

Flash ActionScript 入門

2006年 2月 2日

大島商船高等専門学校

[\$0. Introduction]

Flashって何?

イラストや写真、ビデオなどを組み合わせる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?

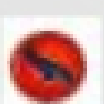
Flash でアニメーションを作ったり、ボタンなどのインターフェイスを作るためのプログラミング言語です。

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～

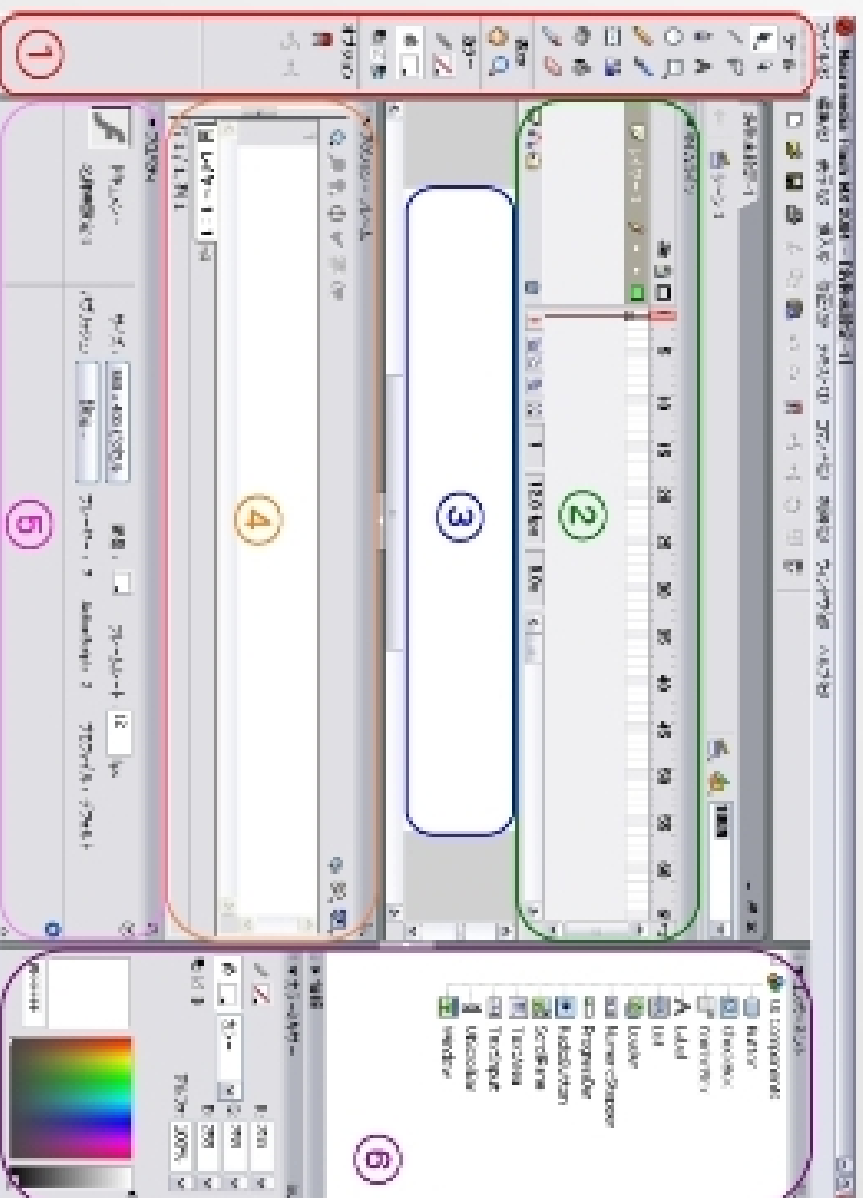
【\$1. Flashを起動しよう～画面説明～】

(起動は、アイコン：



から)

- ・「新規作成」ボタン  をクリックすると、このような画面になります。
- ・アクションパネルが表示されていない場合は、「ウインドウ」→「開発パネル」→「アクション」で表示させておいてください。



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います

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

【§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。



跳ね返る球.swf
Flash ムービー
4 KB

「.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
ブラウザで見える事ができます。編集は出来ません。



跳ね返る球.fla
Flash ドキュメント
22 KB

「.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つ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ができるフィールドです。

グラフィック... トウインシ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ActionScriptを書き足す

完成！動きを見てみよう

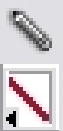
作ってみよう ～手順その1～

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。

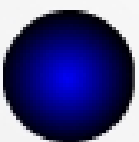
青い球を描きます。ツールパレットの楕円ツールボタン  をクリックします。

カラーは

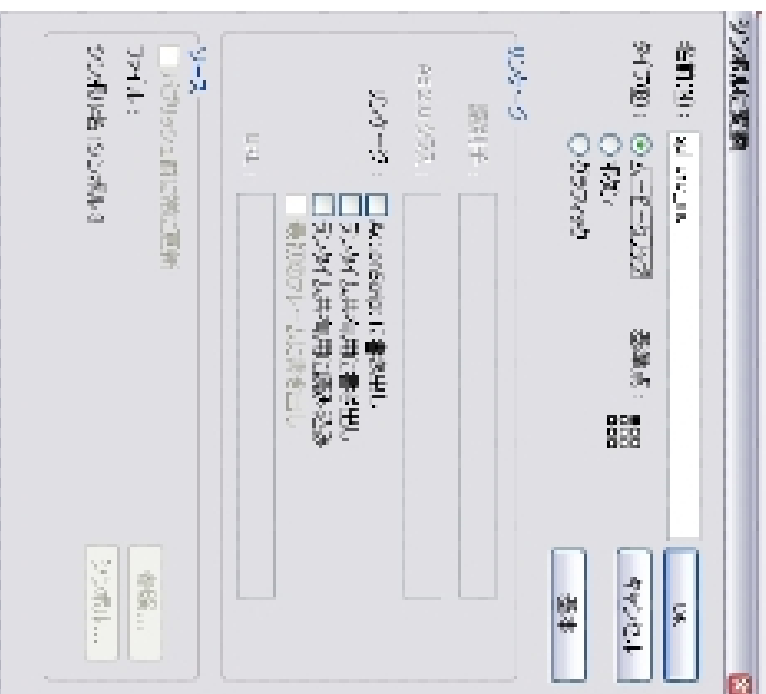
カラー



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]ウイ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。

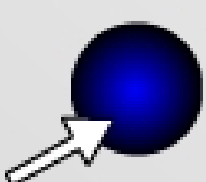
▼ フォント					
	フォント	サイズ: 550 x 400 ピクセル	書体:	フレームワーク: 12 fcs	
名称未設定-8	パブリック	設定...	フレームワーク: 7	ActionScript: 2	フォントファイル: デフォルト

次に、ステージ外の何もない灰色の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
なっている場合は、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。

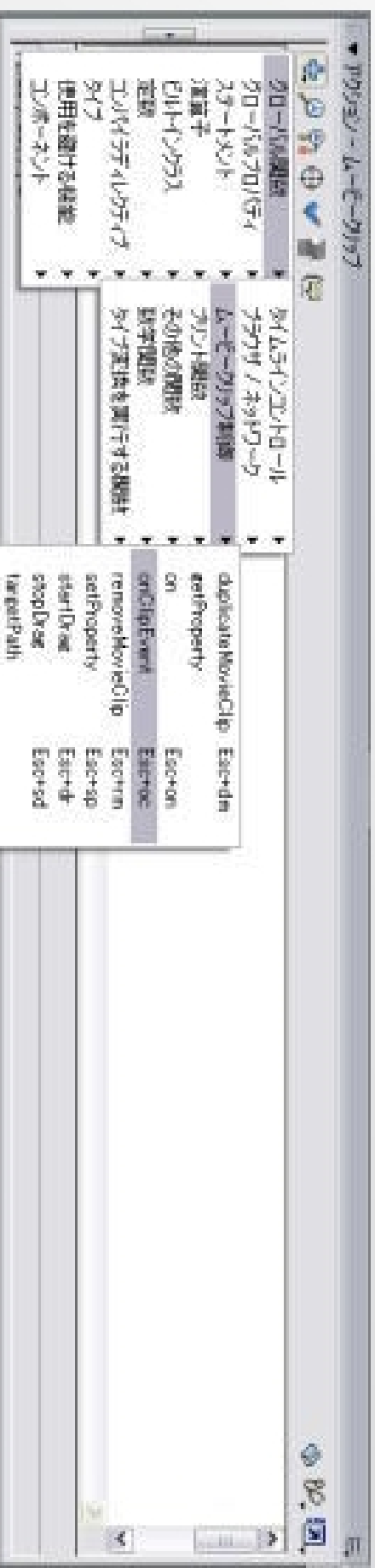
サイズ: 550 x 400 ピクセル

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。



ツ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。

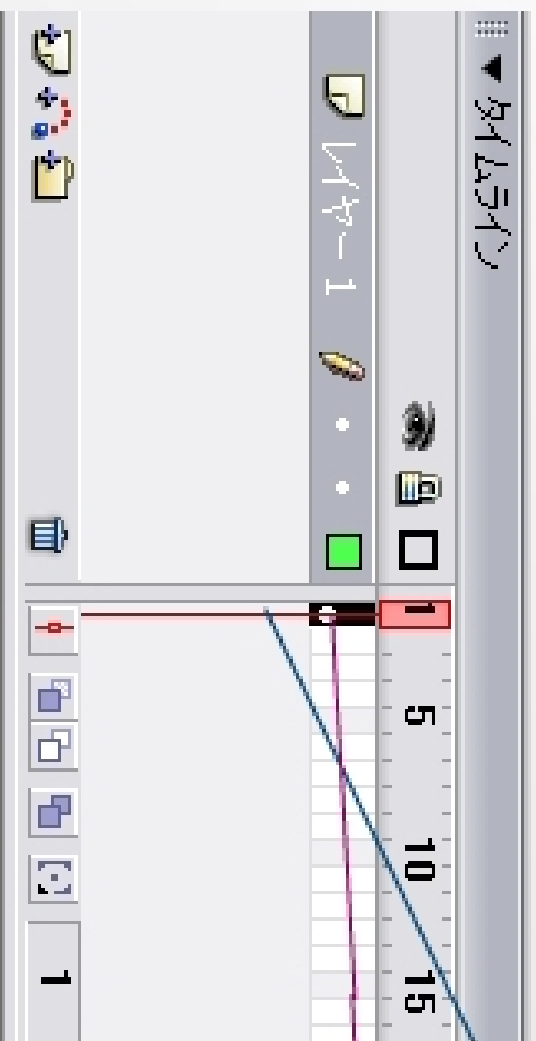
ムービークリップにスクリプトを記述するときには、

このように「onClipEvent({...})」の中に記述します。

描けたら、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(x...)"の{...}の中に記述します。

(1)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方向へも動か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。
ズレビューすると、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。

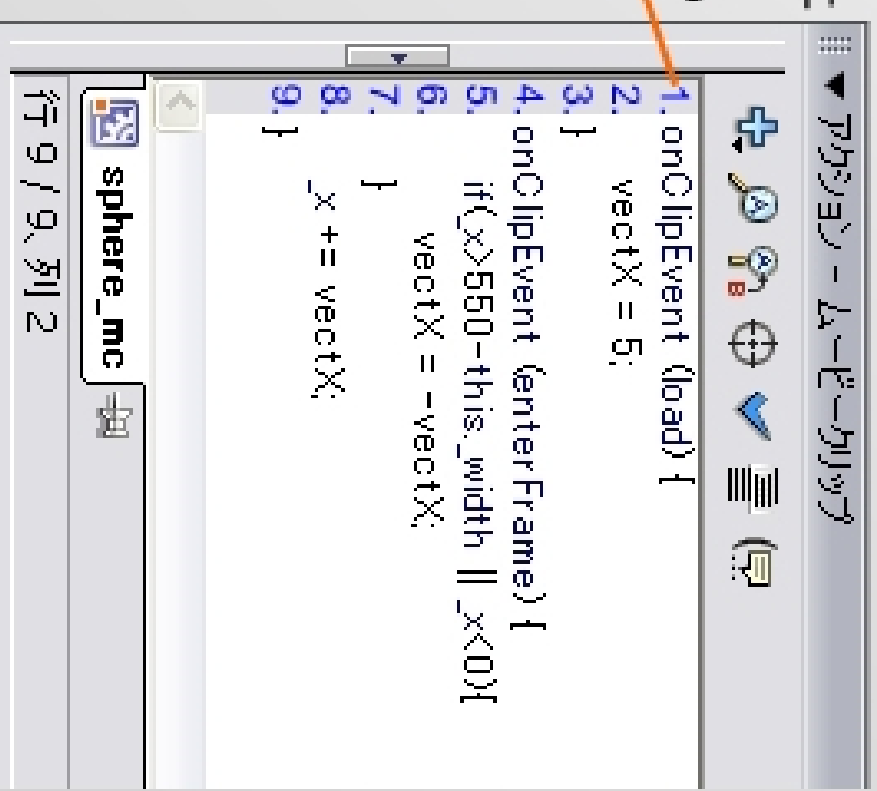
"`onClipEvent(load){...}`"中のスクリプト :

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。

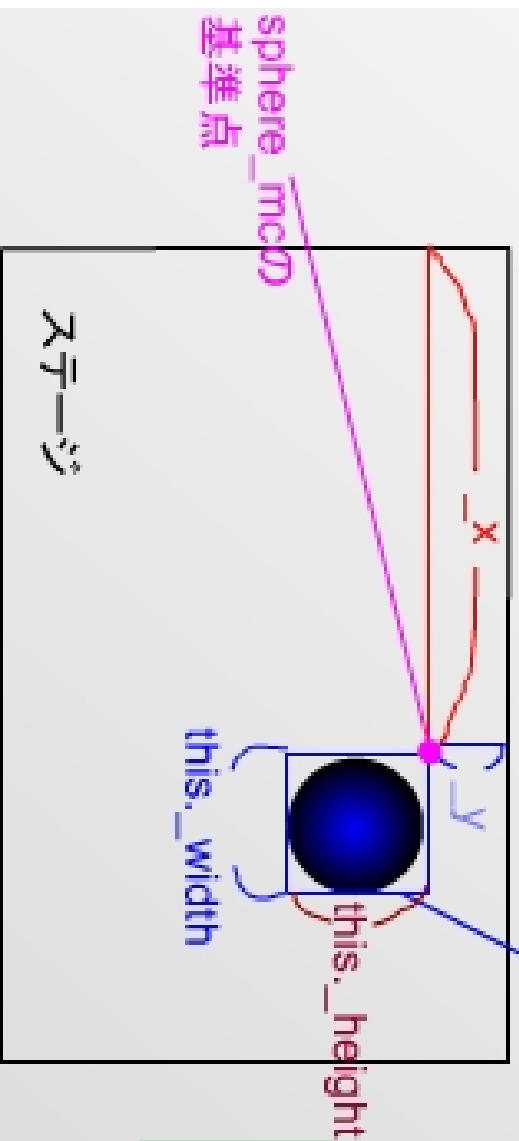
"if(条件){...}"文で、sphere_mcの右端がステージの右端より右に出た時と、sphere_mcの左端がステージの左端より左に出た時に、**vectX**の符号を逆にするように設定しています(下図参照)。

"||"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)



```
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. }
```

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。

(x,y)=(550,400)