

# 2次元アニメーション

図形の移動は、平行移動と回転移動の組み合わせで表現できる。ここでは、簡単な図形を元に、それを平行移動、回転移動する方法について学習しよう。

## 平行移動

---

### ■ 楕円を描く

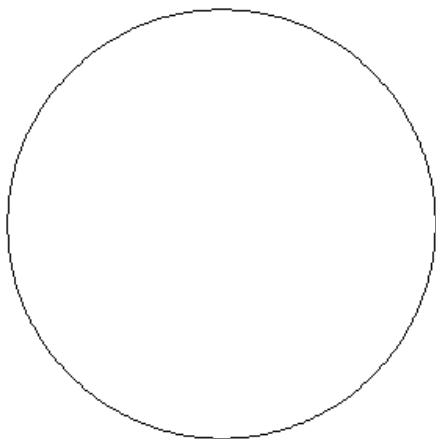
平行移動のために楕円を利用しよう。まず、楕円を描く前に円を描いてみよう。

In[8]=

```
Show[Graphics[Circle[{0, 0}, 2]]]
```

In[9]=

```
Show[Graphics[Circle[{0, 0}, 2], AspectRatio -> Automatic]]
```



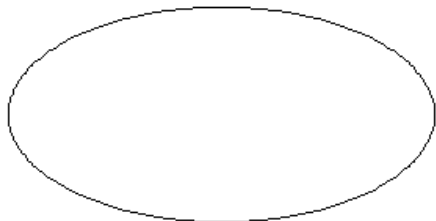
Out[9]=

- Graphics -

次に、楕円を描いてみることにしよう。円と楕円の違いは、楕円には長軸と短軸があることです。楕円を描くには、プリミティブcircleを少し変えて利用する。

In[10]=

```
Show[Graphics[Circle[{0, 0}, {2, 1}], AspectRatio -> Automatic]]
```



Out[10]=

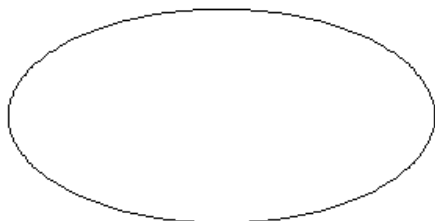
- Graphics -

### ■ 楕円図形を平行移動する

楕円(円)をx軸方向に移動するには、中心のx座標を変化させれば良いよね。中心のx座標を-5%ff5e5まで移動するとしよう。

In[18]=

```
Do[
  Show[Graphics[Circle[{x, 0}, {2, 1}],
    AspectRatio -> Automatic]],
  {x, -5, 5, 1}
]
```

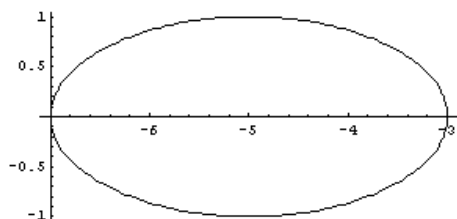


In[7]:=

どうして移動しないのかって?その原因を調べるために座標軸を表示させてみよう.

In[19]:=

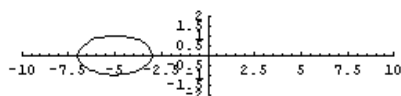
```
Do[
  Show[Graphics[Circle[{x, 0}, {2, 1}],
    AspectRatio -> Automatic, Axes -> Automatic]],
  {x, -5, 5, 1}
]
```



わかったかな?コンピュータは図形が画面の中心になるように平行移動してくれていたのです.これじゃあダメだね.そこでさらにオプションを追加しよう.

In[20]:=

```
Do[
  Show[Graphics[Circle[{x, 0}, {2, 1}],
    AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{-10, 10}, {-2, 2}}]],
  {x, -5, 5, 1}
]
```



軸をなくせばOKだね.

In[21]:=

```
Do[
  Show[Graphics[Circle[{x, 0}, {2, 1}],
    AspectRatio -> Automatic, PlotRange -> {{-10, 10}, {-2, 2}}]],
  {x, -5, 5, 1}
]
```



### 演習問題1

(1)楕円がy軸に平行移動するアニメーションを作成しなさい.

(2)楕円を左下から右上に45度方向に平行移動するアニメーションを作成しなさい。

## ■ 回転移動

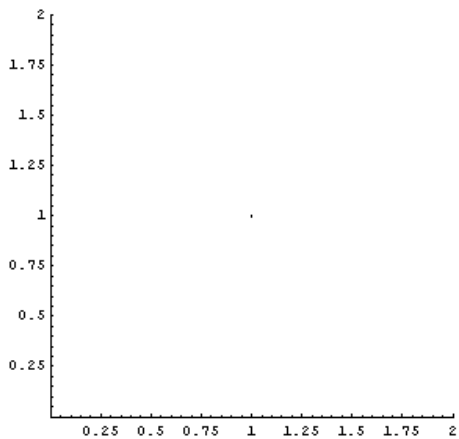
回転移動は、平行移動と比べると難しい。まずは、点で考えてみよう。点の座標が $(x, y)$ であるとき、座標軸を原点周りに角度 $t$ だけ回転した後の点の座標 $(xx, yy)$ は次のようになる。

$$\begin{aligned} xx &= x \cos[t] - y \sin[t] \\ yy &= x \sin[t] + y \cos[t] \end{aligned}$$

具体例を考えることにしよう。最初に、移動前の点 $(1, 1)$ を表示するとつぎのようになる。

In[47]:=

```
Show[
Graphics[Point[{1, 1}],
AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{0, 2}, {0, 2}}]
]
```



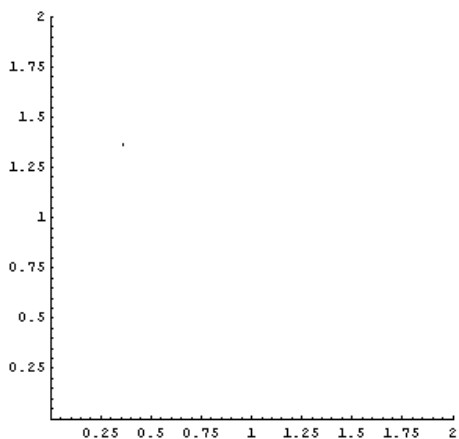
Out[47]=

- Graphics -

次に、点 $(1, 1)$ を30度( $\pi/6$ ラジアン)だけ回転させた後は次のようになる。

In[52]:=

```
t =  $\pi/6$ ;
xx = Cos[t] - Sin[t];
yy = Sin[t] + Cos[t];
Show[
Graphics[Point[{xx, yy}],
AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{0, 2}, {0, 2}}]
]
```



Out[55]=

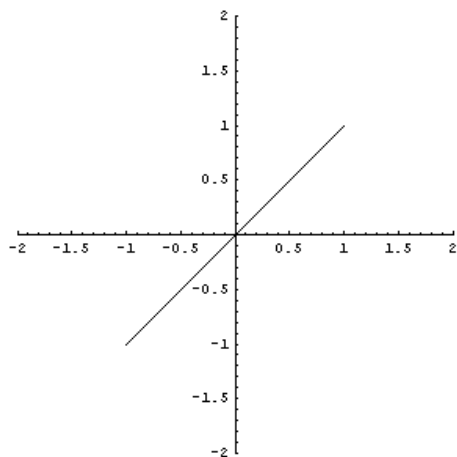
- Graphics -

## ■ 直線の回転移動

最初に直線を描いてみよう。直線は、両端点の座標を与えれば描くことができる。

In[62]:=

```
Show[
  Graphics[Line[{{-1, -1}, {1, 1}}],
  AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{-2, 2}, {-2, 2}}
]
```



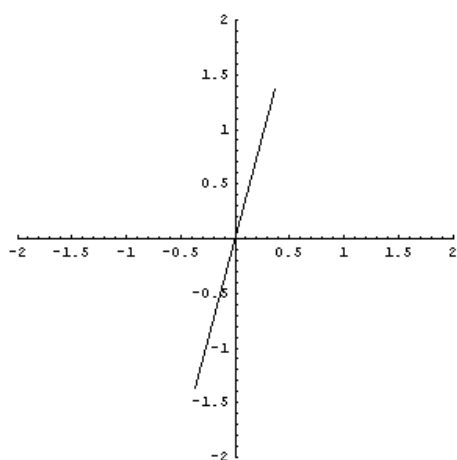
Out[62]=

- Graphics -

直線は対角線上の2点を与えれば描くことができる。ということは、直線を回転させるには、両端点を回転させればよいだろう。具体的に、30度回転させることを考えよう。

In[66]:=

```
t =  $\pi / 6$ ;
x1 = -Cos[t] + Sin[t]; y1 = -Sin[t] - Cos[t];
x2 = Cos[t] - Sin[t]; y2 = Sin[t] + Cos[t];
Show[
  Graphics[Line[{{x1, y1}, {x2, y2}}],
  AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{-2, 2}, {-2, 2}}
]
```



Out[69]=

- Graphics -

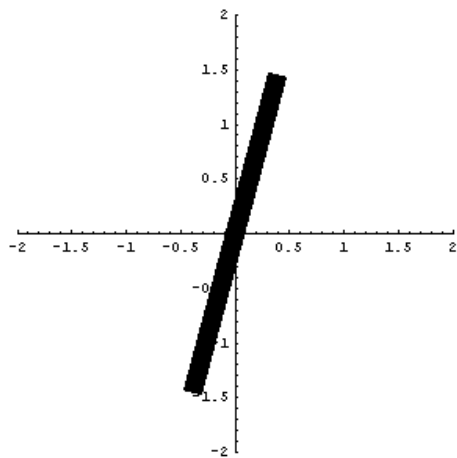
直線を少し太くして棒に見えるようにしてみよう。

In[74]:=

```

t =  $\pi / 6$ ;
x1 = -Cos[t] + Sin[t]; y1 = -Sin[t] - Cos[t];
x2 = Cos[t] - Sin[t]; y2 = Sin[t] + Cos[t];
Show[
  Graphics[{Thickness[0.05], Line[{{x1, y1}, {x2, y2}}]},
  AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{-2, 2}, {-2, 2}}]
]

```



Out[77]=

- Graphics -

ここで使っている Thickness[0.05] はディレクティブである。

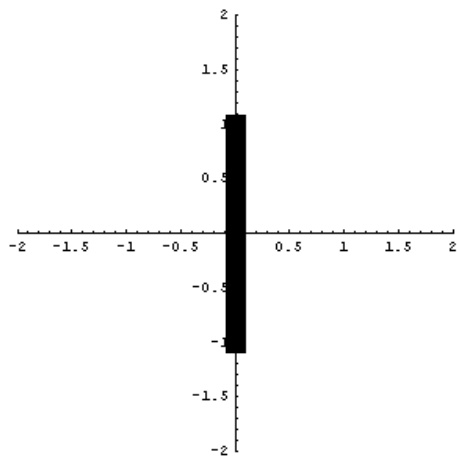
最後に、直立する棒を原点周りに1周させるアニメーションを作成してみよう。まず、直立した棒は次のようになる。

In[78]:=

```

Show[
  Graphics[{Thickness[0.05], Line[{{0, 1}, {0, -1}}]},
  AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{-2, 2}, {-2, 2}}]
]

```



Out[78]=

- Graphics -

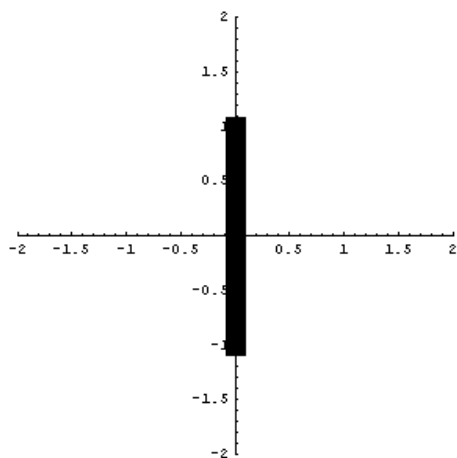
アニメーションとすると次のようになる。

In[80]:=

```

Do[x1 = Sin[t]; y1 = -Cos[t]; x2 = -Sin[t]; y2 = Cos[t];
Show[
Graphics[{Thickness[0.05], Line[{x1, y1}, {x2, y2}]},
AspectRatio -> Automatic, Axes -> Automatic, PlotRange -> {{-2, 2}, {-2, 2}}]
],
{t, 0, 2  $\pi$ ,  $\pi/6$ }
]

```



### 演習問題2

- (1) 三角形を描きなさい。
- (2) 三角形を原点周りに回転するアニメーションを作成しなさい。

### 演習問題3

円がサインカーブ上を移動するアニメーションを作成しなさい。(ヒント: 原点座標は  $(t, \sin[t])$  となる。)

### 努力問題

星形を原点周りに回転するアニメーションを作成しなさい。