PICTOGRAMMING

（英語命令版　20211024 版）

１章　日常生活におけるピクトグラム

ピクトグラムとは？

ピクトグラムとは日本語で絵記号，図記号と呼ばれるグラフィックシンボルで，意味するものの形状を使ってその意味概念を理解させる記号です. ピクトグラムは案内, 安全, 施設，機器等々，様々な用途で標準化されています.

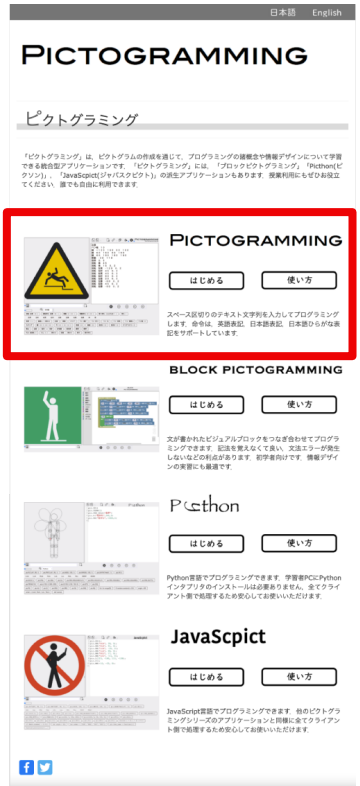
ピクトグラムは，世界共通の記号表現として世界中で用いられていますが，特に近年のグローバル化やその流れに伴う外国人観光客の急激な増加などの理由もあり，ピクトグラムを題材とする研究が盛んになっています.

ピクトグラム作成入門

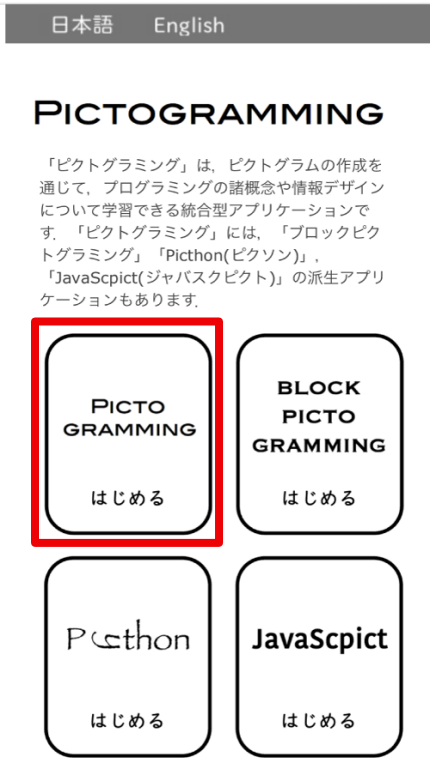
　では実際に，これからピクトグラムを作ってみましょう．
ブラウザでピクトグラミングのホームページへアクセスしてください．アドレスはこちらです．「ピクトグラミング」と検索エンジンで検索してアクセスしても結構です．

`https://pictogramming.org`

ピクトグラミングのトップページが表示されました．PC やノート PC でアクセスする場合は，Pictogramming の「はじめる」ボタンを押してください．スマートフォンで使えるバージョンも用意されています．　スマートフォン版では右下のボタンをタップしてください。



PC 版



スマートフォン版

ボタンを押すと「図　アプリケーションの画面」に示すアプリケーションが現れます．

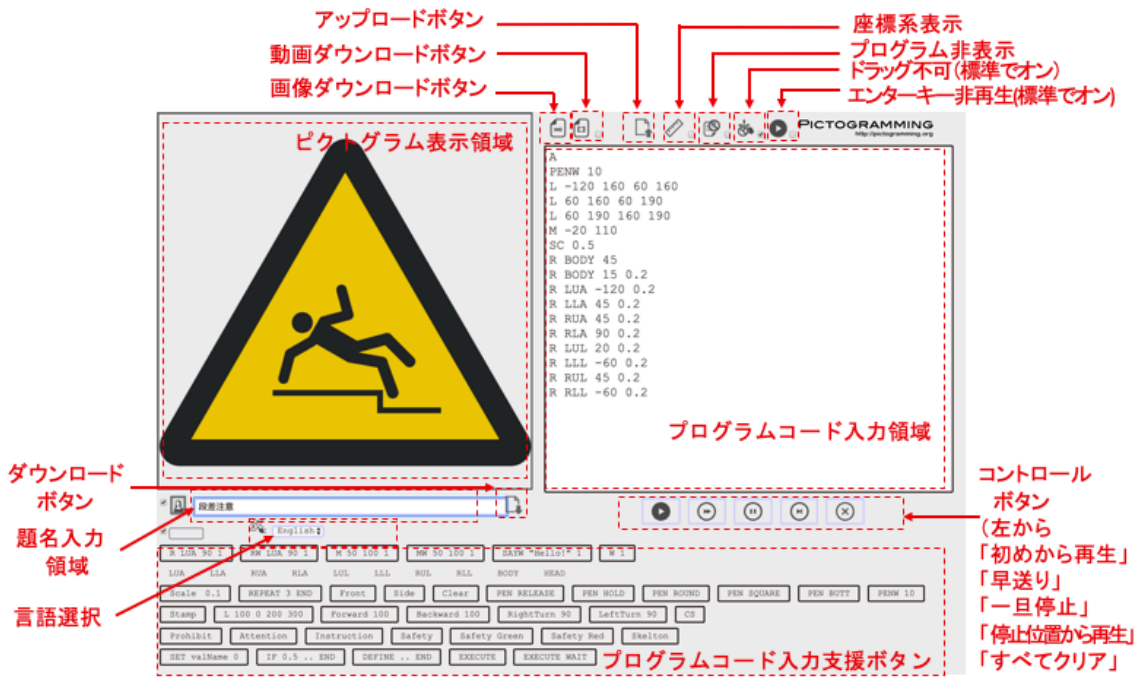


図　Pictogramming の画面(PC 版)

PC 版の画面は主に 3 つの部分から構成されています．スクリーンショットの図において上部左側は，プログラムの実行結果を表示する人型ピクトグラム表示領域，上部右側はプログラムを入力するプログラムコード入力領域，下部にはプログラムの入力を支援するプログラムコード入力支援ボタン群が配置されています．左上部分のパネルに表示されているのが，人型ピクトグラムです．

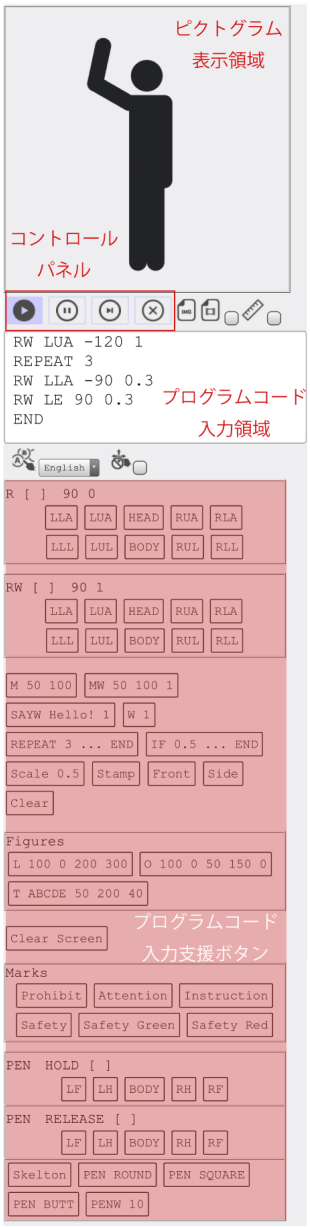


図　Pictogramming の画面(スマートフォン版)

スマートフォン版の画面は,PC 版と同様に主に 3 つの部分から構成されています. スクリーンショットの図において上部には, プログラムの実行結果を表示する人型ピクトグラム表示領域，中央にはプログラムを入力するプログラムコード入力領域，下部にはプログラムの入力を支援するプログラムコード入力支援ボタン群が配置されています. 上部のパネルに表示されているのが，人型ピクトグラムです.

動きを記述する

さて, まずは右上の「プログラムコード入力領域」に以下のように入力してみましょう.

```
R LUA -120
```

続けて, 実行ボタン  を押してみましょう. はい. 左腕が上がりました.

この記述の意味を説明します.
はじめに, R を説明します. R は「回転（Rotate)」の意味です.
次の LUA というのは, 体の部位を示しており, この場合, 左上腕（LUA: Left Upper Arm）の意味です.

さらに 2 行目, 3 行目を追加しました, 下の例では, 左上腕を反時計回りに-120 度, つまり時計回りに 120 度回転させています. さらに右下腿（RLL: Right Lower Leg）を反時計回りに 37 度回転します. それに続けて体全体(BODY)を反時計回りに-52 度, つまり時計回りに 52 度回転させています.

```
R LUA -120
R RLL 37
R BODY -52
```

それぞれの行を**文（statement）**と呼びます. つまり文を複数記述することで, あなたが画面上の人型ピクトグラムに対して命令する訳です. 命令には様式があります. 体の部位を回転する場合の命令の様式を表に示します.

命令の様式	処理
R arg1 arg2	arg1 で指定される体の部位を反時計回りに arg2 度だけ支点を中心に回転する.

初めの R　が回転を意味する命令の種類です．今後 R 命令だけでなく色々な命令が出てきます．次の arg1，arg2 はいずれも引数（argument）といい，外部から与えられる数や文字のことを言います．Pictogramming（ピクトグラミング）では，あなたがあなたの分身である人型ピクトグラムに命令をしますので，外部とはあなたのことです．普通，回転（R）しろ！と言われただけでは，どこを（左腕なの？右足なの？）とか何度回転するのとか聞き返すと思います．つまり曖昧なく命令を実行するために与える数字や文字（ここでは RLL とか 37 とか）のことを引数というわけです．

ここで，1 つ目の引数（argument）arg1 は体の部位を示す文字列を指定します．体と頭を組み合わせた部位が 1 つと，上腕，前腕，大腿，下腿が左右それぞれ 1 つの計 9 種の部位から構成されます．「図　人型ピクトグラムを構成する部品」に示します．BODY 以外の体の部位は英字 3 文字で表しています．1 文字目は左側（Left）か右側（Right），2 文字目は上側（Upper）か下側（Lower），3 文字目は腕（Arm）か足（Leg）かです．

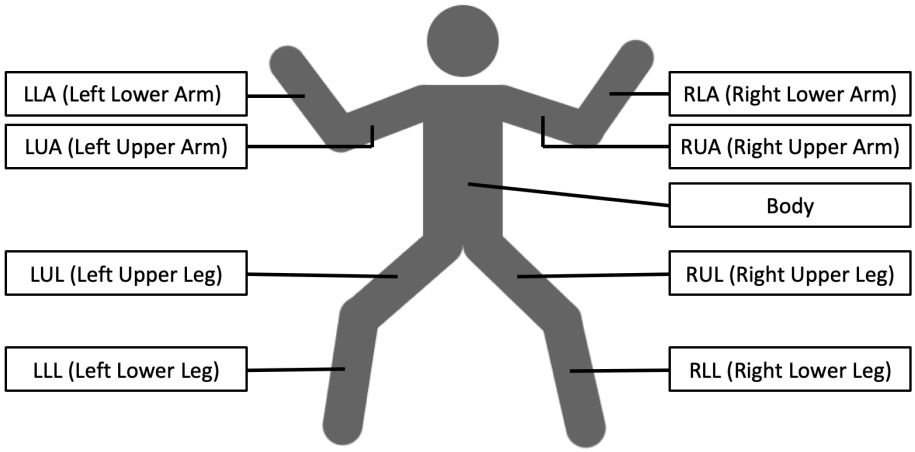


図　人型ピクトグラムを構成する部品

次の 2 つ目の引数 arg2 は回転する角度を示しています．反時計回りが正の値になります．負の値を使って時計回りの角度も表現できます．

ちなみに，命令の名称や体の部位は大文字小文字どちらでも構いません．よって

```
r rll 37
R Body -52
```

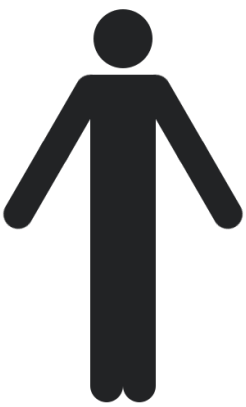
も同じように動きます．

他には次のような命令があります．SD という命令を使うことで，正面からでなく横から見た側面方向のピクトグラムも作ることができます．

命令	処理
FR	人型ピクトグラムを正面向きにする．（初期状態）
SD	人型ピクトグラムを側面向きにする．
C	人型ピクトグラムの状態を直立状態（初期状態）にする．

[演習課題]

(1)から（3）のそれぞれについて，画像と同じ姿勢を作成してください．



(1)

(2)

(3)

2章　逐次（ちくじ）実行，平行（へいこう）実行

```
R LUA -140
```

と入力して実行ボタンを押しましょう．はい．左腕が上がりました．

ではこの命令の-140 のあとに" "（スペース）を挟み，さらに 1 と記述しましょう．以下のようになります．そして，実行ボタンを押しましょう．

```
R LUA -140 1
```

どうなりましたか？徐々に腕が上がりました．これは 1 秒間かけて左上腕を関節点中心に反時計周りに 140 度回転するという意味です．では，さらにこれから，" "（スペース）を挟んで 3 と記述して実行ボタンを押しましょう．

```
R LUA -140  1  3
```

プログラムの実行は実行ボタンが押された瞬間から始まりますが，この場合しばらくは回転せず 3 秒後から 1 秒かけて反時計周りに 140 度回転しました．
前回は，命令 R は 2 つの引数を持つと紹介しましたが，実は，命令 R は最大 4 つまで引数を持つことができます．実際の命令 R の仕様は以下の通りです．

命令の表記	処理
R arg1 arg2 arg3 arg4	arg4 秒後に arg1 で指定される体の部位を反時計回りに arg2 度だけ arg3 秒かけて支点を中心に等速回転する．arg4 が省略された時は，arg4 に 0 が，arg3，arg4 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．

では次に以下の 2 行からなるプログラムを記述して実行してみましょう.

```
R LUA -140 1
R RUA 140 1
```

1 秒間かけて右図のように両腕を同時に上げるアニメーションが再生されます.
では，左腕が上げ終わってから右腕を上げるようにするにはどうすればよいでしょう.
そのための命令として，RW（Rotate Wait: 回転待ち）命令があります.
先ほどのプログラムの命令名 R のところを RW に変えて実行してみてください. つまり以下の通りです.

```
RW LUA -140 1
RW RUA 140 1
```

RW 命令は R 命令と同じですが,「回転が終了するまで次の命令は実行されない.」という意味の命令です.

命令の様式	処理
RW arg1 arg2 arg3	arg1 で指定される体の部位を反時計回りに arg2 度だけ arg3 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.

一方，R 命令は,「次の命令も同時に実行される.」という違いがあります. なので，R 命令では両腕が同時に上がりました. R 命令は,「次の命令も同時に実行される.」ということを注意しなければいけません. 例えば,

```
R LUA -140 1
RW LUA 140 1
```

という命令を実行しても人型ピクトグラムは何も反応しません. これは，左上腕を反時

計回りに 140 度 1 秒間で回転する命令と，左上腕を反時計回りに−140 度 1 秒間で回転する命令が同時に実行されるため，実際には両方が相殺されるからです．では，次に

```
R LUA 40 1
R LUA 40 1
R LUA 40 1
```

という命令を実行するとどうなるでしょう？これは `R LUA 120 1` と同一になります.

一方,

```
RW LUA 40 1
RW LUA 40 1
RW LUA 40 1
```

という命令を実行するとどうなるでしょう？これは `RW LUA 120 3` と同一になります.
このように R 命令と RW 命令をうまく組み合わせることで，人型ピクトグラムに対する様々なアニメーションが可能となります.

次に平行移動してみましょう．座標は XY 座標系（X 方向と Y 方向の値，両方を組み合わせて位置を表現する）です．横方向が X 座標で左端の−320 から右端の 320 まで，縦方向が Y 座標で上端の−320 から下端の 320 までをとります．つまり，中心が（0,0），左上が（−320,−320），右上が（320,−320），左上が（−320,−320），右上が（320,−320），つまり X 軸正方向が右，Y 軸正方向が下となります．数学の座標軸と Y 軸だけ逆方向なので注意して下さい. 画面上部の定規アイコンの右のチェックボックスをオンにすると座標系が表示されますので, 最初のうちはこの座標系を表示しながら命令を実行すると良いでしょう.

（←画面上部の定規アイコンの右のチェックボックスをオンにする）

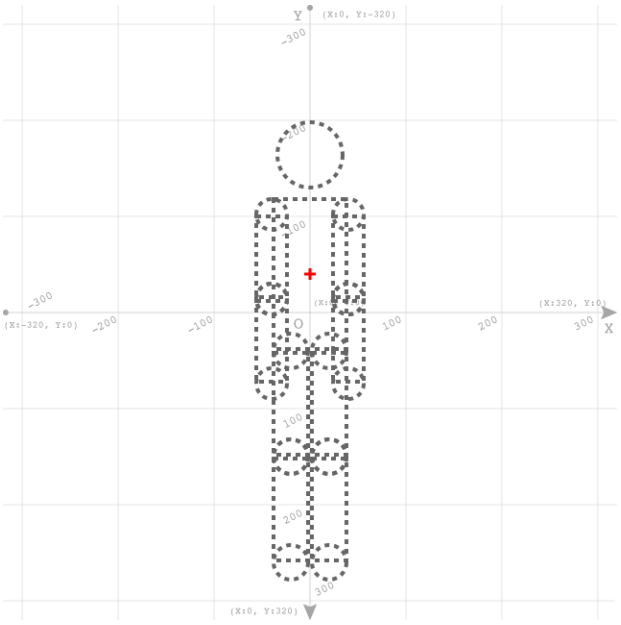


図　座標系を表示した場合の人型ピクトグラム表示領域

ピクトグラムは M 命令または MW 命令を用いて移動できます．命令の様式は下の表の通りです．例えば `M 50 30 1` は，1 秒かけて右に 50 下に 30 平行移動します．
平行移動についても W を記述するか否かで，次の命令も同時に実行するか，並行移動が終了するまで次の命令が実行されないかが決まります．

命令の様式	処理
`M arg1 arg2 arg3 arg4`	arg4 秒後に arg3 秒かけて x 軸正方向に arg1 ピクセル， y 軸正方向に arg2 ピクセルだけ全体を等速直線移動する．arg4 が省略された時は，arg4 に 0 が，arg3，arg4 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 arg1 arg2 arg3`	arg3 秒かけて x 軸正方向に arg1 ピクセル，y 軸正方向に arg2 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．

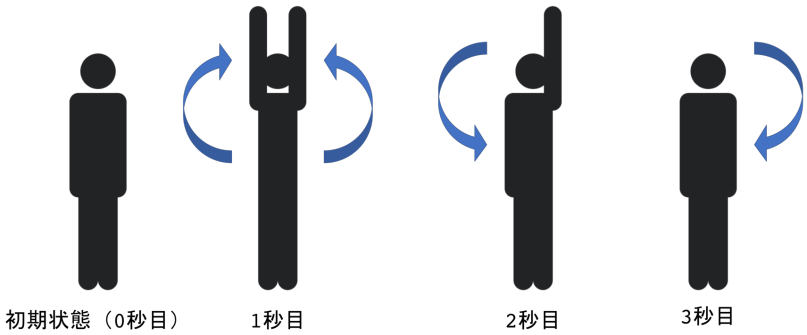
[演習課題]

（1）次のようなアニメーションを作成してください.

0 秒目から 1 秒かけて両腕を上げる.

1 秒目から 1 秒かけて左腕を下ろす.

2 秒目から 1 秒かけて右腕を下ろす.



(2) アニメーションを使った自由作品を作成してみましょう.

3章　似たような処理をまとめる　繰返し処理

```
RW LUA  -140 1
```

と入力して改行キーを押しましょう．はい．左腕が上がりました．
では次に，手を振ってみましょう．友達と別れる時を想像してみてください．

```
RW LUA  -140 1
RW LLA  -60 0.3
RW LLA  60 0.3
RW LLA  -60 0.3
RW LLA  60 0.3
RW LLA  -60 0.3
RW LLA  60 0.3
```

3回左前腕(LLA)を左右に振ることができました．10回20回振るようにもできます．ただ，プログラムがとても長くなってしまいますし，プログラムを見て何回振っているのかをすぐ確認することも難しくなります．そこで繰返しの処理を書く方法を学んでみましょう．使う命令は以下の通りです．

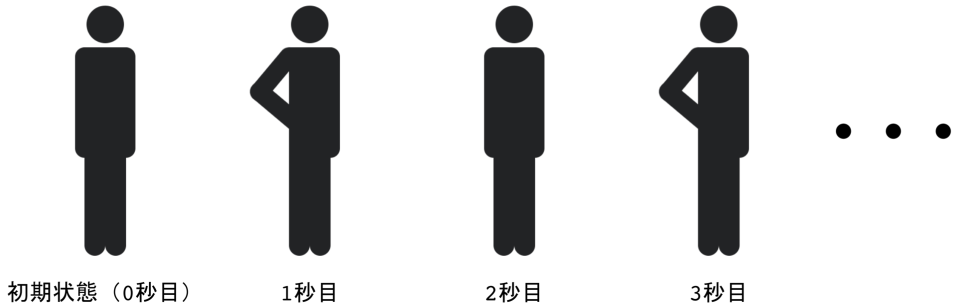
命令の表記	処理
REPEAT arg1	対応する命令群を arg1 回繰り返す．
END	繰り返しの終了．

繰り返しを使うと以下のように書くことができます．

```
RW LUA -140 1
REPEAT 3
RW LLA -60 0.3
RW LLA 60 0.3
END
```

[演習問題]

(1) 次のようなアニメーションを作成して下さい．ただし 繰返し文を使って下さい．
1 秒かけて左腕を曲げて，続けて 1 秒かけて再び左腕を伸ばすということを 10 回繰り返す．



(2) 繰返しを使った作品を自由に作成してみましょう．

4 章　体の部位を動かして絵を描こう

Pictogramming(ピクトグラミング)では，人型ピクトグラムの体の部位を動かすことで絵を描くことができます．まずは，下のプログラムを実行してみましょう．円がかけました．

```
PEN HOLD LH
R LUA 360 1
```

線画は，ペンで描きます．そのペンを持ったり離したりできます．１行目の PEN HOLD LH は，左前腕にペンを持つという命令です．ペン関係の命令の様式は以下の通りです．ペンを持つ，放す体の部位は下の図に示すように，いくつか指定できます．

命令の様式	処理
PEN HOLD arg1	ペンを持つ．ペンを持つ体の部位の名称を R,RW 命令と同様の表記で arg1 に指定できる．arg1 が省略された場合は BODY が記述されているものと見なされる．
PEN RELEASE arg1	ペンを離す．ペンを持つ体の部位の名称を R,RW 命令と同様の表記で arg1 に指定できる．arg1 が省略された場合は BODY が記述されているものとみなされる．
PENW arg1	ペンの太さ（幅）を arg1 にする．初期値は 15.

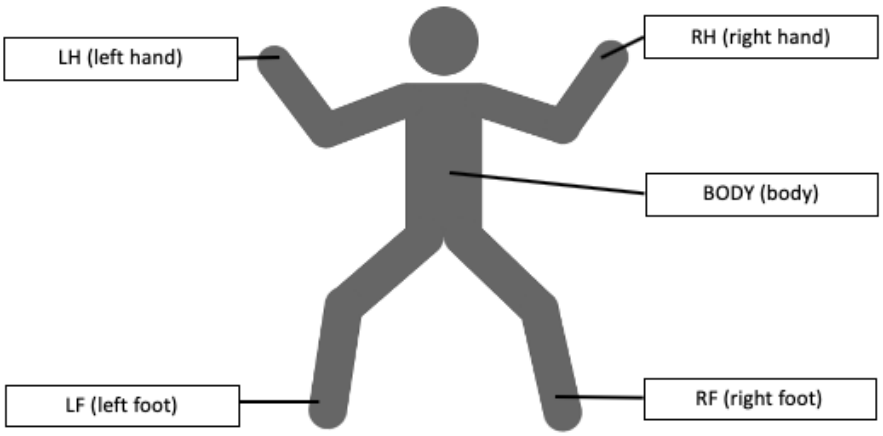
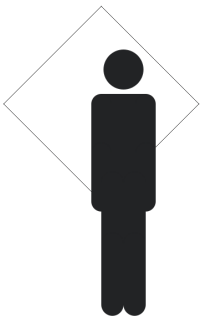


図　ペンを持つことができる位置の名称

では四角形を書いてみましょう.

```
PENW 1
PEN HOLD LH
REPEAT 4
RW LUA 90 0
END
```

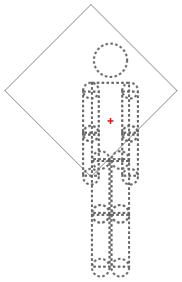


人型ピクトグラムで線画を描くときに, 人型ピクトグラム自身のせいで描いている線画の概形がわかりにくくなることがあるので，人型ピクトグラムを透明にする SK 命令もあります．透明からまた戻すときは，N 命令を実行してください．また，これまでに描いた図形を消す場合は，CS（Clear Screen）命令を使います.

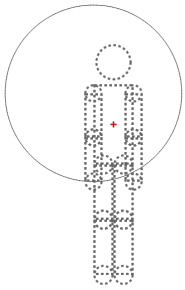
命令の様式	処理
SK	スケルトンモード（Skeleton mode）に変更する。
N	ノーマルモード(Normal mode)に変更する。
CS	ペンによって描画された図形を消去する.

先ほどのプログラムの 1 行目に SK を挿入しました．人型ピクトグラムが透明になるので，描いた線画が見やすくなります.

```
SK
PENW 1
PEN HOLD LH
REPEAT 4
RW LUA 90 0
END
```



5 行目の最後の 0 を 1 にしてみましょう．



```
SK
PENW 1
PEN HOLD LH
REPEAT 4
RW LUA 90 1
END
```

今度は，円になりました．線画は人型ピクトグラムの移動の履歴を描きます．よって普通に回転すれば，その履歴は円になります．
ただし，時間が 0 の時は，瞬間移動です．Pictogramming（ピクトグラミング）では瞬間移動した場合は，移動元と移動先を両端とする線分（直線）を描きます．よって上の例では四角形が書けるのです．

次に 5 行目の RW を R に変更してみましょう．

```
SK
PENW 1
PEN HOLD LH
REPEAT 4
R LUA 90 1
END
```

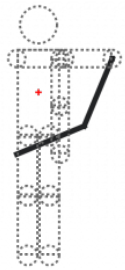
何も線画が描かれなくなりました．これは，次の 2 つのルールがあるからです．

（1）時間 0 の R 命令，M 命令は，元の位置と，R 命令，M 命令からなる複数の命令の動きを全て実行したあとの位置を両端とする線分を描画します．

（2）時間 0 の RW 命令，MW 命令は，元の位置とその命令を実行したあとの位置を両端とする線分を描画します．

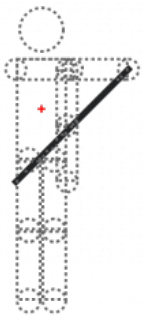
この違いを例に示します.

```
SK
PENW 1
PEN HOLD LH
RW LUA 45
RW LUA 45
```



は右のようになります．左腕の初期位置から，左上腕を反時計回りに 45 度回転した位置まで線分を描きます．さらに左上腕を反時計回りに 45 度回転した位置まで別の線分を描きます.

```
SK
PENW 1
PEN HOLD LH
R LUA 45
RW LUA 45
```

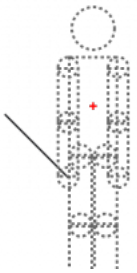


は右のようになります．左腕の初期位置から，左上腕を反時計回りに 90（=45+45）度回転した位置まで線分を描きます.

よって先ほどの例は何も描きません．なぜなら左腕の初期位置と，初期位置から左上腕を反時計回りに 360（=90×4）度回転した位置は同じだからです.

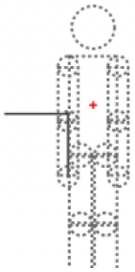
同様に,

```
SK
PENW 1
PEN HOLD LH
M 100 0
MW 0 100
```



は右のようになりますが,

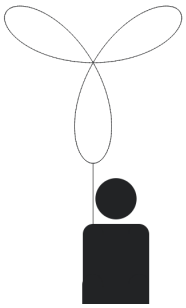
```
SK
PENW 1
PEN HOLD LH
MW 100 0
MW 0 100
```



は右のようになります.

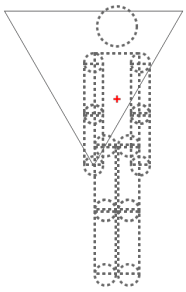
ピクトグラムの体の部位を動かす際と同様，`R, RW, M, MW` 命令をうまく使い分けることで，様々な線画を描くことができます．例えば，下は三つ葉のクローバーです.

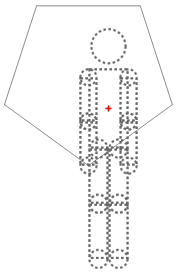
```
PEN HOLD LH
PENW 1
R LUA 360 5
RW LLA -1080 5
MW 0 300
```



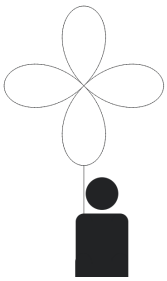
[演習課題]

(1) 正三角形，正五角形をそれぞれ描いてみましょう.

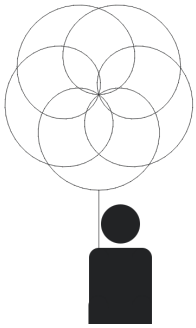




(2) 三つ葉のクローバーのプログラムを変更して，幸せを呼ぶ四つ葉のクローバーを描いてみましょう．

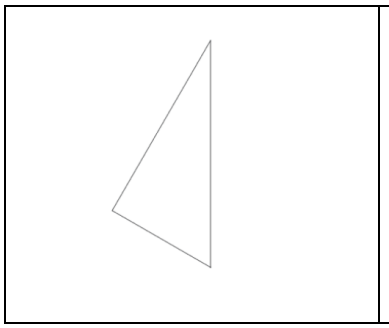


(3) 三つ葉のクローバーのプログラムのいずれかの命令のある引数を一箇所変えるだけで，右のようなお花を描くことができます．やってみましょう．

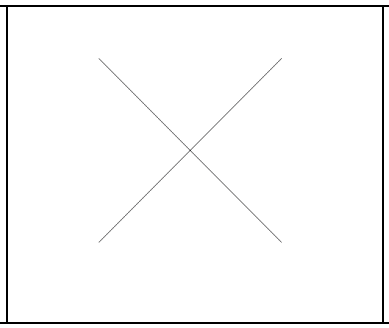


(4) 次の図形を描いてみよう

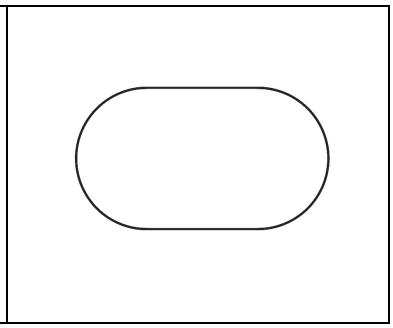
(A) 内角がそれぞれ 90 度，
60 度， 30 度の直角三角形



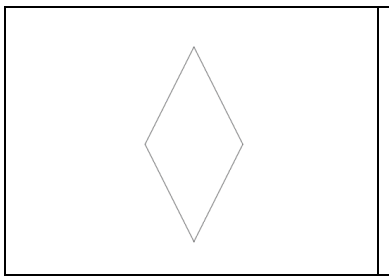
(B) バツマーク



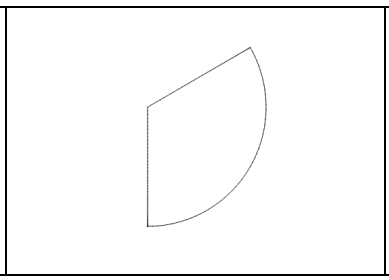
(C) グラウンド



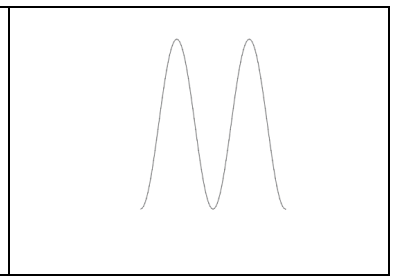
(D) ひし形



(E) 中心角が 120 度の扇型



(F) サインカーブ



5章　泳いで絵を描こう

今回は，人型ピクトグラムの体の部位を動かすのではなく，人型ピクトグラム自体を動かしましょう．みなさんも泳ぐことありますよね．その泳いだ履歴を使って色々な図形を描いてみましょう．

では早速，人型ピクトグラムを泳がせてみましょう．次の 1 行の命令を入力してみましょう．

```
FD 100
```


人型ピクトグラムの大きさが変わり，上からの遠距離視点になりました．　FD 100 は進行方向（頭のある方向）に 100 の距離泳ぐと言う命令です．海やプールの上を泳ぐ人型ピクトグラムを想像してください．

次のように命令を入力してみましょう．

```
FD 100
RT 90
FD 100
```

RT 90　は進行方向を時計回りに 90 度回転すると言う命令です．

```
REPEAT 4
FDW 100 1
RT 90
END
```

アニメーションの正方形が書けました．FDW 100 1 は進行方向（頭のある方向）に 100 の距離 1 秒間かけて泳ぐと言う意味です．最後に W がついていますが，W が付くか付かないかの挙動は，R,RW 命令や M,MW 命令と同様の対応関係になっています．つまりその泳ぎが終了するまで次の命令は実行されません．

泳ぎに関する命令の一覧を表に示します.

命令の様式	処理
FD arg1 [arg2]	人型ピクトグラムを進行方向に **arg2 秒かけて**距離 arg1 だけ**等速で**進める. **arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**
BK arg1 [arg2]	人型ピクトグラムを進行方向と逆向きに**arg2 秒かけて**距離 arg1 だけ**等速で**進める. **arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**
RT arg1 [arg2]	人型ピクトグラムの進行方向を**arg2 秒かけて**時計回り方向に角度 arg1 だけ**等角速度で**回転する. **arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**
LT arg1 [arg2]	人型ピクトグラムの進行方向を**arg2 秒かけて**反時計回り方向に角度 arg1 だけ**等角速度で**回転する. **arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**
FDW arg1 [arg2]	人型ピクトグラムを進行方向に **arg2 秒かけて**距離 arg1 だけ**等速で**進める. **移動が終了するまで次の命令は実行されない. arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**
BKW arg1 [arg2]	人型ピクトグラムを進行方向と逆向きに**arg2 秒かけて**距離 arg1 だけ**等速で**進める. **移動が終了するまで次の命令は実行されない. arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**
RTW arg1 [arg2]	人型ピクトグラムの進行方向を**arg2 秒かけて**時計回り方向に角度 arg1 だけ**等角速度で**回転する. **回転が終了するまで次の命令は実行されない. arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**
LTW arg1 [arg2]	人型ピクトグラムの進行方向を**arg2 秒かけて**反時計回り方向に角度 arg1 だけ**等角速度で**回転する. **回転が終了するまで次の命令は実行されない. arg2 が 省略された時は, arg2 に 0 が入力されているものとして取り扱う.**

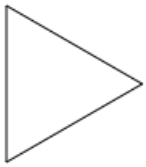
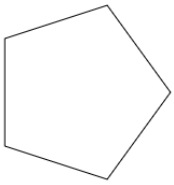
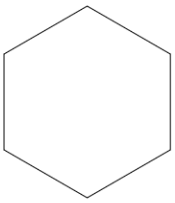
例えば，次の命令を実行すると，「400 の距離を 1 秒間かけて前進する」と「1 秒間かけて 360 度時計回りに回転する」が同時に実行されるので，円を描きます．

```
FD 400 1
RT 360 1
```

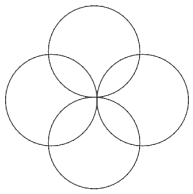
W の有無や命令の組み合わせ方で様々な曲線が描画できます．色々と試してみましょう．

[演習課題]
次の図形を描きなさい（ただし図形が見やすいように人型ピクトグラムは消してあります）

(1) 正三角形　　(2) 正五角形　　(3) 正六角形

（4）4 つの円　　（5）波線




6 章　線と円で図形を書いて動かそう

座標を直接指定して線が描くこともできます．線分の両端の座標を指定して描く L 命令(Line 命令)と，中心座標，幅，高さ，回転角度を指定して楕円を描く O 命令(Oval 命令)いうのが用意されています，仕様は以下の通りです．

命令の様式	処理
L arg1 arg2 arg3 arg4	座標（arg1,arg2）から座標(arg3,arg4)まで描く．
O arg1 arg2 arg3 arg4 arg5	中心座標(arg1,arg2)，幅 arg3，高さ arg4，中心座標を中心に反時計回りに arg5 度回転した楕円を描く．arg5 が省略された時は，arg5 に 0 が入力されているものとして取り扱う．

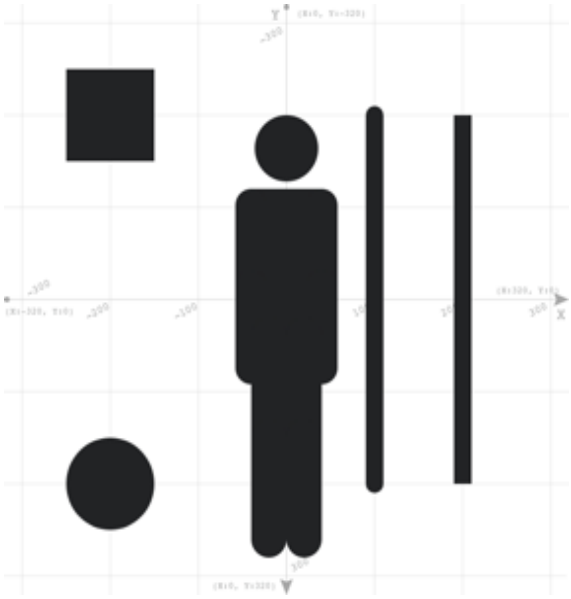
よって，例えば， L 100 -150 200 300 は座標（100, -150）と座標（200, 300）を両端とする線分を（一瞬で）描きます．
実行されます．

線種も変更することができます．

命令の様式	処理
PEN SQUARE	以後描画する線の両端の形状を四角にする．
PEN ROUND	以後描画する線の両端の形状を半円にする．
PEN BUTT	以後描画する線の両端の形状を無しにする．
PEN NORMAL	以後描画する線の線種を実線にする．
PEN ERASE	以後描画する線の線種を消線にする．
PEN XOR	以後描画する線の線種を反転線（すでに描かれていた部分は消し，そうでない部分は描く）にする．

線の太さや線の種類を変更して描画した例を下に示します．

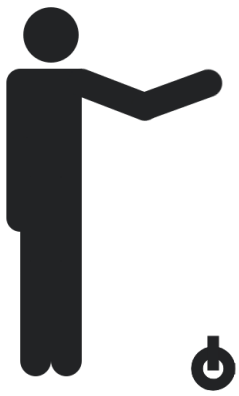
```
PENW 20
PEN BUTT
L 200 -200 200 200
PEN ROUND
L 100 -200 100 200
PENW 100
PEN SQUARE
L -200 -200 -200 -200
PEN ROUND
L -200 200 -200 200
```



一つ以上の基本図形から構成される物体を定義できる DO 命令，定義された物体を等速直線移動する MO,MOW 命令を組み合わせて使うと物体を平行移動できます．「物体を定義するため DO 命令を使って「apple」という名称を設定して定義しています．内部には，円と線分を描く二つの命令があります．これが構成要素となります．平行移動する命令(MO)があり，これにより「apple」が等速で落下します．物体の作成を通じて，複数の命令をまとめることを「手続き化」と言います．

```
R RUA 69
R RLA 41
MO apple 0 320 0.5

DO apple
O 211 -50 42 42 0
L 211 -55 211 -85
END
```



命令の様式は以下の通りです．

命令の様式	処理
DO arg1	対応する END 命令までに含まれる L 命令, O 命令の組み合わせで描かれる線画を arg1 という名称で定義する.
MO arg1 arg2 arg3 arg4	arg4 秒かけて x 軸正方向に arg2 ピクセル, y 軸正方向に arg3 ピクセルだけ arg1 という名称で定義された線画を等速直線移動する.
MOW arg1 arg2 arg3 arg4	arg4 秒かけて x 軸正方向に arg2 ピクセル, y 軸正方向に arg3 ピクセルだけ arg1 という名称で定義された線画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.

[演習課題]

物体を定義して名称をつけましょう. その物体に特徴的な動きを考えて, その動きを再現してみましょう.

7章　分岐を使って異なった処理をする

日常生活において，人は確率や状況によって異なった振るまいをします.
例えば,「サイコロで３以下が出たら，目の前のものを購入しよう」「もし，明日晴れならばピクニックに行こう」「もし，財布の中に 5000 円以上あって，かつ彼女がすでに何か予定が入っていなかったらデートに誘おう」などです.

命令の様式	処理
IF arg1	arg1 の確率で対応する END までの命令を実行する.
END	条件文または繰返しの終了.

例えば，50%の確率で手を振るようなプログラムを記述すると以下のようになります.

```
RW LUA -120 1
IF 0.5
RW LLA -90 0.3
RW LLA -90 0.3
END
```

IF, ELSEIF, ELSE 命令を組み合わせてより複雑な条件を記述することができます.

命令の様式	処理
IF arg1	arg1 の確率で対応する他でもしまたは他または 終わりまでの命令を実行する.
ELSEIF arg2	もし対応する先述の IF または ELSEIF の部分が実行されない場合，arg2 の確率で対応する ELSEIF または ELSE または END の直前までの命令を実行する.
ELSE	もし対応する先述の IF または ELSEIF の部分が実行されない場合，対応する END までの命令を実行する

例えば下のプログラムは，左肘，右肘，右股のいずれかをそれぞれ，40%, 42% ((1-0.4)×0.7)) ，18%((1-0.4)×(1-0.7)) の確率で動かします.

```
R LUA -120 1
RW RUA -120 1
IF 0.4
RW LLA -90 0.3
RW LLA -90 0.3
ELSEIF 0.7
RW RLA -90 0.3
RW RLA -90 0.3
ELSE
RW RUL -90 0.3
RW RUL -90 0.3
END
```

[演習課題]

第 4 回で習った体の部位で図形を描くというのがありました．ランダムに動く人型ピクトグラムの特定の部位にペンを持たせて，ランダムな図形を描いてみましょう．

8 章　正しく伝える難しさを知る -ピクトグラムデザイン-

これまで，自由な発想で様々な作品を作成してきたと思います．今回は，社会や身の回りに役立つピクトグラムというのを作成してみましょう．ピクトグラムは大きく分けてサイン用途とコミュニケーション用途の 2 通りの用途があります．サイン用途は，何か特定の場所に掲示することで，情報を正しく伝えます．コミュニケーション用途は，外国人など人と人との間で言葉などの代わりに情報をやり取りする手段として用います．通常，世の中に広く普及しているピクトグラムは作成ガイドラインに則ってデザインされています．ピクトグラムで大切な事は誰にでも正しく伝わるという事なのです．ピクトグラミングでは通常のモードに加え，禁止，注意，指示，安全（安全，安全緑，安全(赤)）用のピクトグラムを作るための 6 つのモードを備えています．また，コミュニケーション用途向けのピクトグラムを作成するために反転もサポートしています．

命令の様式	処理
P	禁止モード（Prohibit mode）に変更する．
A	注意モード（Attention mode）に変更する．
I	指示モード（Instruction mode）に変更する．指示モード中に描画した線画の色は人型ピクトグラムと同じ白となる．
S	安全モード（Safety mode）に変更する．安全モード中に描画した線画の色は人型ピクトグラムと同じ緑色となる．

SG	安全緑モード(Safety Green mode)に変更する．安全緑モード中に描画した線画の色は人型ピクトグラムと同じ白色となる．
SR	安全赤モード(Safety Red mode)に変更する．安全赤モード中に描画した線画の色は人型ピクトグラムと同じ白色となる．
RV	反転モード(Reverse mode)に変更する。
N	ノーマルモード(Normal mode)に変更する。

さてこれらは，社会の中でよく見かけるピクトグラムの例です．

人型ピクトグラムにポーズをとらせて，伝えたい情報を正しく伝えることができるように，色付きのマークをつけて，さらに線画を付け加えてみましょう．

さて，世の中に溢れるピクトグラムはポスターなど紙媒体上で提示されているので基本は画像（静止画）です．一方最近では，デジタルサイネージ（電子掲示板）の発達により，街中でもアニメーションのピクトグラムを提示させることができるように技術的にはなっています．またスマートフォンなどは位置情報や無線によって，状況に応じて画面上に何らかの表示をすることができるようになっているので，静止画や動画のピクトグラムを表示させて，注意喚起や情報提供をすることができるでしょう．

ピクトグラミングは，静止画だけではなく動画（アニメーション）のピクトグラムを作成することができます．ぜひ，単なる静止画のピクトグラムではなく，オリジナルのアニメーションピクトグラムも作成してみましょう．