

ActionScriptというのは、Flashで使用されるスクリプト言語です。スクリプトというのは、一種のプログラムです。Webページ制作のときに使われるJavaScriptという有名なスクリプト言語があります。少なくとも名前くらいは聞いたことがあるか、あるいは簡単なJavaScriptなら書いたことがあるという人も少なくはないでしょう。

JavaScriptを書いたことがある人は、ActionScriptを見ると、とても似ていると思われるでしょう。実際、ActionScriptはJavaScriptと同じ仕様に基づいて、設計されています。ですから、JavaScriptを知っていればActionScriptにはそれほど違和感なく入っていけるはずで、JavaScriptを知らない人にはチャンスです。ActionScriptを覚えれば、JavaScriptのハードルもずっと低くなります。

プログラムの基本的な考え方というものは、それほどむずかしいものではありません。しかし、どこから手をつけていいのかわからなくなってしまう人もいます。ニューヨークに行きたいけど、どうやって行ったらいいかさっぱりわからない、と騒ぐタイプの人です。おおげさにとらえ過ぎてしまっているために、一向に先に進みません。

旅慣れない人には、当然戸惑いはあるでしょう。でも、まずチェックリストを作ればいいのです。ニューヨークのある州と都市、国際空港、そこに乗り入れる航空会社とその便はどこか。他方で、パスポートやビザなどの手続き、宿泊先の手配、お金や持ち物はどうするか。チェックすべき項目を、とにかく並べ上げます。それらをすべてつぶしていけば、結果として目的地に必ず着きます。

プログラムも同じで、やり方がわからないとただ騒いでも進展はありません。やりたいことを実現するには、何と何ができなければならないのか。それらのうち、どれが自分にできていて、どれを新しく覚えなければならないのか。それらを明確にすれば、やるべきことというのは、それほどおおげさなものではないはずです。

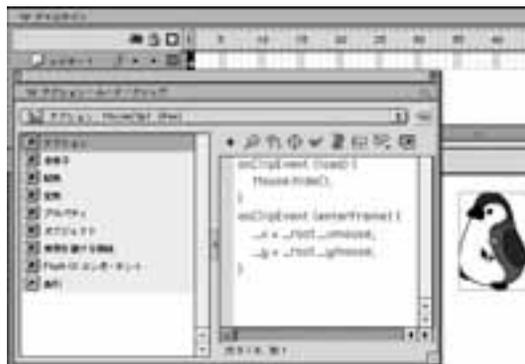
本書は、ActionScriptの書き方と、処理を組み立てる考え方に重点を置いて解説しています。ActionScriptを使わないと実現できない処理の大きなものはインタラクティブティ(ユーザーの操作に対応した動作)です。その中でも、マウス操作に対応したインタラクティブなアニメーションをおもな題材としました。応用範囲が広く結果もすぐに確認できるからです。

第3章を除き、サンプルはすべてタイムラインを1フレームしか使っていません。つまり、純粋にActionScriptだけでアニメーションを実行しています。スクリプトの処理を全体として把握するには、この方がわかりやすいと考えたからです。また、特に第4章以降は、Flash MXの新しいスクリプティングスタイルを積極的に採用しました。そのため、基本的に処理をすべてタイムラインの第1フレームに記述するということが可能になったのです。

各章のサンプルを、簡単に紹介しておきましょう。

「第1章 マウスに追隨する - カスタムカーソル」では、最もシンプルな、マウスに追隨するアニメーションを作成します。まずは、とにかくActionScriptで、動くものを作ってみようというわけです。

マウスに追隨するアニメーションは、マウスポインタを隠してしまうとカスタムカーソルとして使うこともできます。



マウスポインタの座標を調べて、ムービークリップインスタンスの座標を、それに対応して設定することがポイントです。スクリプトを作成するにあたって重要な概念となる「インスタンス」についても解説します。「アクション」や「ハンドラ」といったFlashの基本的な用語もここで紹介します。

「第2章 マウスに遅れて追いかける」では、マウスポインタに若干のタイムラグをおいて追隨するアニメーションを作成します。第1章のアニメーションを、視点を変えて応用します。複数のムービークリップインスタンスを使って、遅れの度合を少しずつ変えれば、見えないゴムやバネでマウスポインタに結びついているような動きをします。

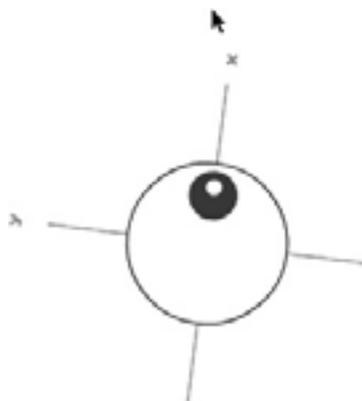
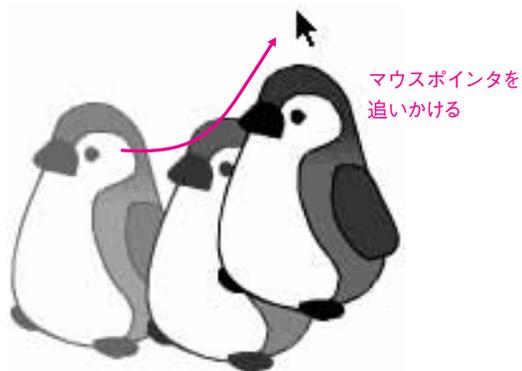


ムービークリップインスタンスから見たポイント座標のとらえ方と遅れを表現する簡単な計算式がポイントです。また、「変数」についてもここで紹介します。設定した値をメモリに保持する変数は、スクリプトではとても重要な機能です。ここでしっかり押さえておきましょう。

「第3章 マウスの方向を向く - キョロ目」では、マウスポインタの方向に向かって、自動的に目玉を回転させるムービークリップのスクリプトを作成します。

目玉を回転させるためには、その角度を計算しなければなりません。そこで登場するのが、数学の時間の悪夢「三角関数」です。聞いただけで意識を失いそうな人も心配する必要はありません。ActionScriptが用意してくれたその関数に座標を放り込みさえすれば、角度を教えてくれます。座標と角度の扱い方だけ覚えれば、三角関数の理屈を知る必要はありません。

エンジンやギアの仕組みがわからなくても、車の運転はできます(もっとも、筆者はペーパードライバーです)。



ただ、エンジンの仕組みも理解した上で運転したいエンジニアタイプの人もいるでしょう。そういう人のために、理屈の部分も簡単に解説しました。ちょっとがんばって覚えていただきたいのが、「ユーザー定義関数」(あるいは「function」)です。これは、ActionScriptの「坂道発進」ともいえます(くだいようですが、筆者はペーパードライバーです)。

「function」の使い方をマスターすると、スクリプティングの世界がまったく変わります。ほかのActionScriptの解説書(Flash 5)を見ると、もっと後半で紹介されることが多いようです。しかし、本書ではそれをあえて前半で解説しました。特に、新しいFlash MXスタイルのスクリプティングを行うには、「function」の理解は避けて通れないと考えたからです。「夜明け前が一番暗い」(It's always darkest before the dawn)といえます^{*1}。また「艱難汝を玉にす」ということわざもあります。ここはひとつ体調を整えて、乗り切ってください。できれば、頭のすっきりした午前中に読まれることをお勧めします。

この章で実際に作成する「function」を、ちょっと見てみましょう。

```
onClipEvent (enterFrame) {
    _rotation = Math.atan2(_parent._ymouse*_parent._xmouse*_x);
    _rotation = this._rotationToDegrees(_rotation);
}

onClipEvent (click) {
    function _rotationToDegrees(nRotation) {
        _Degree = nRotation/180*Math.PI*180;
        return _Degree;
    }
}
```

「第4章 アニメーションするボタン」のトピックは、アニメーションでなく「ボタン」の方にあります。Flash 5では、ボタンは「ボタンシンボル」で作成しました。Flash MXでも、もちろんこの方法でボタンを作ることができます。しかしFlash MXではさらに、Flash 5ではむずかしかった「ムービークリップシンボル」でも、簡単にボタンを作ることができるようになりました。



*1：もっとも、このような自然現象が科学的に認められるかといえば、疑問です。実際には、夜明け前はそれなりに明るいようです。

「ムービークリップ」では、Flash MXの新しいスクリプティングスタイルを使うことができます。Flash 5では、ムービークリップのフレームやインスタンスに分散して記述しなければならなかったスクリプトを、すべてフレームにまとめることが可能になったのです。



この章で作成するのは、まだシンプルなスクリプトです。本書の後半では、このスタイルを使った本格的なスクリプティングを紹介します。また、この章からFlash MXに付属しているオンラインヘルプの「ActionScript辞書」を頻繁に引用します。「辞書」とあるとおり、ActionScriptのキーワードや命令が、その項目ごとに分けて解説されています。海外ソフトのローカライズにありがちな直訳風の文章や、ときおり見つかる誤りは気になるものの、スクリプトを記述する際には必ず参照する必要があります。



ただ、初心者にはその見方や用語が理解しにくいことも確かです。そこで本書は「ActionScript辞書」から必要な部分を引用して、解説を加えるようにしました。わかりづらそうな翻訳は、筆者なりに訳し直して解説している部分もあります。また、誤りと思われる記述は訂正しています（もっとも、中には筆者の誤解がある可能性もありますが、その解釈はあくまで筆者の責任によるものです）。

「第5章 マウスの位置に応じてスクロール」では、マウスポインタがコンテンツの中心から離れるほど、スピードを増してスクロールするアニメーションのスク립トを作成します。条件が1つだけの簡単なスク립トからスタートして、最終的には複数の条件を組み合わせたものを完成させます。



この章では、条件に応じて処理を分岐する「if」アクションを紹介します。条件判定は、ActionScriptでは「function」と並ぶ重要ポイントです。

はじめの一歩



できあがり



「第6章 広がりながら消えていくイメージ」では、何もないステージに、スク립トを使ってライブラリからムービークリップシンボルを配置します。ステージをクリックすると、ムービークリップがその位置に現れてアニメーションが実行されます。イメージが拡大するとともに薄くなって消えていきます。

もちろん、メインのタイムラインもムービークリップシンボルも1フレームしかありません。



数多くのシンボルをステージに配置すると、処理は当然重くなります。スクリプト的に、その処理を軽減するポイントも紹介します。さらにその応用として、クリックした位置から波紋のように複数のムービークリップが現れては広がるスクリプトも作成します。

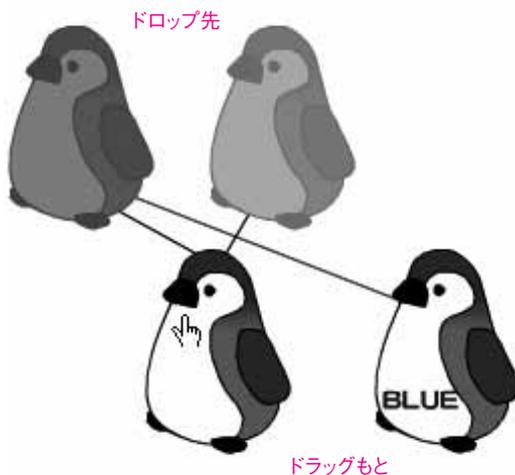
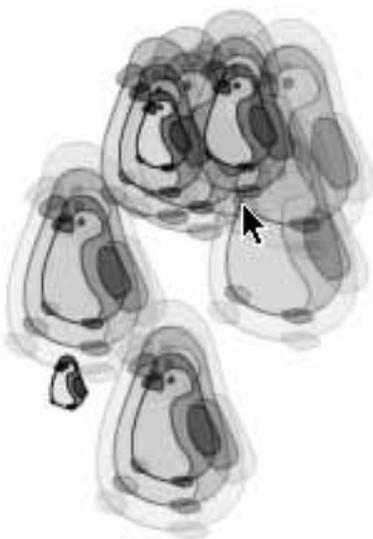
このあたりのスクリプトになると、配置したムービークリップの数を数えたり、タイミングを計ったり、スクリプトとしてコントロールする要素が増えてきます。その意味では、ぐっとプログラミングらしくなります。

「第7章 ドラッグしてスナップ — 重ね合わせる」は、ドラッグしたムービークリップが一定の領域上でスナップしてぴったりと重なり合うというインターフェイスを、スクリプトで作成します。

ポイントは、ステージ上に配置した複数のムービークリップ間で、相手の情報を確認しながら身の振り方を決めるスクリプティングです。ドラッグしたムービークリップが、ドロップ先に重なると、スナップして位置がそろいます。

さらに条件を加えて、ドロップできる組み合わせを指定するテクニックも紹介します。

この段階では、スクリプトを1つの部品のように完成させて、後々活用したり、あるいは機能の変更・拡張にも対応できるアプローチを紹介します。「オブジェクト指向」あるいは「モジュール化」といわれる考え方です。



「第8章 データをまとめて処理する」では、第7章のスクリプトを別の視点で応用します。「配列」や「グローバル変数」、「ループ処理」といったスクリプティングに役立つ概念やテクニックを紹介します。

これらの知識やテクニックを、実際にスクリプトを作成しながら学習していきます。本書ではいきなりスクリプトを完成させずに、小さなステップを進めてはその内容を解説するという方法を採用します。まだるっこしさを感じる人もいるかもしれません。しかし、急にニューヨークに旅行しようとしても、パスポートを忘れてたり、宿の手配をまちがえたりなど、旅慣れた人でも準備をしなければ失敗します。必要なものをひとつひとつそろえていって、その上で実行するというのが大切です。本書に書いたような手順や確認事項を踏まえた上で拡張をしていくというプロセスは、筆者がスクリプトを実際に組むに当たって基本的に採っている方法でもあります。

本書の特徴のひとつに、コラムがあります。本来、コラムというのは、ひとくちメモやおまけ的な欄です。ところが、書いていくうちに数ページにわたるものが続出してしまいました。そうすると、どちらが本文なんだか、わからない状態です。結局ページ数を気にするのは諦めました。その代わりに、本文で扱えなかったテクニックや少し突っ込んだ話題も、思い切って取り上げています。

それでは、さっそくスクリプトを作成していくことにしましょう。