -320)$ 、

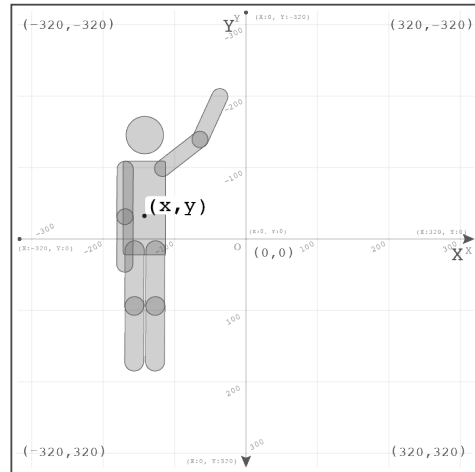
左下が $(-320, 320)$ 、

右下が $(320, 320)$ 、

つまり

x 軸正方向が右、

y 軸正方向が下となります。



ピクさん表示パネルの座標

ピクさんは M 命令または MW 命令を用いて体全体を平行移動することができます。

ピクさんの画面上に、座標系を表示させることができます。

右上にある「座標系の表示」アイコンの右のチェックボックスをクリックすると有効になります。



座標系の表示

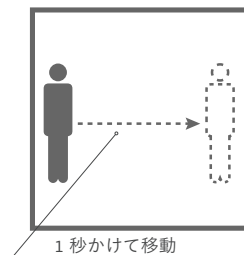
ピクさん表示パネルに座標系を表示します。

命令の様式	処理
M 引数1 引数2 引数3 引数4	引数4秒後に引数3秒かけてx軸正方向に引数1ピクセル、y軸正方向に引数2ピクセルだけ全体を等速直線移動する。引数4が省略された時は、引数4に0が、引数3、引数4の両方が省略された時はいずれも0が入力されているものとして取り扱う。
MW 引数1 引数2 引数3	引数3秒かけてx軸正方向に引数1ピクセル、y軸正方向に引数2ピクセルだけ全体を等速直線移動する。直線移動が終了するまで次の命令は実行されない。

例 ピクさんの位置を動かす

M 命令の使用例です。

```
M -240 0 // 最初の位置
M 480 0 1 // 右に移動する
```



実行結果のイメージ

ピクさんが1秒かけて移動します。

2 ピクさんに考えさせる方法

ピクさんは考える事ができます。ここではピクさんに考えさせる方法があります。その書き方を学びます。まずは、ランダムな数字を考えてもらう方法です。

☑ ランダムな数を思い浮かばせる

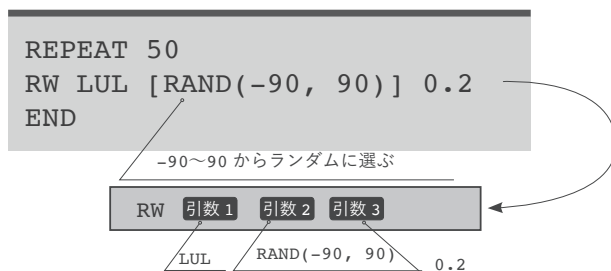
ピクさんに命令するときに、今までは引数に数値を入力していました。その数値をピクさんに考えてもらいます。それには関数を使います。関数の説明は、少し後にしますので、まずは具体的な例を見てみましょう。

関数の表記	機能	戻り値
RAND(<code>min</code> , <code>max</code>)	整数 <code>min</code> 以上 整数 <code>max</code> 以下のランダムな値を返す	整数 <code>min</code> 以上 整数 <code>max</code> 以下のランダムな値。

ここではランダムな値を取得できる関数を使ってみます。

関数を使うときはその両側を「`[]`」で閉じます。

例 脚を回すピクさん（関数の使い方）



「関数」は、与えられた文字や数字の組（引数の組）に対して定められた処理を行って結果（戻り値）を返す機能のことを言います。例えば「-90(度)以上90(度)以下の角度の1つ決めてください」という処理に対して「37」を結果として返すということです。対して、これまで学んできた「命令」は戻り値はありません。これが命令と関数の違いです。ピクトプログラミングではいくつかの関数が定義されています。書き方も命令と少し異なります。ここではランダムな値を取得できる関数を使ってみます。

☑ 引数に関数を使ってピクさんを動かす

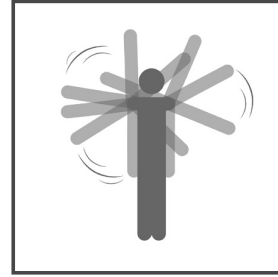
ピクさんに乱数を使って命令します。

例 ぶるぶる震えるピクさん（関数の使い方：引数に関数）

```
SC 0.3
REPEAT 1000
MW [RAND(-10,10)] [RAND(-10,10)] 0.02
END
```

例 ピクさんのロボットダンス（関数の使い方：引数に関数）

```
REPEAT 50
R  LUA [RAND(-90, 90)] 0.2 // 左腕を動かす
RW RUA [RAND(-90, 90)] 0.2 // 右腕を動かす
END
```

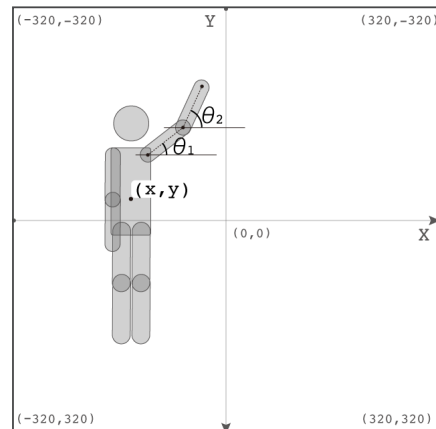


実行結果のイメージ
ピクさんの両腕が
あらゆる向きを指します。

☑ 使用できる関数一覧

関数の表記	機能	戻り値
RAND(min , max)	整数 min 以上 整数 max 以下のランダムな値を返す。	整数 min 以上 整数 max 以下のランダムな値。
ANGLE(parts)	体の部位を示す文字列 の向きを右向き (x 軸正方向) を 0 度として反時計回りの角度を返す。また値は、0 から 359 の整数値をとる。体の部位を示す文字列は二重引用符 (") で囲うこと。例えば左上腕の角度を知りたいときは「ANGLE("LUA")」となる。	体の部位を示す文字列 parts の向き (0 から 359 の整数値をとる)。
X ()	ピクさんの x 座標を整数値で返す。	ピクさんの x 座標
Y ()	ピクさんの y 座標を整数値で返す。	ピクさんの y 座標
COS(theta)	theta 度の余弦を返す	theta 度の余弦
SIN(theta)	theta 度の正弦を返す	theta 度の正弦
TAN(theta)	theta 度の正接を返す	theta 度の正接
ABS(val)	val の絶対値を返す	val の絶対値

関数の中には、ANGLE(**parts**) や X()、Y() といったピクさんの状態を知ることができるものもあります。このような関数を使用することで、複雑な動きをピクさんにさせることができます。



ピクさんの θ (theta)

例 ぶらんぶらんダンス (関数を使った例)

```
R LUA -90
R RUA 60
SET :DL 0
SET :DR 0
REPEAT 5000
SET :DL [ :DL - COS ( ANGLE ( "LUA" ) ) ]
SET :DR [ :DR - COS ( ANGLE ( "RUA" ) ) ]
RW LUA :DL 0.02
RW RUA :DR 0.02
END
```



実行結果のイメージ

ピクさんの両腕が振り子のように動きます。

ここまでに紹介した命令の一覧

命令の様式	処理
R 引数1 引数2 引数3 引数4	引数4秒後に引数1で指定される体の部位を反時計回りに引数2度だけ引数3秒かけて支点を中心に等速回転する。引数4が省略された時は0が、引数3、引数4の両方が省略された時はいずれも0が入力されているものとして取り扱う。
RW 引数1 引数2 引数3	引数1で指定される体の部位を反時計回りに引数2度だけ引数3秒かけて支点を中心に等速回転する。回転が終了するまで次の命令は実行されない。
M 引数1 引数2 引数3 引数4	引数4秒後に引数3秒かけてx軸正方向に引数1ピクセル、y軸正方向に引数2ピクセルだけ全体を等速直線移動する。引数4が省略された時は、引数4に0が、引数3、引数4の両方が省略された時はいずれも0が入力されているものとして取り扱う。
MW 引数1 引数2 引数3	引数3秒かけてx軸正方向に引数1ピクセル、y軸正方向に引数2ピクセルだけ全体を等速直線移動する。直線移動が終了するまで次の命令は実行されない。
FR	ピクさんを正面向き(初期状態)にする。 Front の意味。
SD	ピクさんを側面向きにする。 Side の意味。
C	ピクさんの状態を直立状態(初期状態)にする。 Clear の意味。

命令の様式	処理
SET : 引数1 引数2	変数引数1に引数2を代入する。引数1の前には:をつける。
REPEAT 引数1	対応する END までの命令を引数1回繰り返す。
END	繰り返しの終了。
W 引数1	引数1秒間、何もせずに待つ。待ちが終了するまで次の命令は実行されない。

ピクさんからの挑戦状

4 回目のピクさんからの挑戦状です。ピクさんからの挑戦状は、ピクさんからの挑戦状は、あなたがやりたいと思うところからやってください。ピクチャレ 1 から順に取り組む必要はありません。また、全てのピクチャレに取り組む必要もありません。あなたがやりたいピクチャレに取り組んでください。さあ、ピクさんからの挑戦状「ピクチャレ」に挑戦してみましょう。

ピクチャレ 1 ピクさんにパタパタさせよう

次のプログラムを作ってみましょう。

- (1) 次に示す (2) を 50 回繰り返してください。
- (2) 0.1 秒かけて -90 度から $+90$ 度のいずれかの角度、左腕と右腕を同時に（その同じ角度だけ）動かしてください。ただし (2) でうごかす -90 度から $+90$ 度のいずれかの角度は 50 回の繰り返して毎回ランダムで選ばれるようにしてください。

ピクチャレ 2 ピクさん宇宙遊泳

次のプログラムを作ってみましょう。

- (1) はじめに原点以外のどこかにピクさんを移動させましょう。
- (2) 次に書かれている (3) を 1000 回繰り返します。
- (3) ピクさんが座標 (x, y) にいるときに 0.02 秒かけて x 軸方向に $(-0.05 \times y)$, y 軸方向に $(0.05 \times x)$ だけ移動 (MW 命令を使う) させましょう。

ピクチャレ 3 オリジナルピクさんをつくろう

自由に作品を作ってみてください。
ただし関数を使うこと。

分からないことがあったら、前のページなどを参考にしましょう。

MEMO

本テキストの著作権は青山学院大学 社会情報学部 伊藤一成に帰属します。

© 2017 青山学院大学 社会情報学部 伊藤一成研究室 All rights reserved.