

Part 1 入門編

Chapter

1

はじめての
ActionScript 3.0

この章ではまず ActionScript 3.0（以下、AS3）とは何か、そもそもプログラミングとはどういった仕組みで動作しているのかを簡単に説明していきます。後半には、サンプルプログラムも用意しています。チュートリアル形式で実際の AS3 プログラミングを体験してみましょう。

1-1 ActionScript 3.0 とは

AS3 は、Flash 用のプログラミング言語です。プログラミングとはプログラムを書くことなのですが、ここではまずプログラミングをはじめて学ぶ方のために、プログラムの概要を簡単に説明します。

■ プログラムとは

プログラムとは、コンピュータがおこなう仕事の流れを示した「手順書」のようなものです。まずは私たちの日常生活の一場面を、簡単なプログラムに置き換えてイメージしてみましょう。

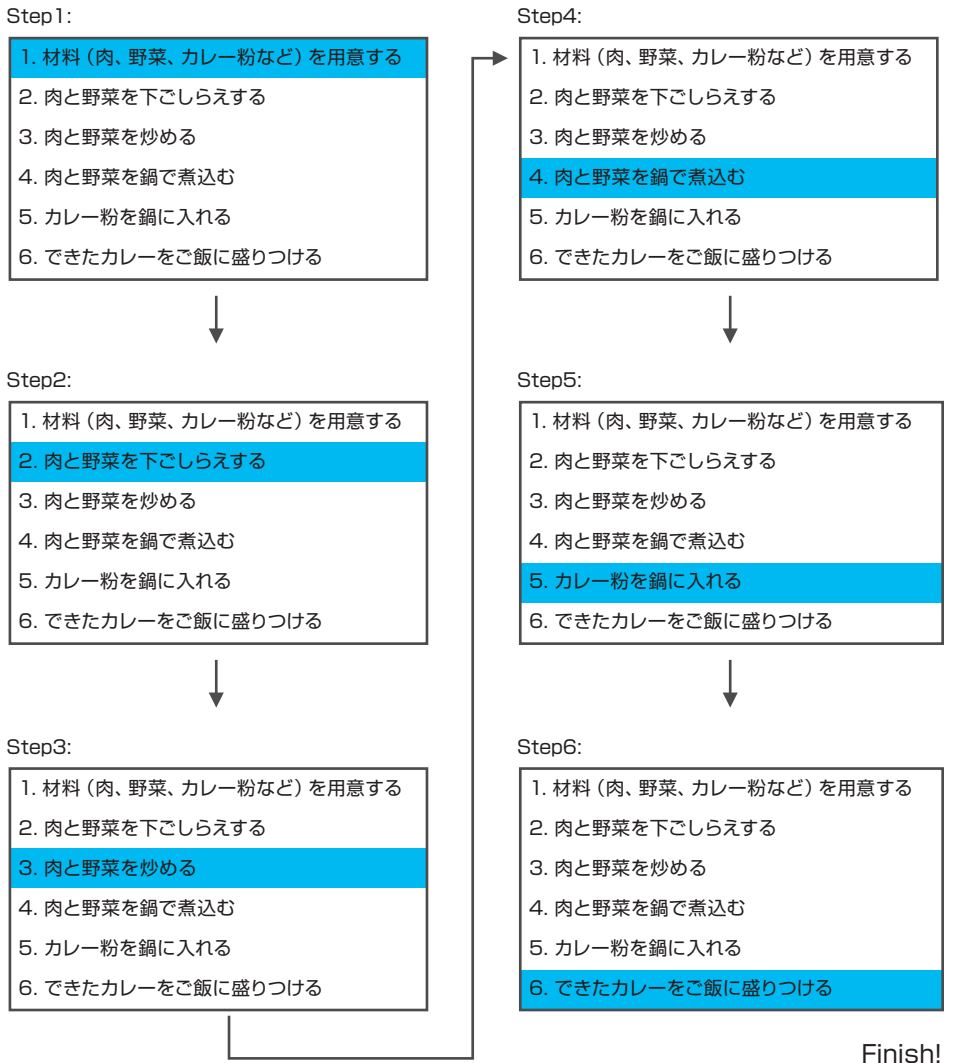
■ 「カレーライスを作る」プログラム

「カレーライスを作る」というプログラムと、それを実行するときの流れを次に示します。

「カレーライスを作る」

1. 材料（肉、野菜、カレー粉など）を用意する
2. 肉と野菜を下ごしらえする
3. 肉と野菜を炒める
4. 肉と野菜を鍋で煮込む
5. カレー粉を鍋に入れる
6. できたカレーをご飯に盛りつける

カレーライスを作るには、この6つのステップを上から順におこないます。それぞれのステップは、「カレーライスを作る」という目的を達成するために、私たちがおこなうべき作業を指示しています。いわば、私たちに対する命令文です。カレーライスをきちんと作るためには、これらの命令文を適切な順序で実行しなければなりません。



図●「カレーライスを作る」プログラムの処理の流れ

プログラムとは、このような「命令文」と「処理の流れ」を記述したものと考えることができます。

「Google のロゴを表示する」プログラム

プログラムの概要を理解したら、次に Flash のプログラムを考えていきます。

Flash への仕事の指示書「Google のロゴを表示する」というプログラムは次のように書くことができます（Flash に日本語の文章は通じないのですが、便宜上日本語で説明しています）。

「Google のロゴを表示する」

1. ロゴを読み込むための入れ物を用意する。入れ物には「logo」と名前を付けておく
2. 用意した「logo」に Google のロゴ（gif ファイル）を読み込む
3. 「logo」をステージに表示する

ここで気がつくことが2つあります。

第一に、Flash のプログラムを書くには、まず「Flash に何ができるか？」を知っておく必要があります。上の例では、Flash で gif ファイルを扱うことができるのかどうかを知らなければ、そもそもプログラムを書き始めることはできません。

第二に、Flash に正しく処理させるための「Flash 流ルール」を理解しておく必要があります。画像ファイルを読み込むには「まず入れ物が要る」というのは、いってみれば Flash 側の都合です。しかし、このルールを理解しておかないと、正しい命令文を書くことはできません。

つまり、Flash のプログラムを書くためには、「Flash にできること」と「Flash 流ルール」の理解が必要なのです。

次に実際のプログラムを見てみましょう。

Flash のプログラムは、Flash 用のプログラミング言語である ActionScript で記述します。

プログラミング言語の文法は、私たちが普段使う言語とはまったく異なります。プログラミング言語を習得するためには、新しい外国語を学ぶように、その言語そのもののルールや文法を理解する必要があります。

上のプログラムは、ActionScript の最新バージョンである AS3 を使うと、次のように書くことができます。

List Google のロゴを表示する google fla

```
var logo:Loader = new Loader();
logo.load(new URLRequest("http://www.google.co.jp/intl/ja_jp/images/logo.gif"));
addChild(logo);
```

このプログラムを実行すると、Flash は次のような流れで命令を処理していきます。

Step1: ロゴを読み込むための入れ物を用意する。入れ物には「logo」と名前を付けておく

```
var logo:Loader = new Loader();
logo.load(new URLRequest("http://www.google.co.jp/intl/ja_jp/images/logo.gif"));
addChild(logo);
```



Step2: 用意した「logo」にGoogleのロゴ (gifファイル) を読み込む

```
var logo:Loader = new Loader();
logo.load(new URLRequest("http://www.google.co.jp/intl/ja_jp/images/logo.gif"));
addChild(logo);
```



Step3: 「logo」をステージに表示する

```
var logo:Loader = new Loader();
logo.load(new URLRequest("http://www.google.co.jp/intl/ja_jp/images/logo.gif"));
addChild(logo);
```

Finish!

図●プログラム「Google のロゴを表示する」の処理の流れ

すべての処理が正常に実行されると、Google のロゴが画面に表示されます。



図●実行結果 google.swf

プログラムを書く人は「プログラマー」と呼ばれますが、この本では ActionScript のプログラマーを便宜的に「ActionScripter (以下、ASer)」と呼ぶことにします。

また、ActionScript はプログラミング言語の中でも特に「スクリプト言語」と呼ばれることがあります。この本では「プログラミング言語」という呼び方で統一して説明しています。

Part 1

1

2

Part 2

3

4

5

6

Part 3

7

8

付録

A

B

■ ActionScript の歴史

もともと Flash は、アニメーションを作るためのソフトウェアとして開発されました。初期の Flash には、ActionScript の機能はなく、Flash 5 で最初のバージョン「ActionScript 1.0」が搭載されました。ActionScript 1.0 はアニメーションを補助するための簡単なプログラミング言語でした。

次に、Flash MX 2004 で「ActionScript 2.0」が搭載され、Flash でより複雑な処理を扱えるようになりました。

Flash CS3 から搭載された ActionScript の最新バージョンである「ActionScript 3.0」では、さらに高度なことが可能になりました。

Flash CS4 では、3D 回転ツールやボーンツールなど、デザイナーのための機能が搭載されましたが、これらは ActionScript 3.0 のみをサポートしており、現在では ActionScript 3.0 は Flash デザイナーにとっても必須のスキルとなっています。

■ AS3 の特徴

AS3 は、**オブジェクト指向言語**に分類されるプログラミング言語です。

AS3 では、**オブジェクト指向**という考え方に基づいて、プログラムを作っていくことになります。オブジェクト指向については Chapter 2 で説明していきます。

■ AS3 に必要な環境

AS3 でプログラムを作るには、Flash CS3 または CS4 が必要です^{※1}。この本では Flash CS4 を使って AS3 を記述する際の手順を解説していきます。

また AS3 で作られたプログラムは、Flash Player 9 以降のバージョンでのみ動作します。

Flash Player 8 以前では動作しないので、公開時には気をつけましょう^{※2}。

※1 Flash 以外では、Flex Builder や FlashDevelop といった開発環境があります。

※2 本書執筆時における日本での Flash Player 9 の普及率は 99.0% です。各バージョンの最新の普及率は下記サイトでチェックできます。

Adobe - Flash Player: Version Penetration

(http://www.adobe.com/products/player_census/flashplayer/version_penetration.html)

1-2 はじめての ActionScript 3.0 プログラミング

それではいよいよ、AS3 プログラミングを始めます。

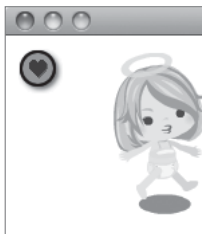
AS3 プログラムを作成する一番簡単な方法は、Flash コンテンツと同様、fla ファイル内にスクリプトを記述していくことです。この fla ファイルをパブリッシュして生成した swf ファイルに、実行可能な AS3 プログラムが含まれる形になります。ここでもこの方法ではじめての AS3 プログラムを作成します。

まずはこの章のサンプルとして用意しておいた **before.swf** を実行してみましょう。実行すると、中央の女の子の-avatarがその場歩きを始めます。左上のボタンを押しても何も起きません。

次に、同じフォルダにある **after.swf** を実行してみましょう。今度は実行しても、アバターは立ち止まったままですが、左上のボタンをクリックすると、アバターが右に歩き始めます。



図● 実行結果：アバターがその場歩きしている **before.swf**



図● 実行結果：ボタンをクリックすると、アバターが右に歩き出す **after.swf**

次のページ以降のチュートリアルでは、この before.swf のアニメーションに変更を加えて、after.swf を作成する手順を紹介します。実際のプログラムは、before.swf のもとになる **base fla** に 10 行程度のコードを記述していきます。以降ではこの fla ファイルを使って説明していきます。

Part 1

1

2

Part 2

3

4

5

6

Part 3

7

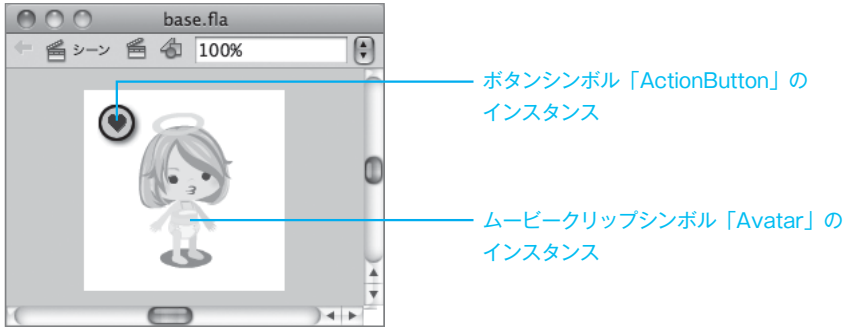
8

付録

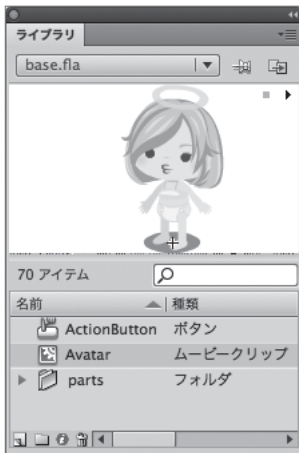
A

B

Flash CS4 で **base fla** を開くと、ステージ上にはムービークリップシンボル「Avatar」のインスタンスと、ボタンシンボル「ActionButton」のインスタンスが配置してあります。パブリッシュプレビューを表示すると、**before.swf** と同じようにアバターがその場歩きを始めます。



図●ステージ：ムービークリップシンボル「Avatar」のインスタンスとボタンシンボル「ActionButton」のインスタンスが配置されている **base fla**



図●ライブラリパネル **base fla**

■ プログラムを書くための確認

プログラミングを始める前に、ActionScript と Flash Player のバージョンを確認します。プロパティパネルの [パブリッシュ] セクションを開き、Player が「Flash Player 9」に、スクリプトが「ActionScript 3.0」になっていることを確認しましょう。Flash Player のバージョン設定は、Flash Player 9 か Flash Player 10 であれば OK です。Flash Player 10 は Flash CS4 でのみ設定できます。この本では、Flash Player 9 で進めます。

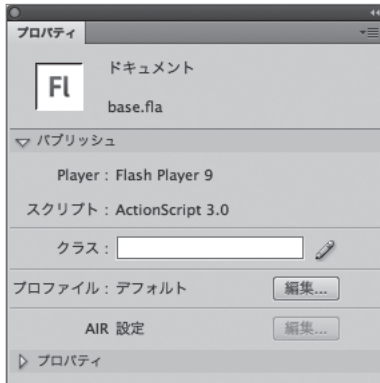


図 ● プロパティパネル base.fl a

MEMO

これ以外の設定になっているときは、パブリッシュ設定を変更してください。設定を変更するには、メニューから [ファイル] - [パブリッシュ設定] を選択し、[パブリッシュ設定] パネルの [Flash] タブにある [Player] と [スクリプト] で適切な項目を選択します。

■ プログラムを書く場所

プログラムの書き方は、シンボルの配置とよく似ています。表示したいタイミングのキーフレームにシンボルを配置するように、命令を実行したいタイミングのキーフレームにコードを書いていきます。

今回はメインタイムラインの再正開始と同時に命令を実行したいので、1 フレーム目にコードを書きます。

タイムラインパネルを開き、メインタイムラインの as レイヤーの 1 フレーム目のキーフレームを選択しましょう。

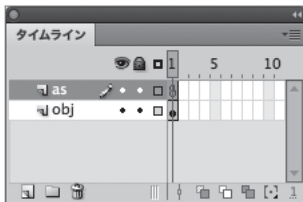


図 ● タイムラインパネル：as レイヤーの 1 フレーム目のキーフレームを選択する base.fl a

この状態のままメニューから [ウィンドウ] - [アクション] を選択して、**アクションパネル**を開きます。あらかじめ `trace("Hello AS3!");` というコードが書いてあります。ここが、実際にコードを書いていくエディタです。

このエディタは、AS3 プログラミングをサポートするいろいろな機能を持っています。たとえば `trace` や `"Hello AS3!"` の部分には色が着いています。これはカラーリングと呼ばれる機能で、特別な意味を持つ単語に自動で色をつけて、コードを読みやすくしています。

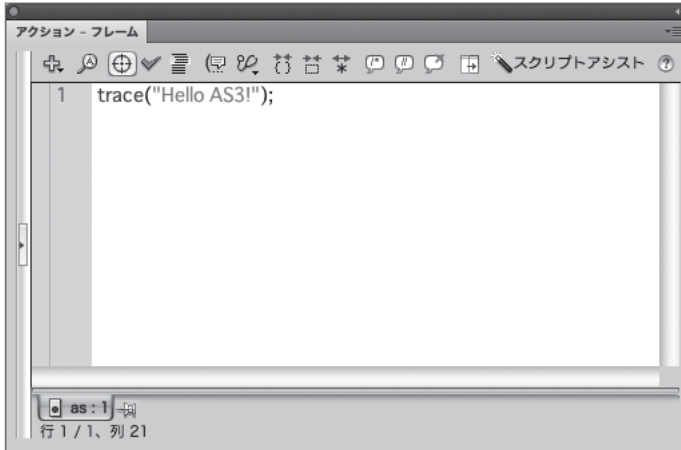


図 ● アクションパネル：このエディタにコードを書いていく base.fla

ちなみにこの `trace("Hello AS3!");` は、ムービープレビュー時の出力パネルに「Hello AS3!」と表示させるためのコードです。このような、ある1つの命令を表すコードは**ステートメント**と呼ばれます。

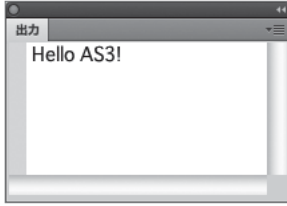
MEMO

出力パネルは、ムービープレビュー時にトラブルシューティングに役立つ情報などを表示するための場所です。メニューから [ウィンドウ] - [出力] を選択すると開きます。

■ プログラムを実行する

それでは、ムービープレビューをおこなって、プログラムの実際の動作を確認してみましょう。

AS3 プログラムの場合は、そのコードを書いたタイムラインが再生されるタイミングで命令が処理されます。今回の例では1フレーム目にコードを書いたので、実行してすぐに命令が処理されます。`trace("Hello AS3!");` は単純な命令なので、処理にかかる時間は一瞬です。出力パネルに「Hello AS3!」と表示されたでしょうか？



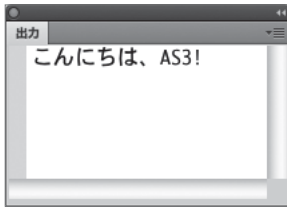
図● 出力パネル：「Hello AS3!」と表示される

ごく簡単なものですが、とりあえずはじめてのプログラムを動かすことができました。

プログラムは、あなたの指示を何度でも正確に処理します。このプログラムをあと何度実行しても「Hello」が「Hey」に変化することはありません。これがプログラムです。

■ プログラムの一部を変更してみる

先ほどの例の「Hello AS3!」の部分は、自由に変更することができます。日本語も OK です。「こんにちは、AS3!」に変更して動作を確認してみましょう。



図● 出力パネル：「こんにちは、AS3!」と表示される

出力パネルに表示されるこのようなメッセージは実際の Flash コンテンツに影響があるわけではないため、この処理に何の意味があるのか?と思われるかもしれません。

確かに Flash コンテンツを見る側のユーザーにとっては特に意味がないのですが、プログラムを作っていく過程で必ずお世話になるコードなので、`trace()` という命令で出力パネルにメッセージが表示されることを覚えておきましょう。

■ ステージ上のインスタンスを扱う場合の準備

ステージ上のインスタンスに命令を出したい場合は、プログラミングの前に、まずそのインスタンスの**インスタンス名**を設定します。

インスタンス名とは、プログラム上でそのインスタンスを識別するために必要な名前です。シンボル名とは異なるので注意してください。

インスタンス名の設定は次の手順でおこないます。

1. ステージ上のインスタンスをクリックして選択する
2. [ウィンドウ] - [プロパティ] でプロパティパネルを開く (まだ開いていない場合)
3. [<インスタンス名>] と表示されているエリアにインスタンス名を入力する