

Flash ActionScript 入門

2005年 11月 10日

西図書館マルチメディア工房

[§0. Introduction]

Flashって何？

イラストや写真、ビデオなどを組み合わせてアニメーションなどの作品を作ることのできるソフトウェアです。
ユーザーにボタンをクリックしてもらい、それに応じて動作を変えろといった演出もできます。
今日は、このFlashを使って物理教材を作るための基礎を皆さんに知っていただきたいと思います。

Action Scriptって？

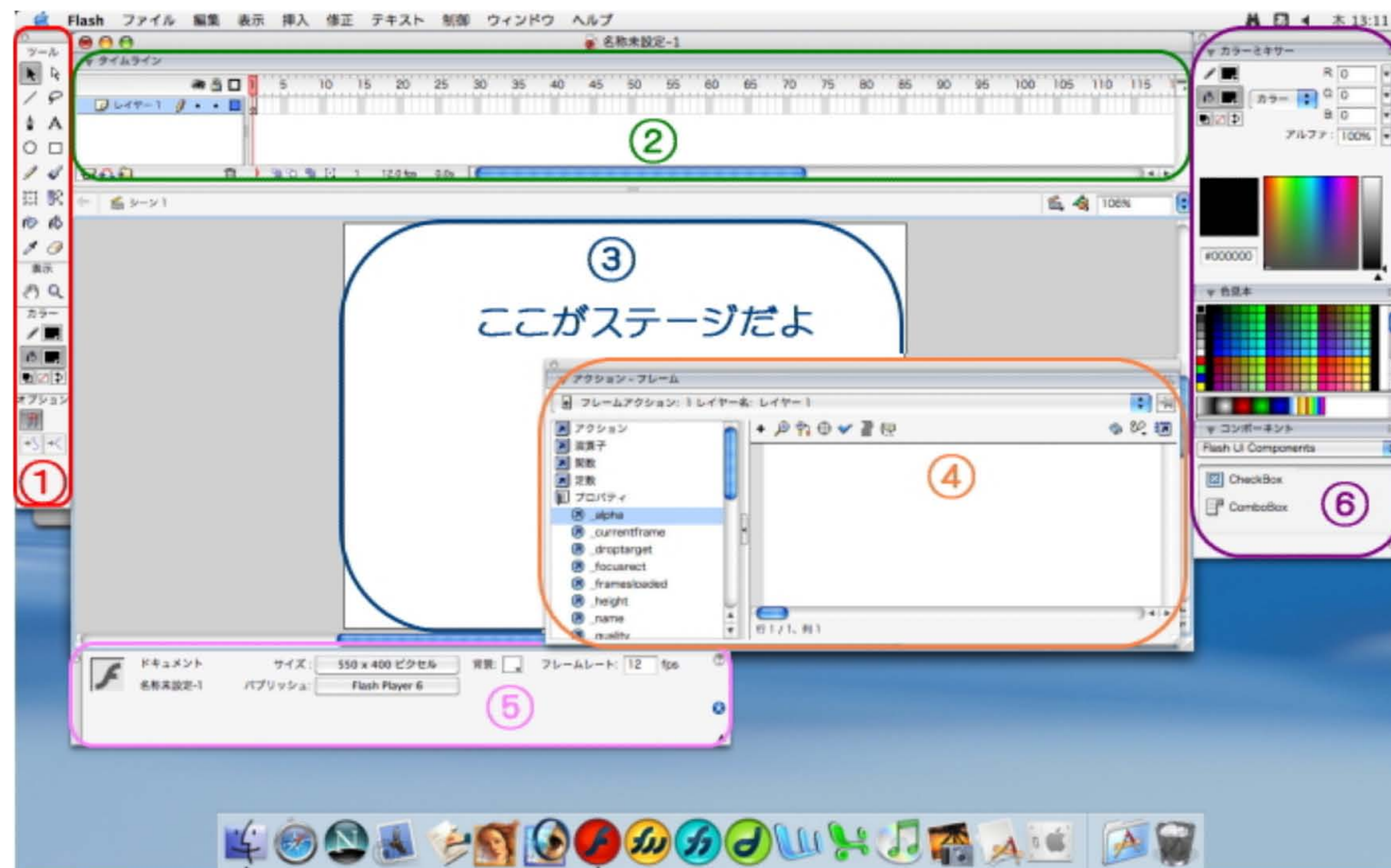
Flashでアニメーションを作ったり、ボタンなどのインターフェイスを作るためのプログラム言語です。

Today's Menu

- ☐ §1. Flashを起動しよう～画面説明～
- ☐ §2. ファイルの種類とムービーの階層構造
- ☐ §3. 作ってみよう～跳ね返るボール～

【§1. Flashを起動しよう～画面説明～】

(起動は、アイコン:  から)



① ツールパレット

絵を描くための色々なツールが入っています

② タイムライン

アニメーションの時間軸やレイヤーを管理します

③ ステージ

編集集中の作品が表示されます

④ アクションパネル

ムービークリップやボタンに変換した画像を動かすための命令をActionScriptで記述します

⑤ プロパティインスペクタ

ステージ上の素材の位置やサイズなどの設定を行えます

⑥ パネル色々

操作が効率良く行えるように、よく使う機能を表示させておきます

※アクションパネルが表示されていない場合は、「ウィンドウ」→「開発パネル」→「アクション」で表示させておいてください。
また、ノーマルモードになっている場合は、 をクリックしてエキスパートモードに設定してください。

【§2. ファイルの種類とムービーの階層構造】

- **Flashのファイル～「.swf」ファイルと「.flaファイル」～**
Flashのファイルには、2種類のファイルがあります。
それぞれの特徴は次のようになります。



「.swf」ファイル

普段、皆さんがホームページなどで見ているファイルです。容量が比較的小さく、ほとんどのブラウザで見る事ができます。編集は出来ません。



「.fla」ファイル

編集出来るファイルです。
動画を動かすスクリプトなどが記述されています。
このファイルから「.swf」ファイルを生成します。

Flashでは、「.fla」ファイルの事を「ドキュメント」と呼ぶ事があります。

□ファイルの階層構造

ステージ... ムービーを再生するための情報を持ちます。

ムービー... ステージ上で再生される動画です。

ムービークリップ... ムービー内で、ムービーとは独立のタイムラインを持って再生されるムービーです。
ActionScriptによって動かしたり、色の変更や移動を命令する事ができます。
ムービーの上にムービーを置くことも出来ます。

ボタン... マウス操作などのイベント発生時にActionScriptを実行することができます。

テキストフィールド... ActionScriptによってテキストの内容を読み込んだり、
書き換えたりする事のできるフィールドです。

グラフィック... トゥイーンアニメーションによって移動・変形などが可能なオブジェクトです。
ActionScriptでは動かさせません。

今日は、ActionScriptで
ムービークリップを動かす
アニメーションを作ります。



[§3. 作ってみよう～跳ね返るボール～]

跳ね返るボールのアニメーションを作る

ActionScriptでのアニメーション作成の基礎として、ここでは壁に衝突し、跳ね返るボールのアニメーションを作ってみます。

手順

その1～ ボールの絵を描き、ムービークリップに変換する

その2～ 作ったムービークリップにActionScriptで命令を記述し、とりあえず動かしてみる

その3～ ステージの端でボールが跳ね返るようにActinScriptを書き足す

完成！動きを見てみよう

作ってみよう ～手順その1～

まずは球の絵を描き、それをActionScriptで動かす事のできるムービークリップに変換します。

青い球を描きます。ツールパレットの楕円ツールボタン  をクリックします。

カラーは  のように、線なし、青グラデーションを選んでみます。

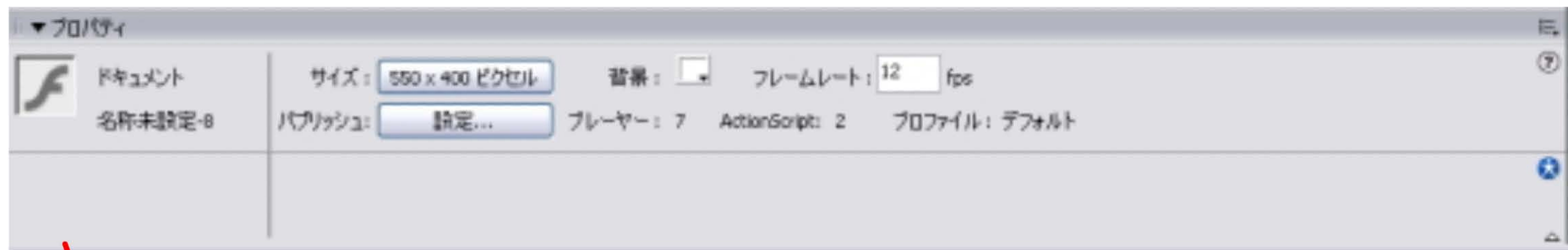
正円を描くには、[Shift]キーを押しながら、マウスをドラッグします。



球の上でCtrl+クリック→[シンボルに変換]をクリックすると、[シンボルに変換]ウィンドウが出てきます。

ここでは、[名前]は「sphere_mc」とし、タイプは「ムービークリップ」を選択します。

[基準点]は左上隅を選択し、最後に[OK]をクリックしましょう。



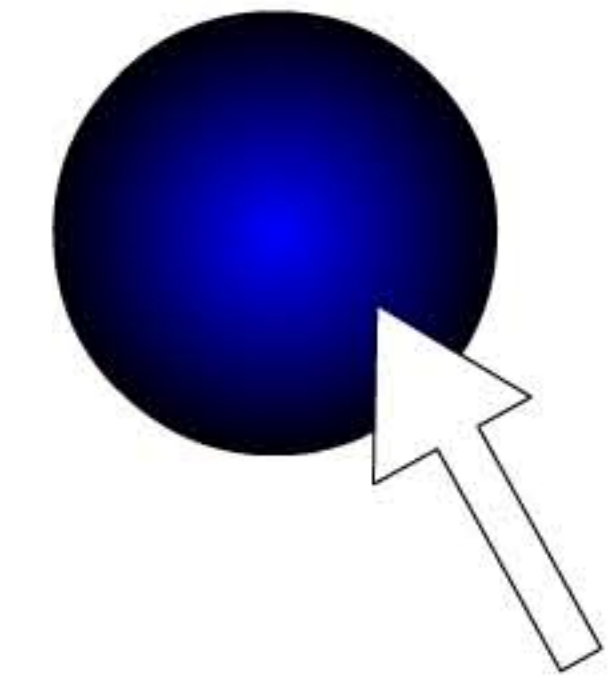
次に、~~ステージ外~~の何もないところをクリックします。
すると、**プロパティインスペクタ**にドキュメントのプロパティが表示されます。
ここでは、ドキュメントのサイズを幅:550、高さ:400ピクセルに設定します。
プロパティインスペクタ内のドキュメントプロパティボタンが**このように**
なっていないならば、このボタンをクリックしてドキュメントのサイズを
550×400ピクセルに設定してください。

サイズ: 550 x 400 ピクセル

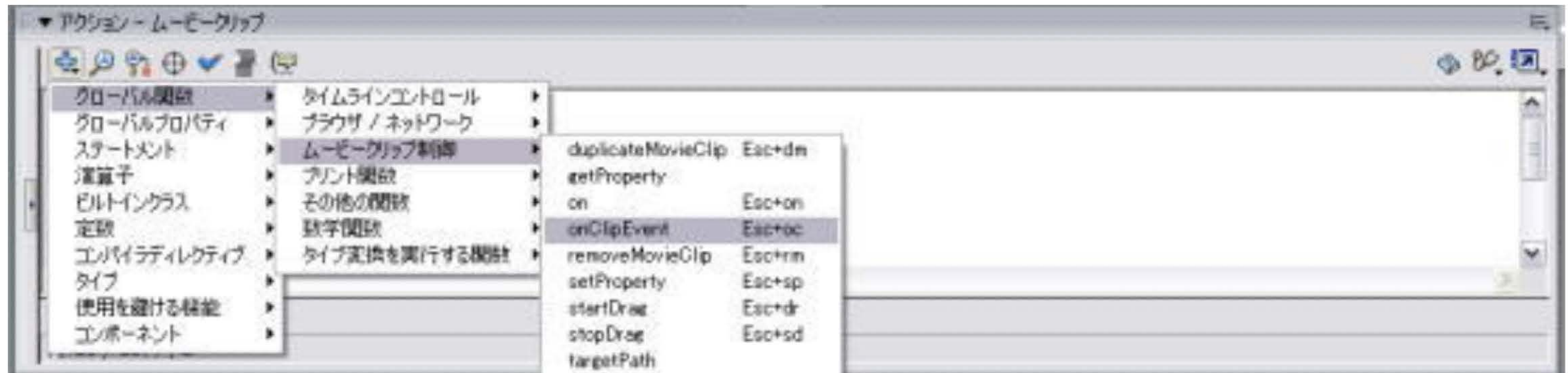
作ってみよう
～手順その2～

ムービークリップに球を動かす命令を書き、とりあえず動かしてみよう

作った球をクリックし、選択状態にします。



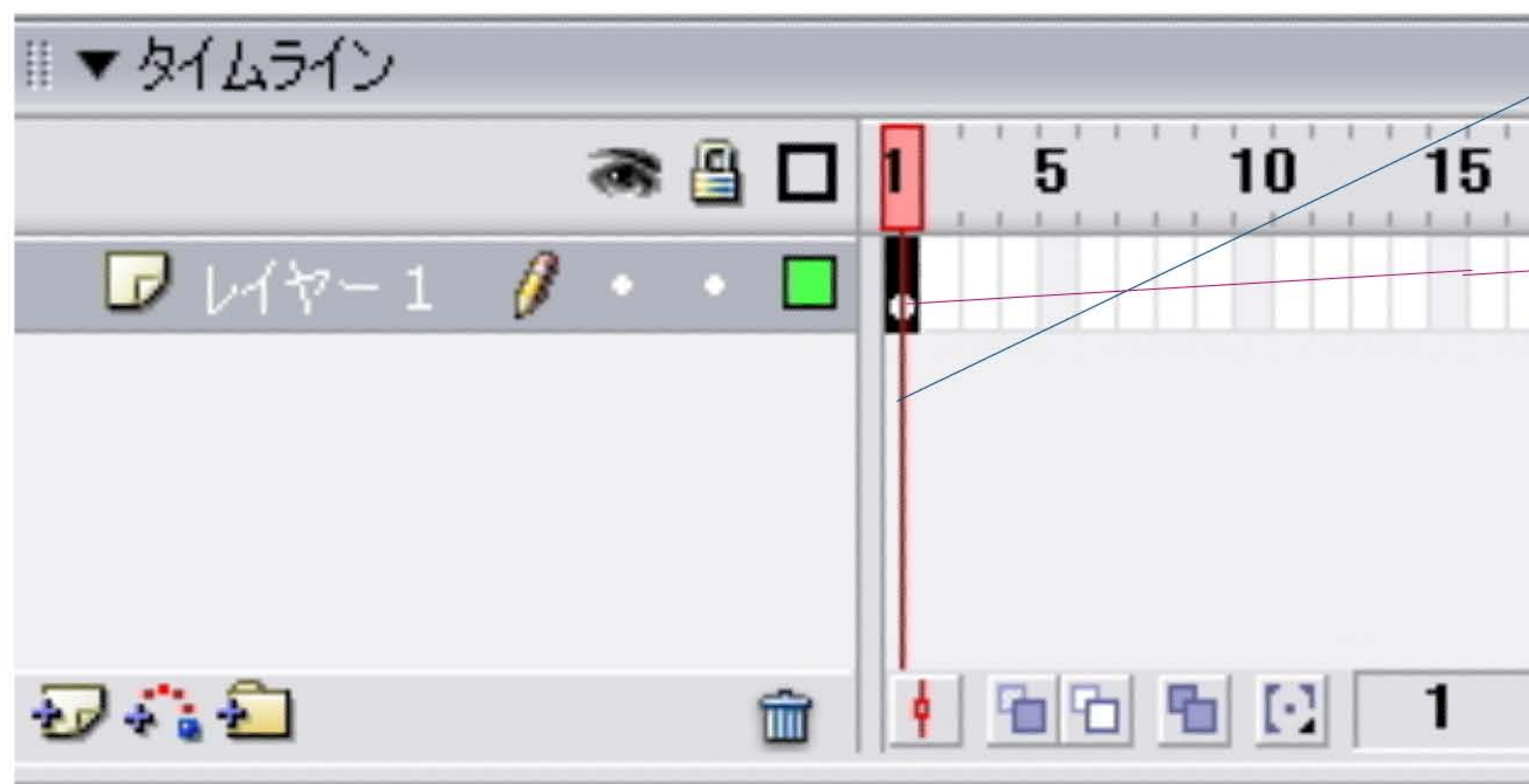
[アクション]パネルの十字キーをクリックし、
[グローバル関数]→[ムービークリップ制御]→[onClipEvent]と選択してください。



プルダウンメニューの中から「enterFrame」をダブルクリックします。
{ }の中に、右のようにスクリプトを記述します。
ムービークリップにスクリプトを記述するときには、
このように「onClipEvent(){...}」の{...}の中に記述します。
描いたら、メニューバーの[制御]→[ムービープレビュー]をクリックして
球が動く様子を見てみましょう。

```
onClipEvent (enterFrame) {  
    _x = _x+5;  
}
```

～解説～



赤い線を「タイムライン」と呼び、これはアニメーション中での時間軸を表しています。小さな短冊形の部分を「フレーム」といいます。Flashのアニメーションは、パラパラ漫画のようにこのフレーム上の図を少しずつ変えながら1つ1つ表示させていく事により、全体として図が動いているように見せているのです。

今作った動く球のアニメーションでは、フレームは1つしかありません。Flashは、フレームを1フレーム目から最後のフレームまで再生し終わると、自動的に1フレーム目に戻ります。

つまり、今の場合は第1フレームが何度も繰り返し再生されているのです。

球が動いているように見えるのは、ActionScriptによって第1フレーム上にあるムービークリップの位置を少しずつ変えながら表示しているからです。

```
onClipEvent (enterFrame) {
    _x = _x+5;
}
```

新しいx座標 古いx座標

先程書いたスクリプトを見てください。
ムービークリップにスクリプトを記述するときには、このように
"onClipEvent(){...}"の{..}の中に記述します。

()の中の"enterFrame"というのは、タイムラインがこのフレーム上に来る度に、このスクリプトを実行しなさい、という意味です。

"_x"は、このムービークリップのx座標、"_y"はy座標を意味します。

"_x=_x+5;"

という式は、「前のムービークリップのx座標に5ピクセルを足した値を、次にこのフレームが再生された時のムービークリップのx座標としなさい」という意味です。

```
onClipEvent (enterFrame) {
    trace(_x);
    _x = _x+5;
}
```

"trace()関数"を使うと、その時点での変数や座標などの値を「出力パネル」に表示してくれます。

左のように、"trace(_x);"というスクリプトを書き足した後、ムービーをプレビューしてみましょう。

(終わったら、"trace(_x)"は消しておいてください。)

```
onClipEvent (enterFrame) {
    _x = _x+5;
    _y = _y+5;
}
```

また、y方向へも動かすために、x軸方向と同様にスクリプトを書き、y方向へも動くかプレビューして確かめましょう。

注意)x座標とy座標は、ステージの左上端で(x,y)=(0,0)となります。
xの正方向は右、yの正方向は下です。

作ってみよう
～手順その3～

ボールをステージの端で跳ね返るようにしましょう

現在、ステージの広さは幅550×高さ400ピクセルに設定されています。
このステージの端で、ボールが跳ね返ってくるようにしましょう。

まず、x方向(水平方向)に向かったボールが跳ね返るように
してみます。右のようにスクリプトを書き直してください。
プレビューすると、x方向に向かったボールが跳ね返ってくる
ようになっています。

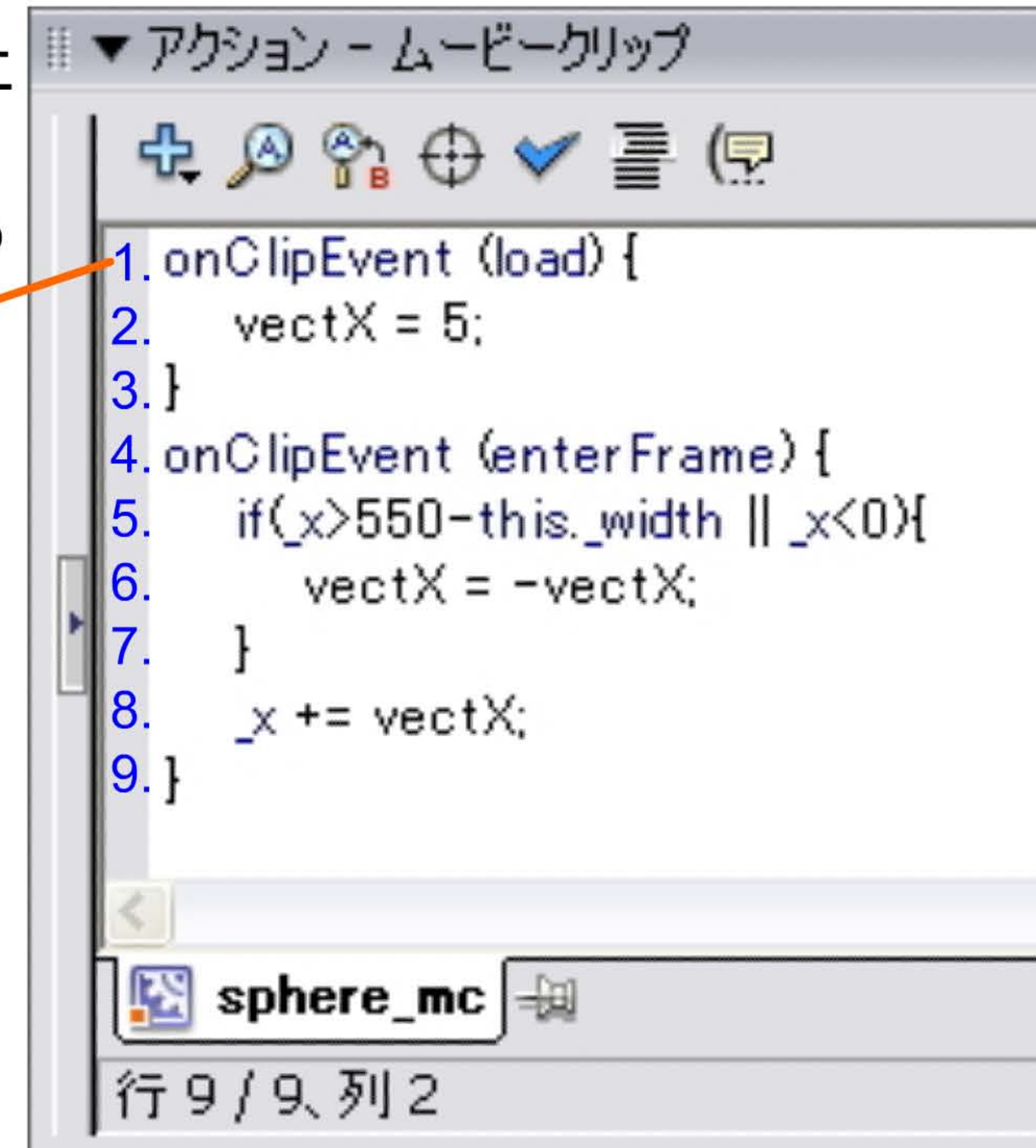
～解説～

1行目の"onClipEvent(load){...}"の{...}中に書いた
スクリプトは、タイムラインがこのフレーム上に初めて来た
時にのみ、1回だけ実行されます。

"onClipEvent(load){...}"中のスクリプト :
タイムラインがフレーム上に来た最初の1回だけ実行される

"onClipEvent(enterFrame){...}"中のスクリプト :
タイムラインがフレーム上に来る度に実行される

この違いに注意してください。



2行目では、変数"vectX"に5を代入しています。

5行目の"this._width"は、このムービークリップ
"sphere_mc"の幅を意味します。

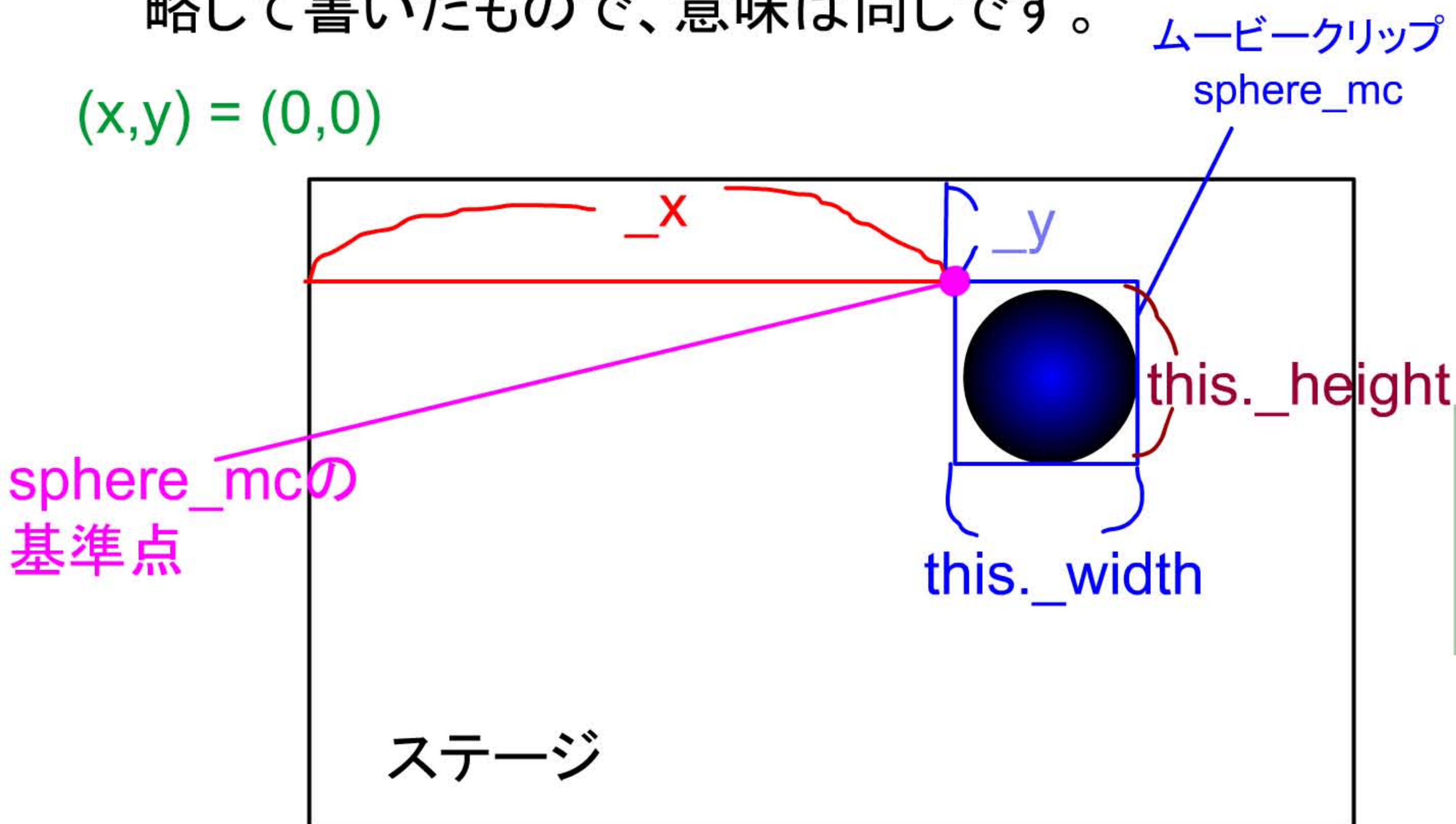
"if(条件){...}"文で、sphere_mcの右端がステージの
右端より右に出た時と、sphere_mcの左端がステージの
左端より左に出た時に、**vectXの符号を逆**にするように
設定しています(下図参照)。

"||"は"OR演算子"で、「又は」という意味です。

8行目の式

"_x += vectX;"は、"_x = _x + vectX;"を
略して書いたもので、意味は同じです。

(x,y) = (0,0)



(x,y)=(550,400)

```
▼ アクション - ムービークリップ
+ 🔍 🔑 ⌕ ✓ ≡ (💬)
1. onClipEvent (load) {
2.     vectX = 5;
3. }
4. onClipEvent (enterFrame) {
5.     if(_x>550-this._width || _x<0){
6.         vectX = -vectX;
7.     }
8.     _x += vectX;
9. }
```

sphere_mc

行 9 / 9、列 2

x軸方向についてのスクリプトが理解できたら、
y軸方向についても同様にスクリプトを
書いてみましょう。

注意)ムービークリップの高さは、
"this._height"で取得できます。