



# Lesson 3

学習日：

## ピクさんアニメーション

ピクさんの動画を作ってみよう。

### 1 ピクさんアニメーションを作ろう

#### ☑ アニメーションさせてみよう

今から示す①～③の3つのプログラムを実行してみましょう。  
そして、その3つの違いを観察しましょう。

① `回転 左上腕 -140`

② `回転 左上腕 -140 1`  
間はスペースキーで空白を入力

③ `回転 左上腕 -140 1 2`



#### 得られる結果

左の①～③を実行すると、いずれもこのような姿勢になりますが、何かしらの違いがあります。Lesson3ではその違いがとても重要なポイントです。

①は Lesson2 でやりましたね。左手を上げているピクさんになります。

②は 左手を上げているピクさんになりますが、少し様子が違います。  
徐々に手が上がりました。これはアニメーションです。  
1秒間かけて左上腕を関節点を中心に反時計回りに-140度、つまり時計回りに140度回転するという意味です。

③は プログラムの実行は改行キー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、改行キーを押して2秒後から  
1秒かけて反時計回りに140度回転しました。  
実は回転命令は最大4つまで引数を持つことができます。

命令の様式	処理
<code>回転 引数1 引数2 引数3 引数4</code> 引数1: 体の部位 引数2: 反時計回りに回す角度 引数3: 数値(秒数)かける(省略時は自動的に0) 引数4: 数値(秒数)後に動く(省略時は自動的に0)	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。

☑ タイミングを考えたアニメーション

では次に以下の 3 行からなるプログラムを記述して実行します。

例 アニメーションの例 1

```

回転 左上腕 -140 1 // 左腕をあげる
回転 右上腕 140 1 // 右腕をあげる
回転 右下腿 25 1 // 右足をあげる
    
```



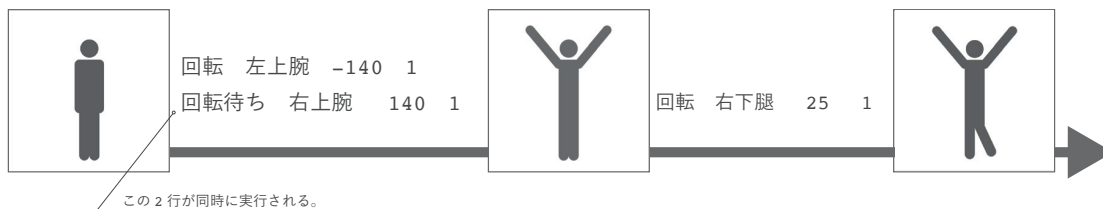
1 秒間かけて両腕と右脚を同時に上げるアニメーションが再生されます。では、両腕を上げてから右脚を上げるようにするにはどうすれば良いでしょうか。そのために回転待ち命令というのがあります。

先ほどのプログラムの命令名回転のところを回転待ちに変えて実行してみてください。つまり以下の通りです。

例 アニメーションの例 2

```

回転 左上腕 -140 1 // 両手を同時にあげる
回転待ち 右上腕 140 1
回転 右下腿 25 1 // 足をあげる
    
```



回転待ち命令の仕様を下に示します。ポイントは「回転が終了するまで次の命令は実行されない。」という点です。

命令の様式	処理
回転待ち <b>引数 1</b> <b>引数 2</b> <b>引数 3</b> 体の部位 反時計回りに回す角度 数値 (秒数) かける (省略時は自動的に 0)	<b>引数 1</b> で指定される体の部位を反時計回りに <b>引数 2</b> 度だけ <b>引数 3</b> 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。

## ☑ 同じ部位のアニメーション命令が同時に実行される場合の注意

回転命令は、「次の命令も同時に実行される。」されます。そのため、回転命令では両腕が同時に上がりました。  
回転命令は、「次の命令も同時に実行される。」ということに注意しなければいけません。

### 例 手をあげる...?

```
回転 左上腕 -140 1  
回転待ち 左上腕 140 1
```

この命令を実行してもピクさんは何も反応しません。  
これは、左上腕を140度1秒間で回転すると  
左上腕を-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。



得られる結果  
時計回りに腕を回転、反時計回りに腕を回転する。同時に行うので打ち消していますね。

### 例 手をあげる（3つの命令を同時に）

40(度)手をあげる命令を3つ実行します。

```
回転 右上腕 40 1  
回転 右上腕 40 1  
回転 右上腕 40 1
```

この命令を実行するとどうなるでしょう？これは 

```
回転 右上腕 120 1
```

 と同一になります。

回転命令と回転待ち命令をうまく組み合わせることでピクさんの様々なアニメーションが可能となります。

## ☑ プログラムを制御するボタン

3つのボタンを改めて紹介します。



プログラム最初から実行  
プログラムを最初のから実行します。



プログラム一旦停止  
プログラムの実行を一旦停止します。  
このボタンを押した瞬間の状態がピクさんは状態で停止します。



プログラム一旦停止箇所からスタート  
一旦停止した状態から再び実行します。

## 2 繰返し

### 例 手をあげる

```
回転 左上腕 - 1 4 0 1
```

と入力して改行キーを押すと左手があがります。

### 例 手を振る

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。

```
回転待ち 左上腕 - 1 4 0 1
回転待ち 左前腕 - 6 0 0.3 // 1往復分 手を振る
回転待ち 左前腕 6 0 0.3
回転待ち 左前腕 - 6 0 0.3 // 1往復分 手を振る
回転待ち 左前腕 6 0 0.3
回転待ち 左前腕 - 6 0 0.3 // 1往復分 手を振る
回転待ち 左前腕 6 0 0.3
```

3回左前腕を左右に振ることができました。では手を振ることを10回や50回にするにはどうしますか。同じ命令をたくさん記述することで実現できますが、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。「〇〇回やって」ということと同じです。使う命令は以下の2つです。

命令の様式	処理
繰り返し <b>引数1</b>	対応する終わりまでの命令を <b>引数1</b> 回繰り返す。
終わり	繰り返しの終了。

```
繰り返し ●
//
終わり
```

### 例 手を振る（繰返しを使って）

上の「手を振る」の例を、繰返しを使って書き直します。

```
回転待ち 左上腕 - 1 4 0 1
繰り返し 3 // 手を振るのを3回繰り返す
回転待ち 左前腕 - 6 0 0.3 // 1往復分 手を振る
回転待ち 左前腕 6 0 0.3
終わり
```

さらに、前回学んだ変数の定義と組み合わせると、手を振る角度や回数を変数で設定できるようになり、より応用性の高いプログラムを書くことができます。

例 手を振る（変数を使って）

```

回転待ち 左上腕 - 1 4 0 1
代入 :角度 6 0 // 曲げる前腕の角度
代入 :数字 3 // 手を振る回数
繰り返し :数字
回転待ち 左前腕 - :角度 0.3
回転待ち 左前腕 :角度 0.3
終わり
    
```

## ここまでに紹介した命令の一覧

命令の様式	処理
回転 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、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。
回転待ち 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。
正面	ピクさんを正面向き（初期状態）にする。
側面	ピクさんを側面向きにする。
クリア	ピクさんの状態を直立状態（初期状態）にする。

命令の様式	処理
代入 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。
繰り返し 引数1	対応する 終わり までの命令を引数1回繰り返す。
終わり	繰り返しの終了。
待ち 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。

---

MEMO

## ピクさんからの挑戦状

3 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。

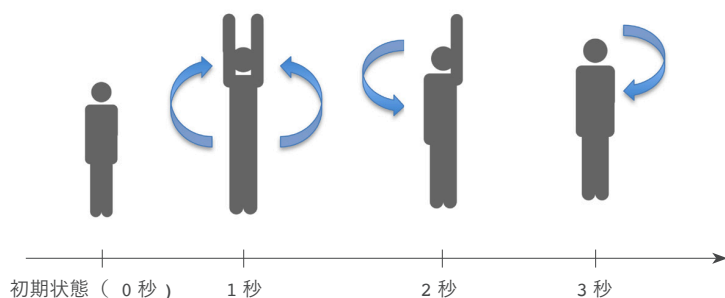
さあ、ピクさんからの挑戦状「ピクチャレ」に挑戦してみましょう。

### ピクチャレ 1) ピクさんに体操をさせよう

次のようなピクさん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。

参考画像



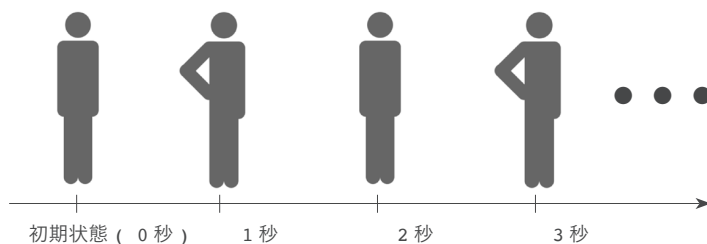
### ピクチャレ 2) ピクさんに体操をさせよう 2

次のようなピクさんアニメーションを作成してください。

ただし 繰り返し 命令を用いること。

- 1 秒かけて左手を曲げて、続けて 1 秒かけて再び左手を伸ばす  
ということを 10 回 繰り返す。

参考画像



### ピクチャレ 3) オリジナルピクさんを作ろう

繰り返しを使った自由作品を作成してみましょう。



---

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.