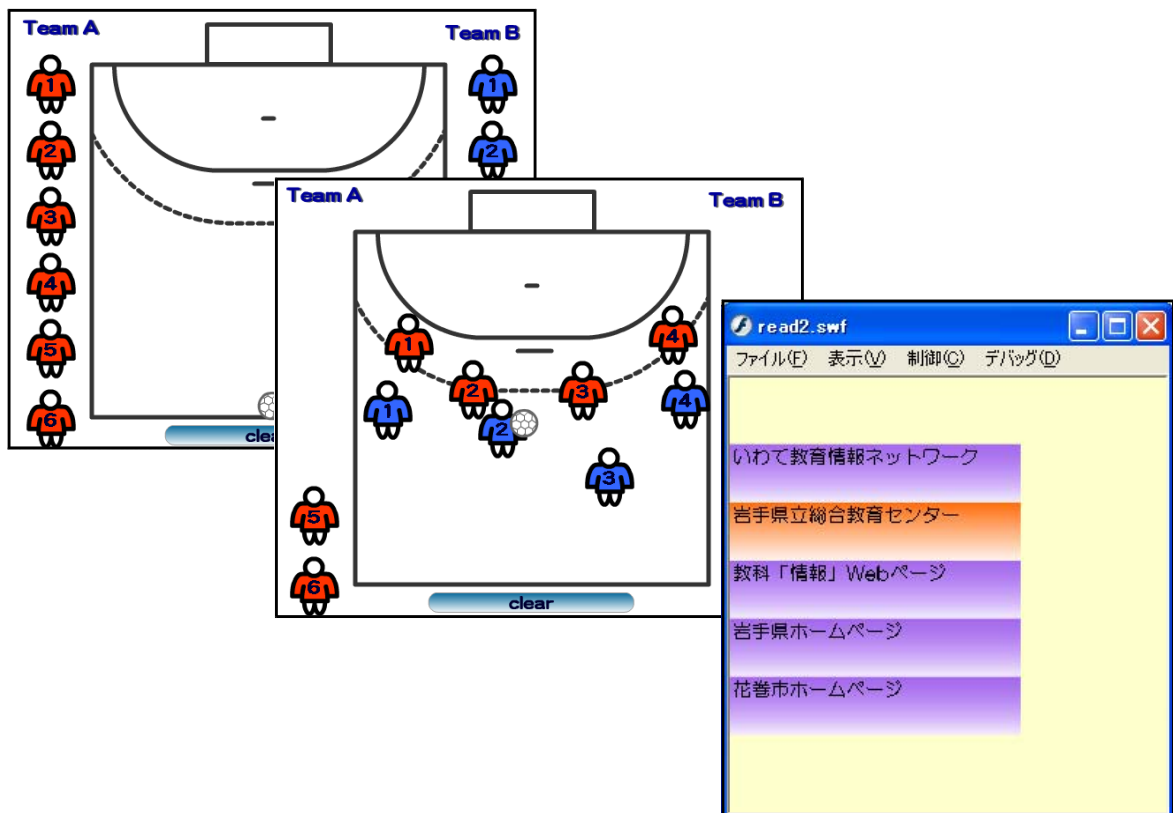


平成18年度

Flash

を活用した教材作成 研修講座

応用(ActionScript)編



岩手県立総合教育センター
<http://www1.iwate-ed.jp/>

| | | |
|-----------------------------------|----------------------------------|----|
| ■■■ 第1章 ActionScriptとは | | |
| 1 | ActionScript | 1 |
| 2 | ActionScriptが得意なこと | 1 |
| 3 | ActionScriptの実行環境 | 1 |
| 4 | ActionScriptの安全性 | 2 |
| 5 | ActionScriptの作成方法 | 2 |
| ■■■ 第2章 ドラッグ&ドロップの活用 | | |
| 1 | ドラッグ&ドロップできるムービーの作成 | |
| (1) | Flashの起動 | 3 |
| (2) | ムービークリップの作成 | 3 |
| (3) | ActionScriptの設定 | 4 |
| 2 | 応用：「作戦盤」の作成 | |
| (1) | 背景の作成 | 6 |
| (2) | 選手アイコンの作成 | 7 |
| (3) | 選手アイコンの作成2 | 10 |
| (4) | 選手アイコンの配置 | 13 |
| (5) | ActionScriptの設定 | 14 |
| | 【追加機能1】～ボールがあればいい！！ | 16 |
| | 【追加機能2】～選手アイコンがワンクリックで整列できればいい！！ | 17 |
| | 【追加機能3】～アイコンの重ね順を変えれば！！ | 21 |
| ■■■ 第3章 外部ファイル (txt) 読込の活用 | | |
| 1 | 外部ファイルからデータを読み込むムービーの作成 | |
| (1) | テキストフィールドの作成 | 23 |
| (2) | ActionScriptの設定 | 24 |
| (3) | 外部ファイルの作成 | 25 |
| (4) | 動作確認 | 25 |
| 2 | 応用：「リンク集」の作成 | |
| (1) | ドキュメントのサイズ変更 | 26 |
| (2) | ボタンシンボルの作成 | 26 |
| (3) | ファイルの保存 | 28 |
| (4) | テキストファイルの作成 | 28 |
| (5) | ファイルの読込設定 | 29 |
| (6) | リンクの設定 | 30 |
| ■■■ 参考資料 ActionScript | | 31 |

第1章 ActionScriptとは？

1 ActionScript

ActionScript とは、**Flash** に搭載されたオブジェクト指向のスクリプトです。**Flash** で作成したムービーに対して、命令を記述して実行することにより、ムービーを制御し、今まで以上にインタラクティブなコンテンツとして表現することができます。

2 ActionScriptが得意なこと

ActionScript を使用すると、通常の **Flash** ムービーに次のような機能や処理を追加することができます。

- ムービークリップの位置・サイズ・色などを動的にコントロール
- 点数計算や移動場所の座標計算
- タイムラインのコントロール
- オプションボタンやリストボックスなどの使用
- マウスの位置に合わせたムービークリップ制御
- ムービークリップのドラッグ&ドロップ
- ムービークリップの入れ替え
- キーボードのキー判断
- サウンドコントロール
- 文字列表示
- 外部テキストの読み込み
- CGI 利用
- クラスやオブジェクトの自作



左図のように、特定の粒子が画面内をランダムに動き回る場合を考えてみましょう。壁にぶつかって反射し、粒子どうしでぶつかって反射するような動きは、その瞬間での位置について衝突判定を判別する必要があります。

このようなコンテンツを作成する場合には、**ActionScript** が得意とするところです。

3 ActionScriptの実行環境

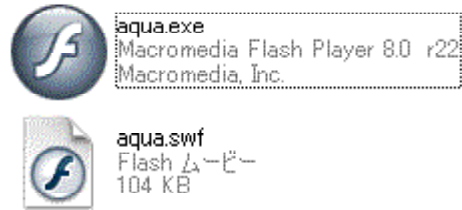
Flash ムービーの配布形式には、大きく分けて2つのパターンがあります。**ActionScript** を使用している場合も、同じように配布を行うことができます。

●ブラウザ上で再生

ブラウザ上で再生できる「*.swf」形式で配布する方法です。多くのブラウザは **Flash** ムービーを再生するためのプラグインや **ActiveX** コントロールを標準で備えているため、どのようなブラウザで見ても、**Windows**、**Macintosh**、そして **UNIX** でさえも同じ **Flash** ムービー + **ActionScript** で同じようにコンテンツを表示することができます。

●プロジェクトで再生

「プロジェクト」と呼ばれる実行形式のファイルで再生する方法です。プロジェクトは **Flash** の開発環境があればすぐに作成することができます。ブラウザで再生する場合には、ブラウザがなかったり、最新のプラグインや **ActiveX** コントロールがなかったりする場合には再生されませんが、実行形式のプロジェクトならば単体のファイルのみでムービーを再生することができます。



4 ActionScriptの安全性

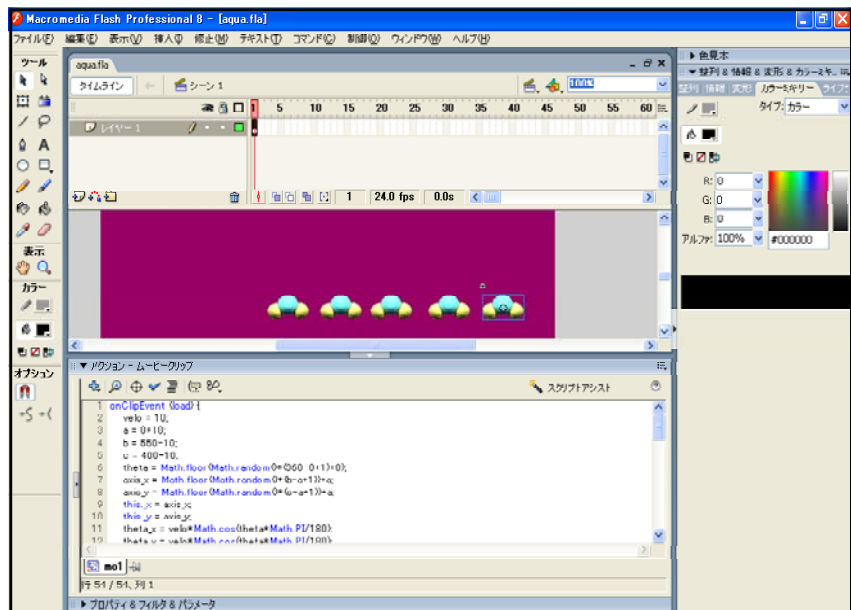
Web に公開する場合に編集するファイル (*.fla) を **Flash** ムービー (*.swf) に変換しますが、このとき **ActionScript** も書いたスクリプトそのままではなく、**FlashPlayer** 用のコードに変換されます。**swf** 形式はオープンフォーマットで、そのファイルの仕様は公開されています。そのため、コードから **ActionScript** へ戻す解析ツールもあるので、パスワードや重要な情報は、**ActionScript** 内に残さないようにしましょう。また、複雑な処理も容易に書けるようになりましたが、あまりに重い処理をさせると **FlashPlayer** 側で **ActionScript** の実行を停止して、**PC** 全体に影響を及ぼさなくなっています。

5 ActionScriptの作成方法

ActionScript を記述するには、まずムービーやムービークリップ、サウンドを用意しておいて、それに対して動きを付けていきます。

具体的に **ActionScript** を記述する作業は、「アクションパネル」を使用します。作成したムービーのタイムラインやムービークリップなどを選択し、メニューから [ウィンドウ] → [アクション] とたどるか、[F2] キーを押すとした図のようなアクションパネルが表示されます。

このアクションパネルを使用してスクリプトを記述していきます。アクションパネルは、選択した場所（フレーム、インスタンス、ボタンなど）に対してスクリプトを記述し、すでにスクリプトが記述されている場合には、そのスクリプトを表示します。



第2章 ドラッグ&ドロップの活用

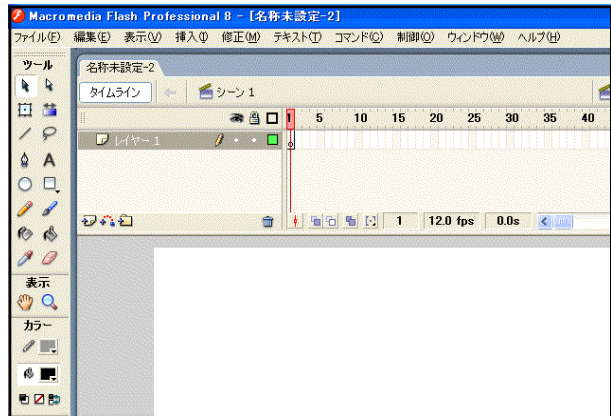
1 ドラッグ&ドロップできる ムービーの作成

(1) Flash の起動


(a) [スタート] から [すべてのプログラム] - [Macromedia] - [Macromedia Flash 8] をクリックします。

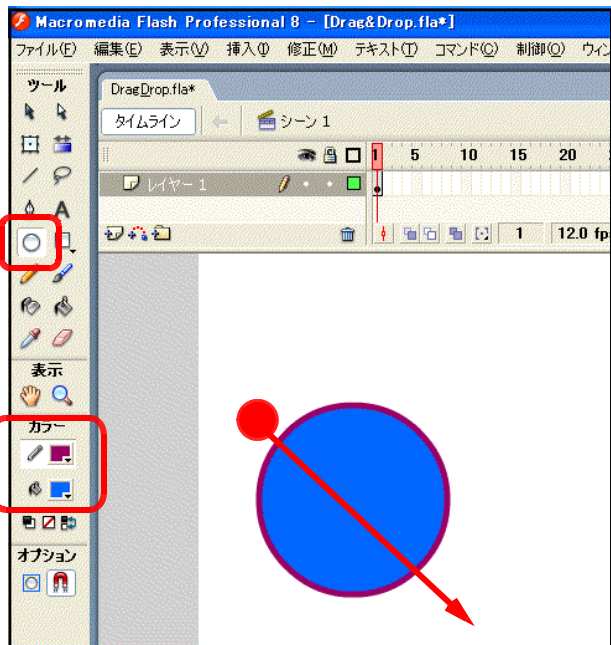
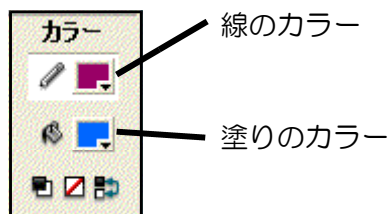



(b) Flash のメニューバーから [ファイル] - [新規] もしくは、[Flash ドキュメント] をクリックします。

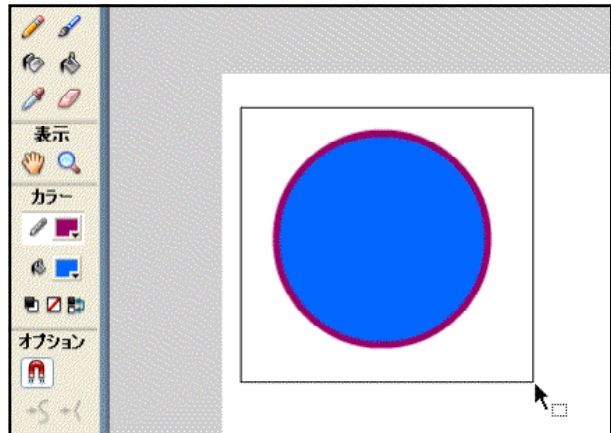


(2) ムービークリップの作成

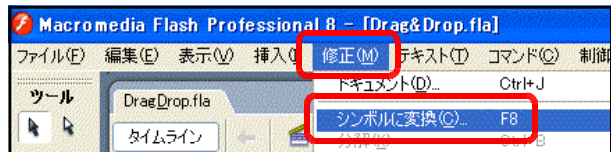
(a) ツールパネルの [楕円ツール]  をクリックして、キャンパス上に円を描きます。このとき、事前に [線のカラー] と [塗りのカラー] を設定しておきます。



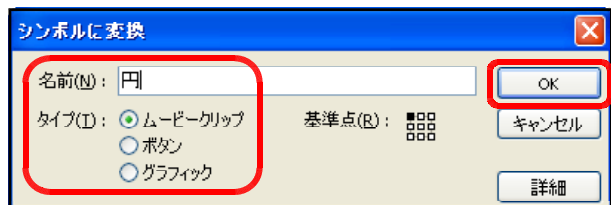
- (b) ツールパネルから [選択ツール]  をクリックして、円全体を範囲指定します。



- (c) メニューから [修正] - [シンボルに変換] をクリックします。



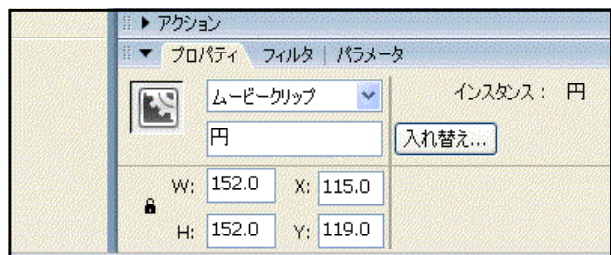
- (d) [シンボルに変換] ダイアログが表示されますので、[名前:] を『円』、[タイプ:] は『ムービークリップ』にチェックして、[OK] ボタンをクリックします。




- (e) 画面下の [プロパティ] をクリックし、プロパティパネルを表示させます。

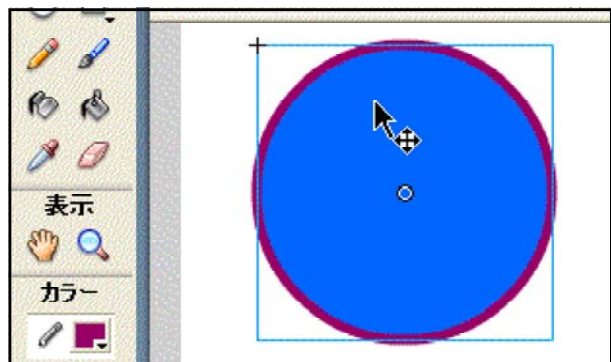


- (f) ムービークリップに『円』と名前を入力します。これで描画された円は、シンボルとして変換され、『円』という名前を付けました。

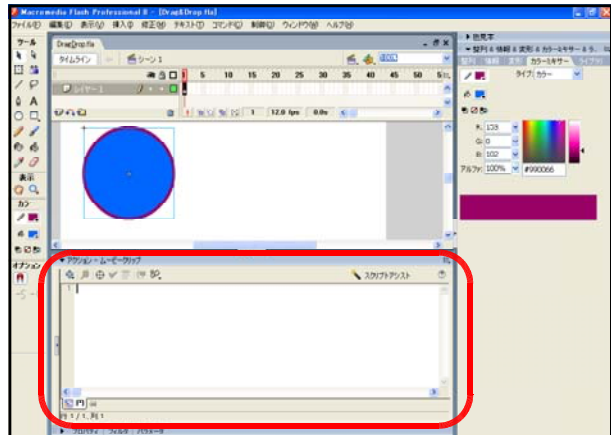


(3) ActionScript の設定

- (a) ツールパネルから [選択ツール]  をクリックし、円のムービークリップをクリックして選択します。



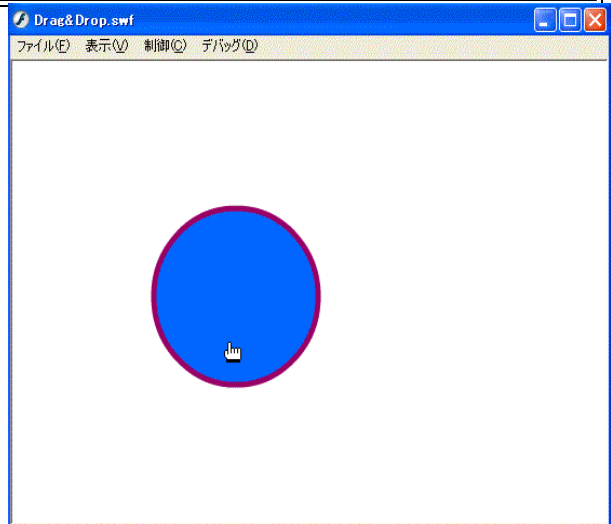
- (b) 画面下の [アクション] をクリックして、アクションパネルを表示させます。



- (c) 以下のスクリプトをアクションパネルに記述します。

| | |
|--|---|
| <pre>on (press) { this.startDrag(); }</pre> | <p>※ マウスのボタンが押されたら (press)、このインスタンスは、ドラッグ開始 (startDrag) です。</p> |
|--|---|

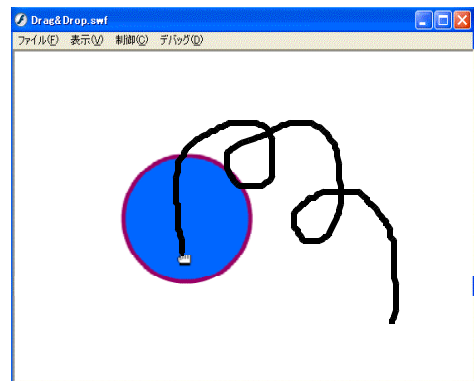
(参考) ここまでの設定で、どのような動作をするか確認してみましょう。
 [Ctrl]キーと [Enter] キーを同時に押して、ムービーのプレビューを表示させてみます。
 一度クリックすると、円のムービークリップがついて回る (もう離れません) ことが確認できます。



- (d) 次にマウスボタンから手が離れたら、その場所にムービークリップが止まるためのスクリプトを入力します。先のスクリプトに続いて、以下のスクリプトを記述します。

| | |
|--|---|
| <pre>on (release,releaseOutside) { this.stopDrag(); }</pre> | <p>※ マウスのボタンが戻ったら (release)、このインスタンスは、ドラッグ終了 (stopDrag) です。</p> |
|--|---|

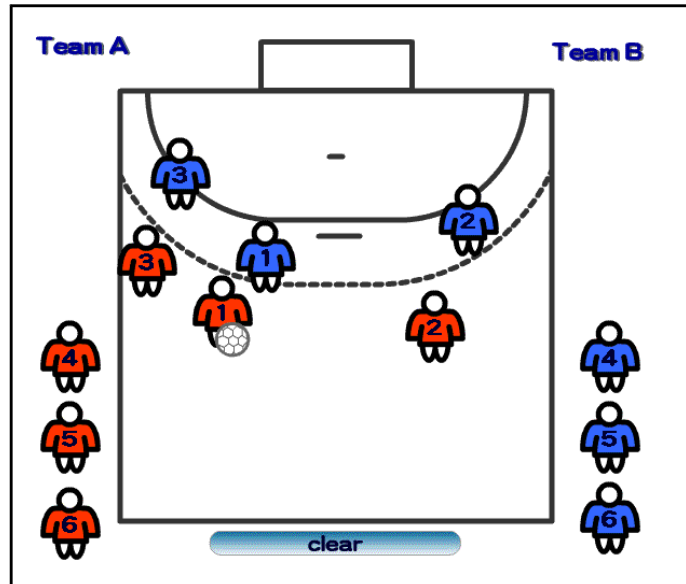
これで画面上の円に対して、ドラッグ&ドロップができるようになりました。自由に移動してみてください。



2 応用：作戦盤の作成

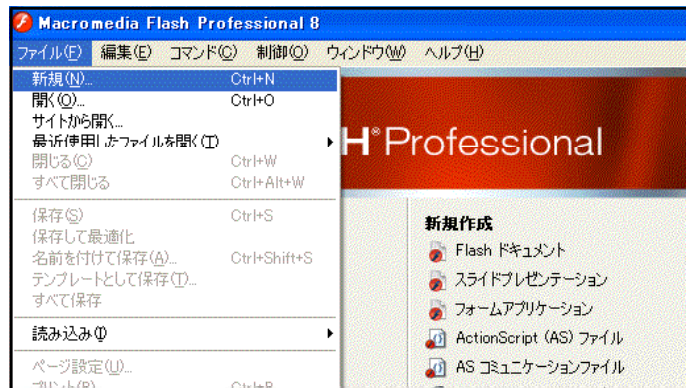
それでは、先に作成した「ドラッグ&ドロップできるムービー」を応用して、作戦盤を作成してみましょう。

手順は簡単です。背景にその競技のコードを配置して、さらに2チーム分の選手のアイコンを配置して、ドラッグ&ドロップができる **ActionScript** を適用させれば、作戦盤が完成です。

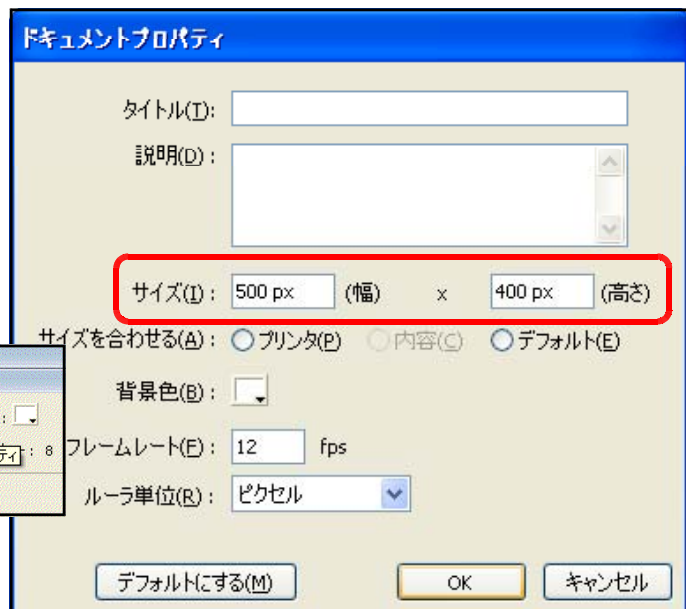


(1) 背景の作成

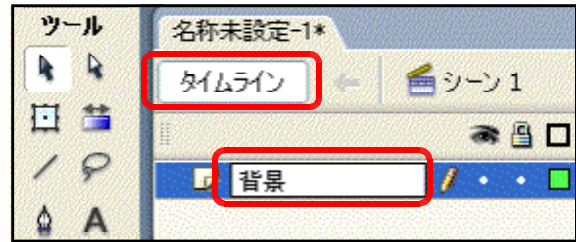
- (a) **Flash** を起動して、[ファイル] - [新規] をクリックして、新しいドキュメントを準備します。





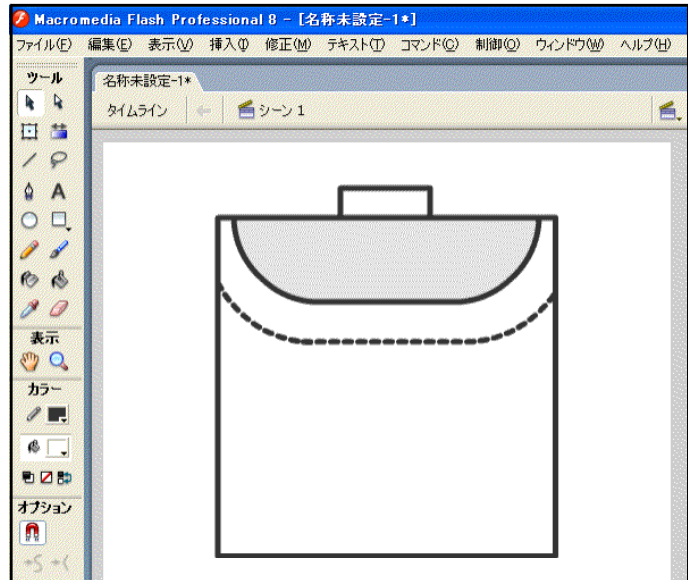
- (b) ドキュメント上を一度クリックします。画面下の [プロパティ] をクリックし、サイズボタンをクリックします。[ドキュメントプロパティ] のダイアログが表示されますので、サイズを [500px] × [400px] に設定して、[OK] ボタンをクリックします。



- (c) [タイムライン] をクリックして、レイヤー名を [背景] に変更します。

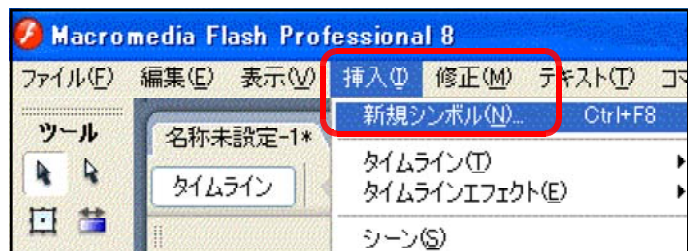


- (d) 矩形ツール  や楕円ツール  を用いて競技のコード（反面）を作成します。右図はハンドボールのコート半面です。

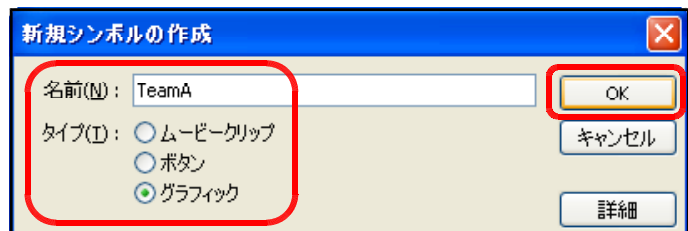



(2) 選手アイコンの作成

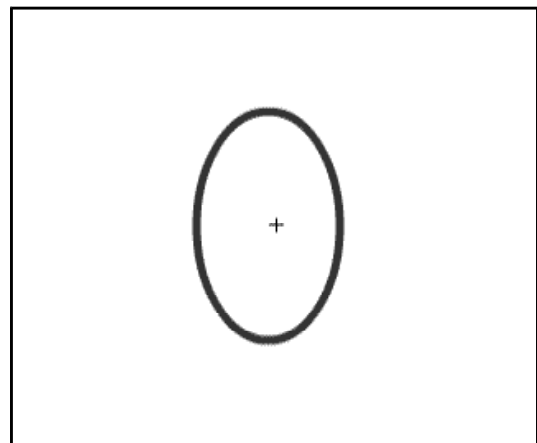
- (a) メニューから [挿入] - [新規シンボル] をクリックします。




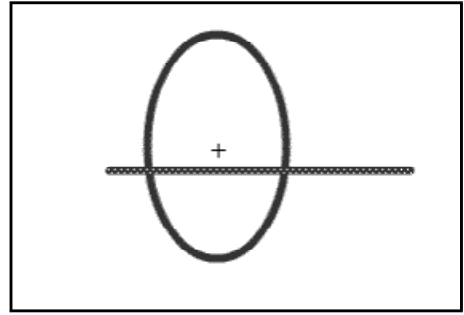
- (b) [新規シンボルの作成] ダイアログが表示されますので、タイプは [グラフィック] を選択し、名前を [TeamA] と入力して、[OK] ボタンをクリックします。



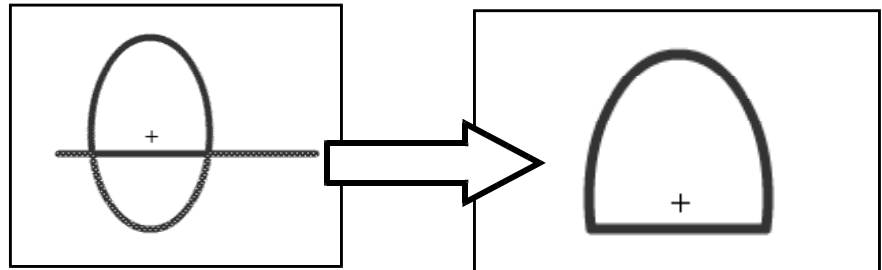
- (c) 楕円ツール  を用いて、任意のサイズで縦長の楕円を描画します。



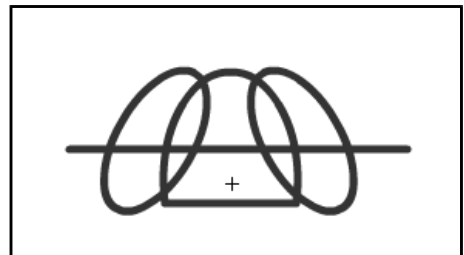
- (d) 線ツール  を用いて、楕円を横切るように線を描画します。



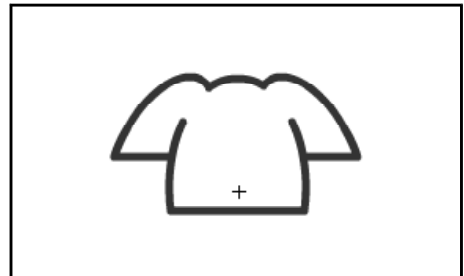
- (e) 不必要な線を [Shift] キーを押しながら、選択して、[Del] キーをクリックして、削除します。



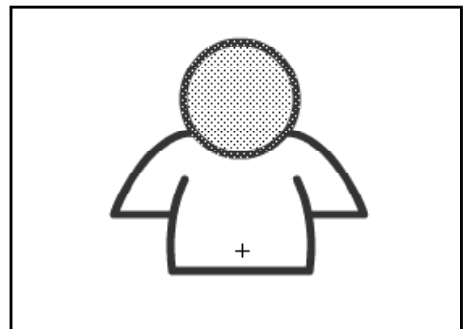
- (f) さらに袖（そで）を作成するために、楕円と直線を組み合わせて描画します。



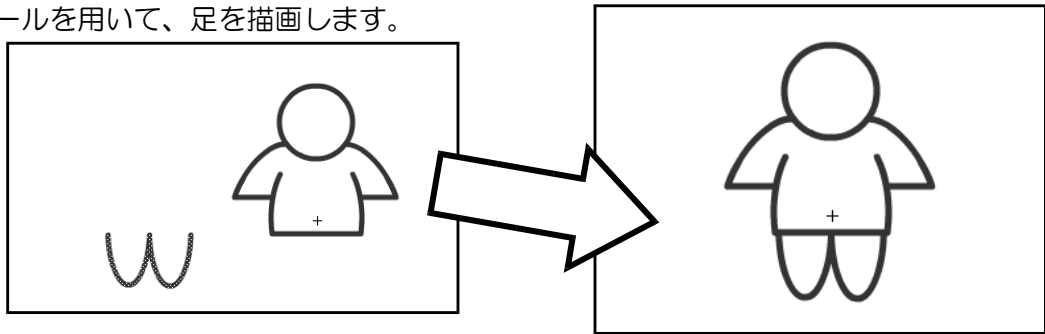
- (g) 不必要な部分を削除して、シャツを完成させます。



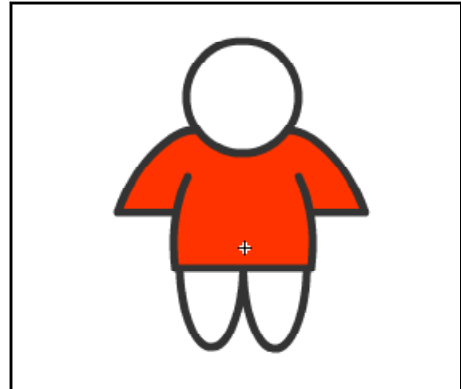
- (h) 楕円ツールを用いて、顔を描画します。



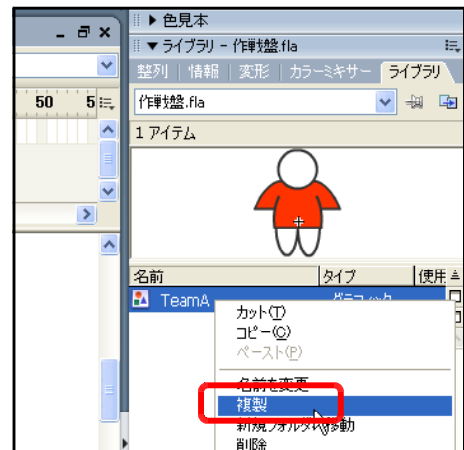
(i) 楕円ツールを用いて、足を描画します。



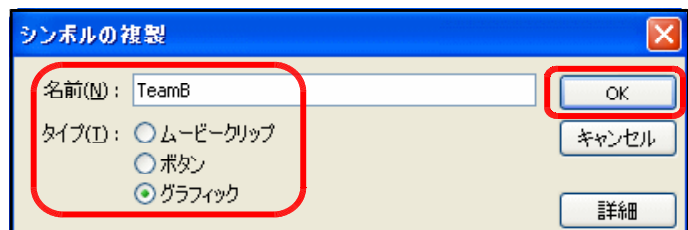
(j) バケツツールを選択して、任意の色で着色します。これで、[TeamA] というグラフィックシンボルが作成されました。



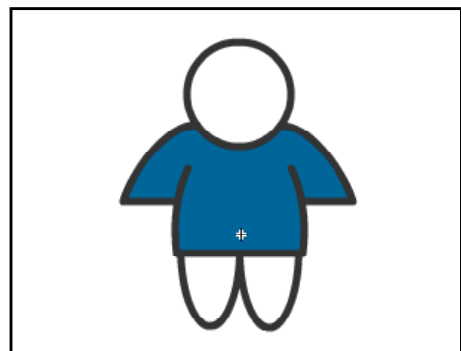
(k) 画面右のライブラリパネル [TeamA] 上で右クリックから、[複製] をクリックします。



(l) 名前を [TeamB] に変更して、[OK] ボタンをクリックします。

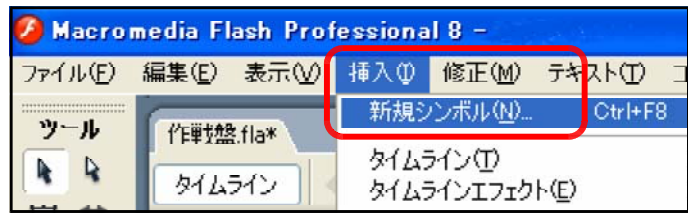


(m) ライブラリの [TeamB] をダブルクリックして選択します。バケツツールをクリックして、任意の色で着色します。これで TeamB のグラフィックシンボルが完成しました。

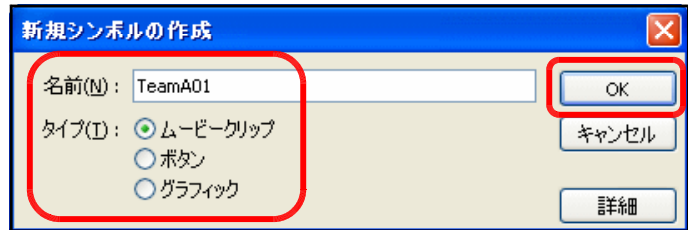


(3) 選手アイコンの作成2

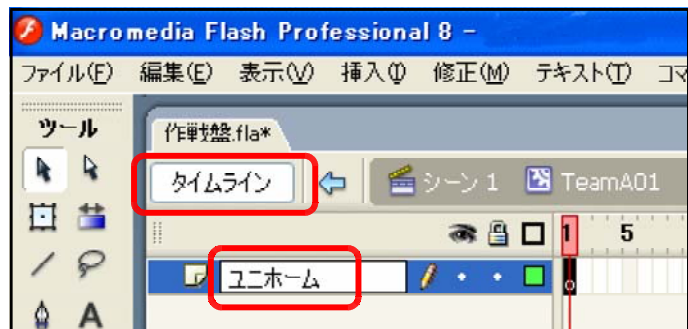
(a) メニューから [挿入] - [新規シンボル] をクリックします。



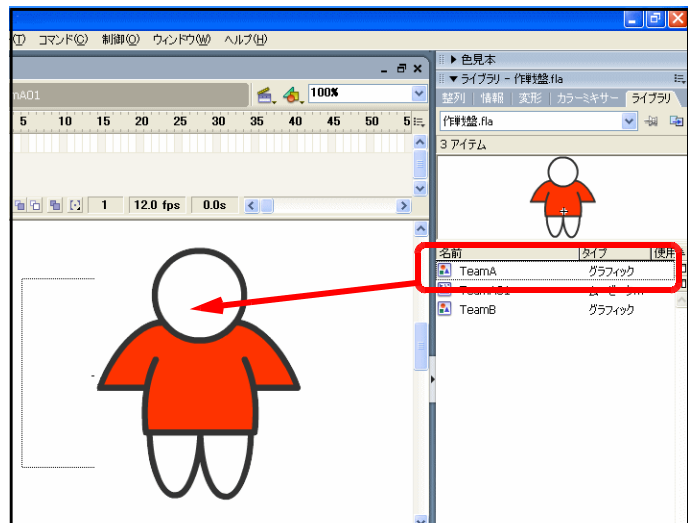
(b) タイプを [ムービークリップ]、名前を [TeamA01] に変更して、[OK] ボタンをクリックします。



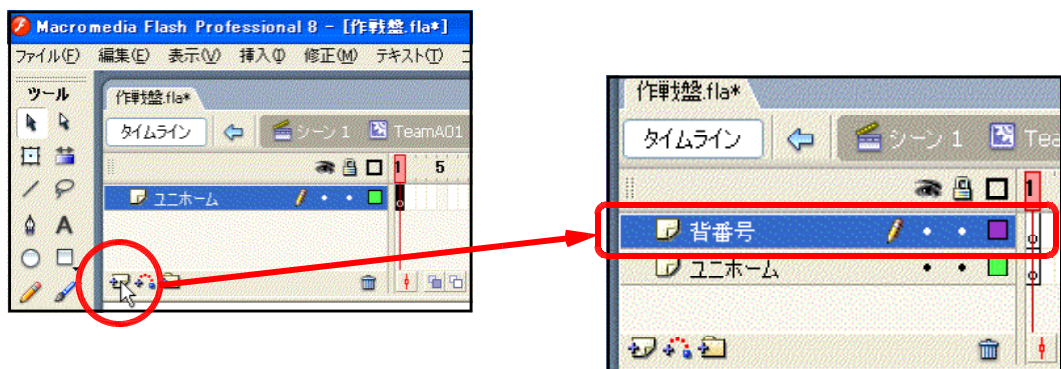
(c) [タイムライン] をクリックして、レイヤー名を [ユニホーム] に変更します。



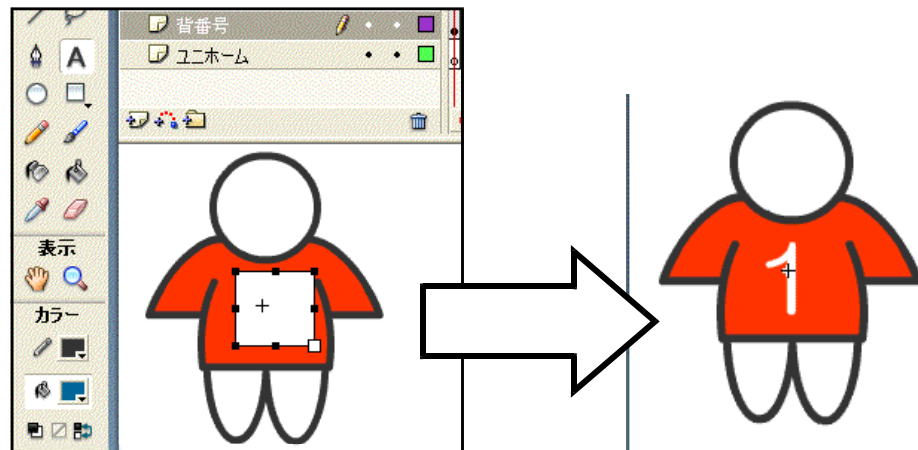
(d) ライブラリパネル内の [TeamA] グラフィックシンボルをステージ上にドラッグして配置させます。



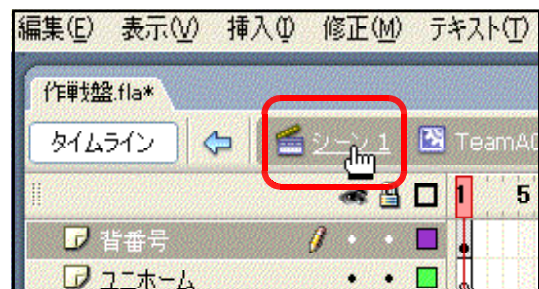
(e) [レイヤーの追加] ボタンをクリックします。レイヤー名を [背番号] に変更します。



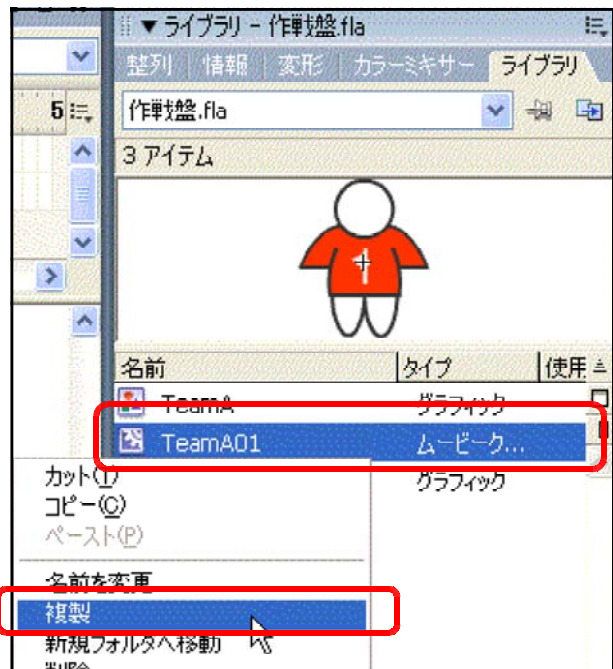
- (f) [テキストツール] をクリックして、[背番号] レイヤー上にテキストボックスを作成します。



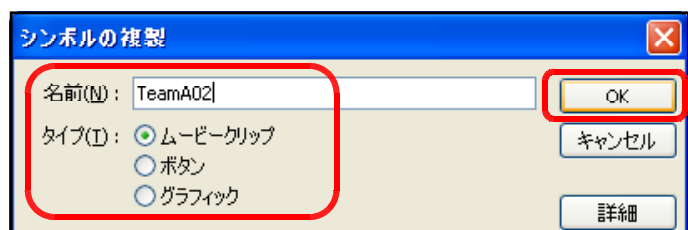
- (g) [シーン1] をクリックして、ドキュメントに戻ります。ライブラリには [TeamA01] ムービークリップシンボルが追加されていることを確認します。



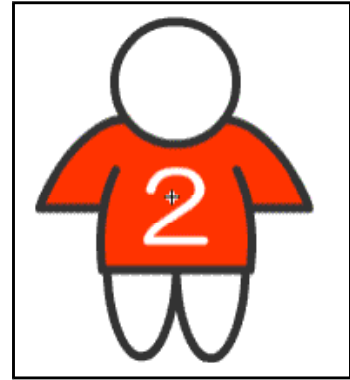
- (h) ライブラリパネル内の [TeamA01] をクリックから、右クリック [複製] をクリックします。



- (i) 名前を [TeamA02] に変更して、[OK] ボタンをクリックします。



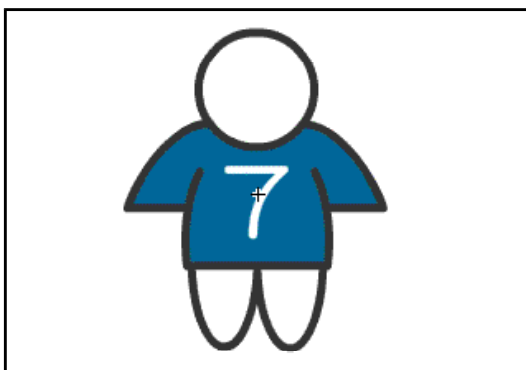
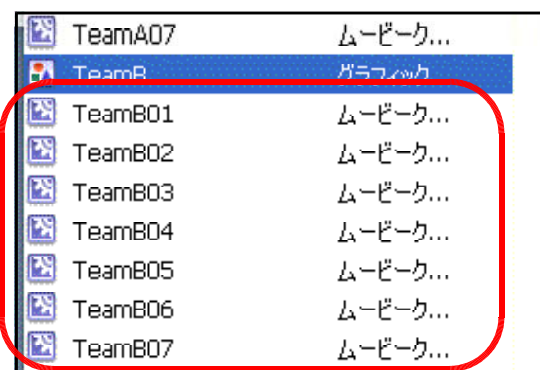
- (j) テキストツールを選択して、背番号1の番号を [2] に変更します。



- (k) 同様に上記の(h)~(j)を繰り返して、背番号3~7まで合計7個の選手アイコンを作成します。ただし、ムービークリップ名は、[TeamA01] ~ [TeamA07] とする。



- (l) チームBの選手アイコンの作成を上記操作(a)~(k)を繰り返して作成します。ただし、ムービークリップ名を [TeamB01] ~ [TeamB07] とします。

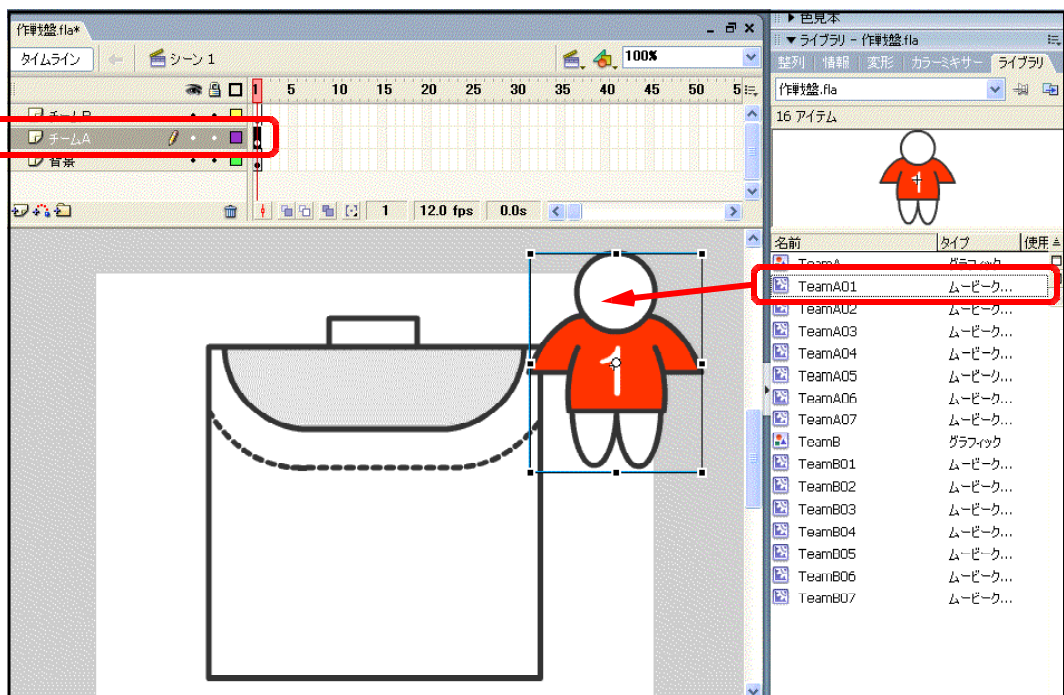



(4) 選手アイコンの配置

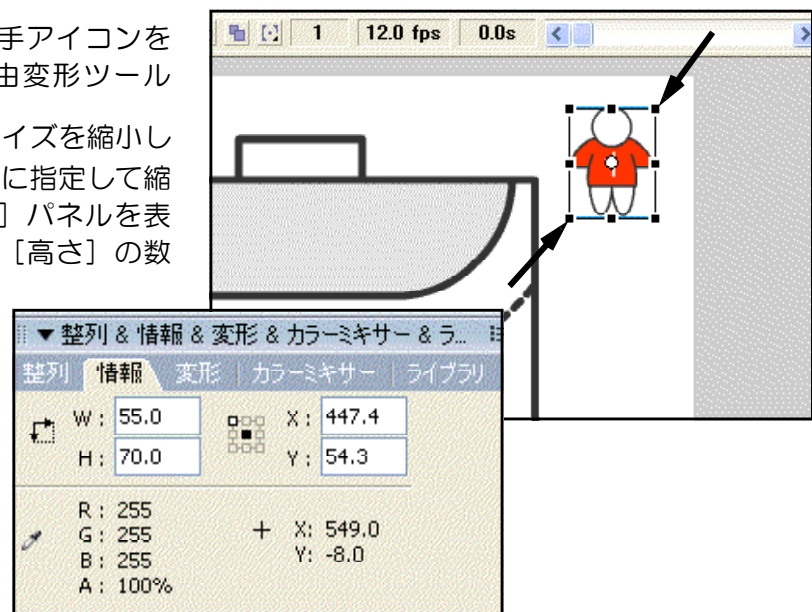
- (a) [レイヤーの追加] ボタンをクリックして、[背景] レイヤー上に [チームA] と [チームB] レイヤーを配置します。



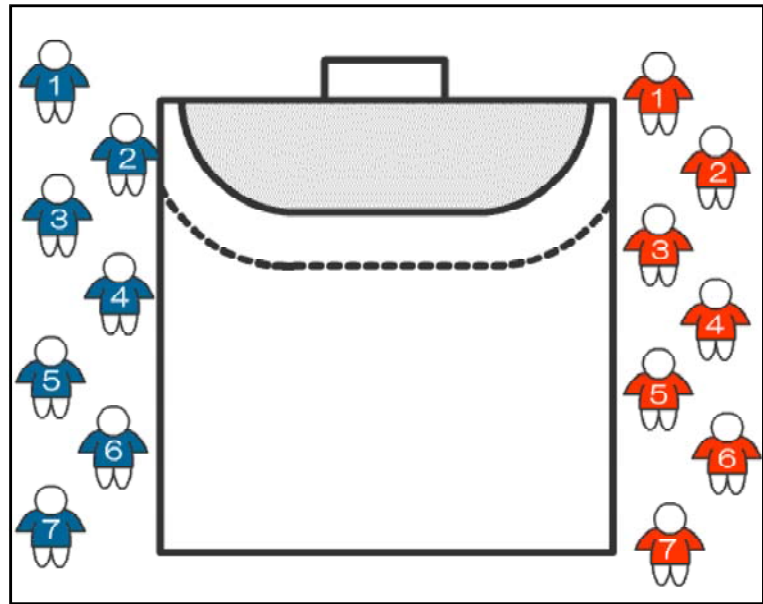
- (b) [チームA] レイヤーをクリックして、ライブラリパネル内から [TeamA01] ムービークリップシンボルをステージ上にドラッグします。



なお、ステージ上に選手アイコンを重ねて大きい場合は、自由変形ツール  を選択して、適宜、サイズを縮小します。また、サイズを正確に指定して縮小したい場合には、[情報] パネルを表示させて、**W** [幅] と **H** [高さ] の数値で指定します。

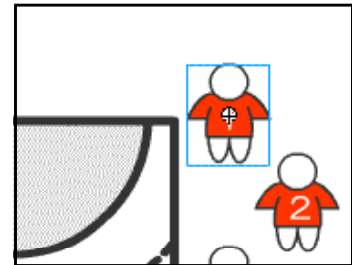


- (c) (b)の操作をくり返して、ステージ上に各チームのアイコンを整列させます。

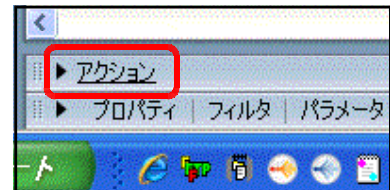


(5) ActionScript の設定

- (a) ステージ上のチームAの背番号1の選手アイコンをクリックして、選択します。



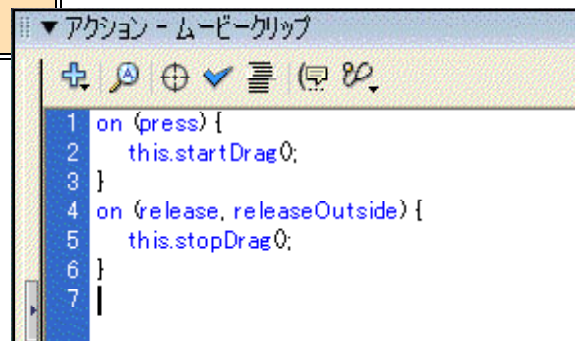
- (b) [アクション] をクリックして、アクションパネルを表示させます。



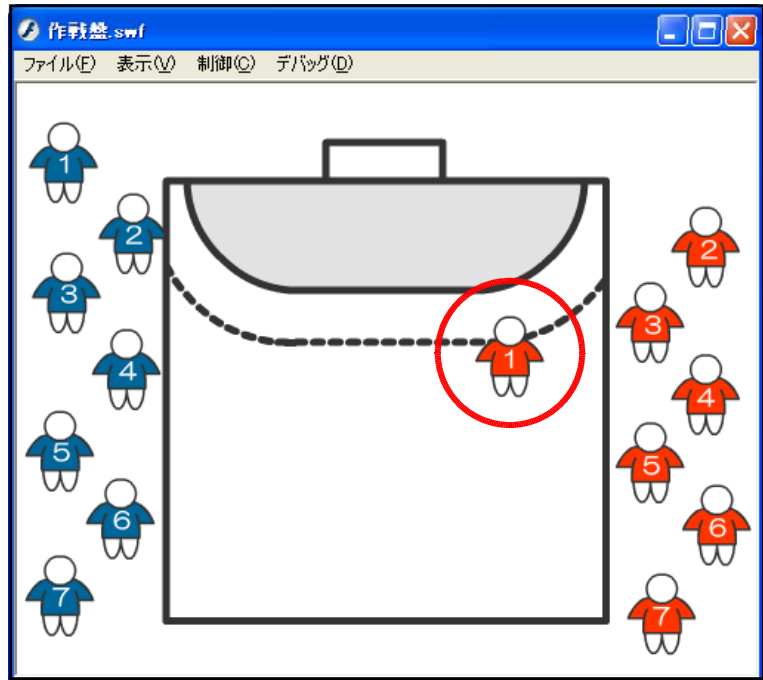
- (c) アクションパネル内に以下のスクリプトを入力します。

```

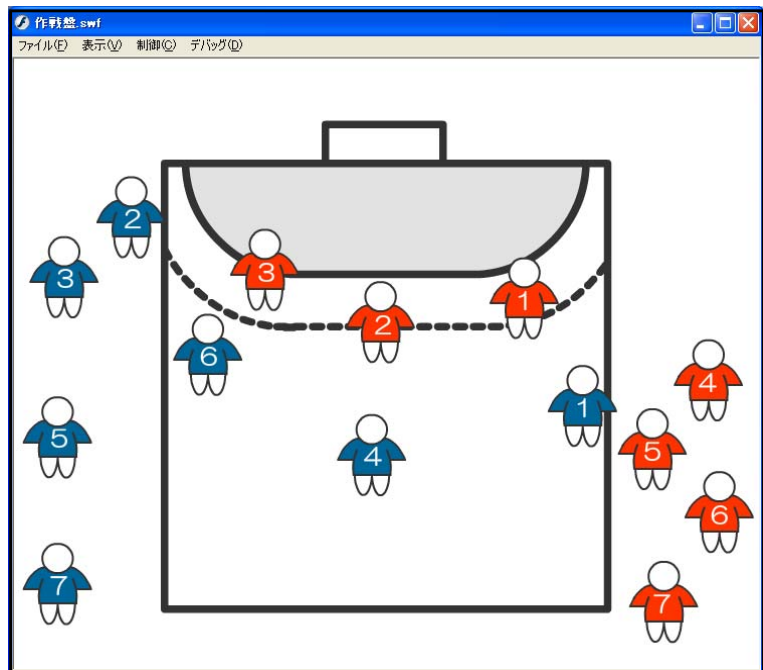
on (press) {
    this.startDrag();
}
on (release,releaseOutside) {
    this.stopDrag();
}
    
```



- (d) [Ctrl] キーと [Enter] キーを同時に押して、ムービーのプレビューを作成します。チームAの1番の選手アイコンをドラッグして移動させてみます。



- (e) 他の選手アイコンについても、(a)～(c)の操作をくり返して、**ActionScript** を設定します。これで作戦盤は完成です。各選手をドラッグして、フォーメーションの確認やディフェンス形態の確認ができます。



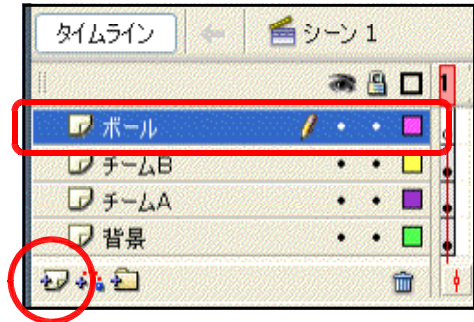
<memo>

作成した作戦盤を使ってみると、さまざま改善点が思い浮かびませんか？最低限の動きに加えて、バージョンアップしてみましょう。

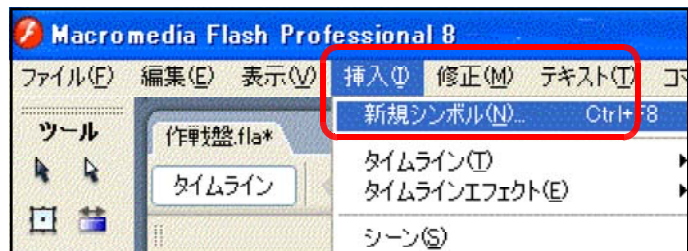
【追加機能1】 ～ ボールがあればいい！！

もちろんドラッグ&ドロップができるように、ボールのアイコンも同様に追加してみましょう。

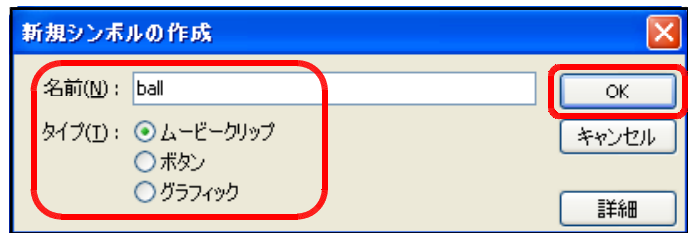
- (a) [レイヤーの追加] ボタンをクリックして、[チームB] レイヤー上に [ボール] レイヤーを作成します。



- (b) メニューから [挿入] - [新規シンボル] をクリックします。



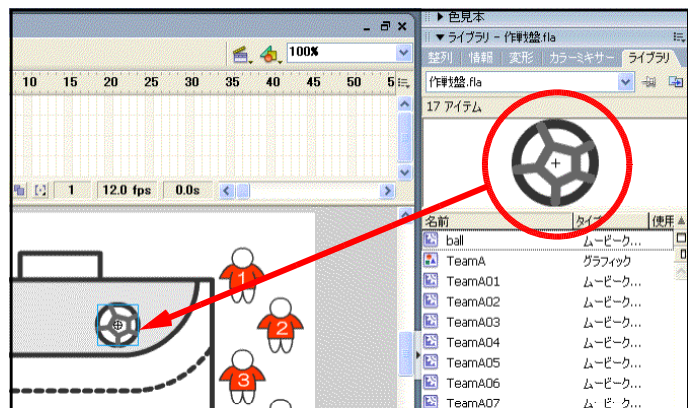
- (c) タイプを [ムービークリップ]、名前を [ball] として、[OK] ボタンをクリックします。



- (d) 楕円ツールと線ツールを用いて、ボールを描画します。その後、[シーン1] をクリックして、ドキュメントに戻ります。



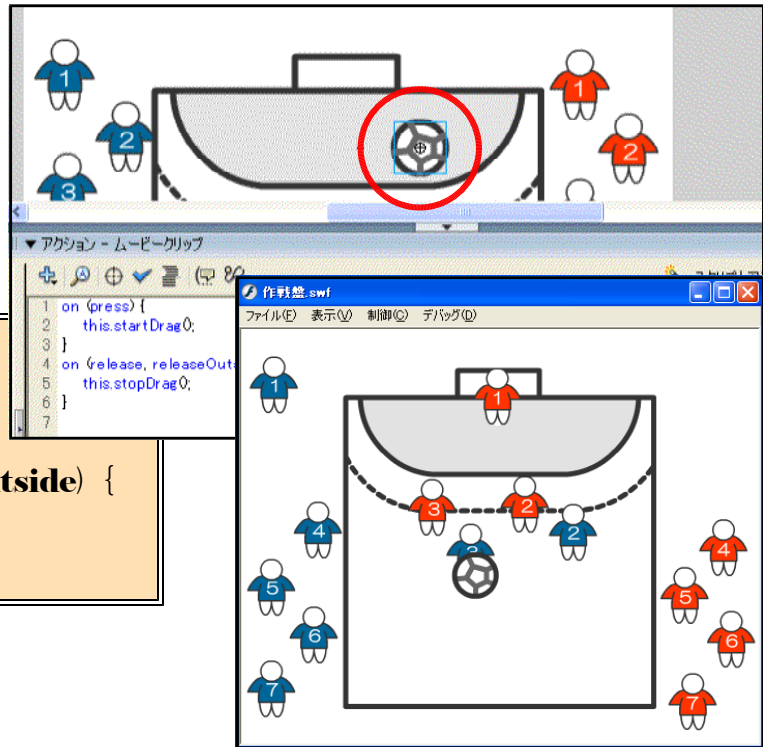
- (e) ライブラリから [ball] ムービークリップをドラッグして、ステージ上に配置します。サイズが大きい時には、変形ツールをクリックして、適宜、サイズを変更します。



- (f) ボールをクリックして、アクションパネルに以下の **ActionScript** を入力します。これで、ボールの動きも加えて、作戦盤を利用することができます。

```

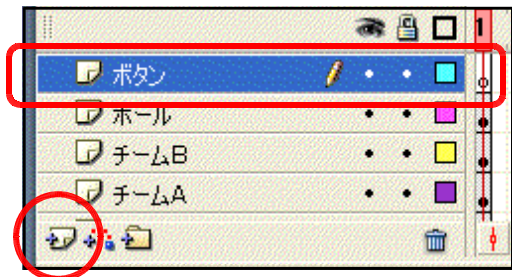
on (press) {
    this.startDrag();
}
on (release,releaseOutside) {
    this.stopDrag();
}
    
```



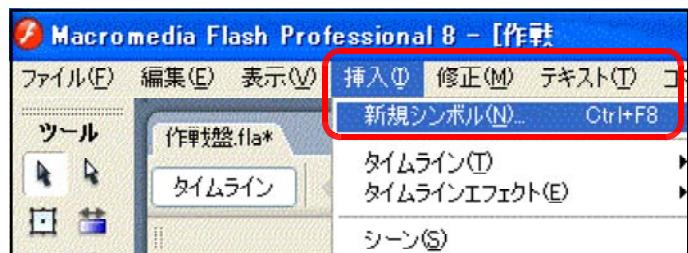
【追加機能2】 ～ 選手アイコンがワンクリックで整列できればいい！！

作戦盤を用いてフォーメーションの確認後、アイコンを整列させるためのボタンを追加してみましょう。

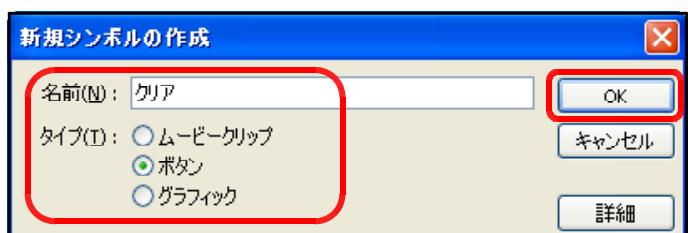
- (a) [レイヤーの追加] ボタンをクリックして、[ボール] レイヤー上に [ボタン] レイヤーを作成します。




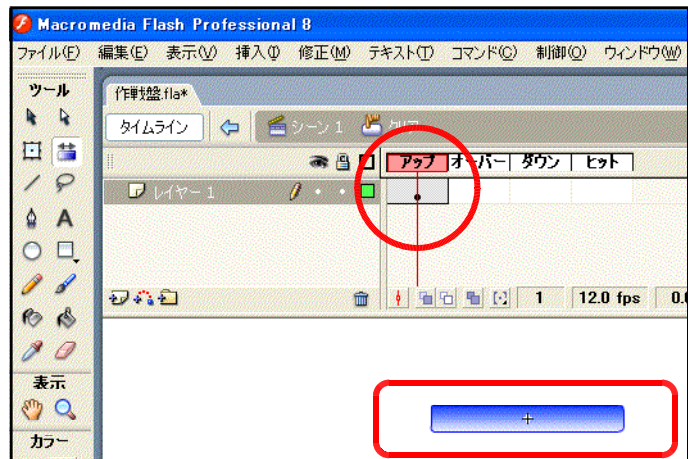
- (b) メニューから [挿入] - [新規シンボル] をクリックします。



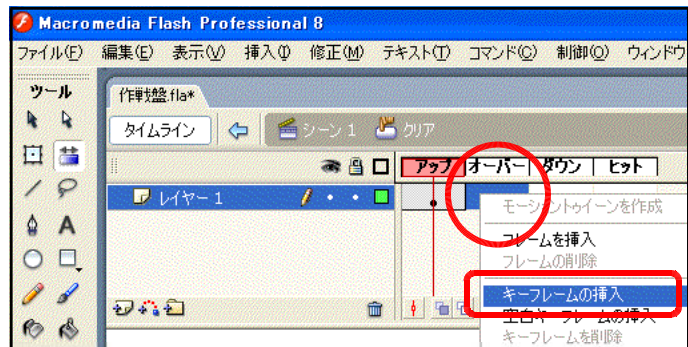
- (c) タイプを [ボタン]、名前を [クリア] として、[OK] ボタンをクリックします。



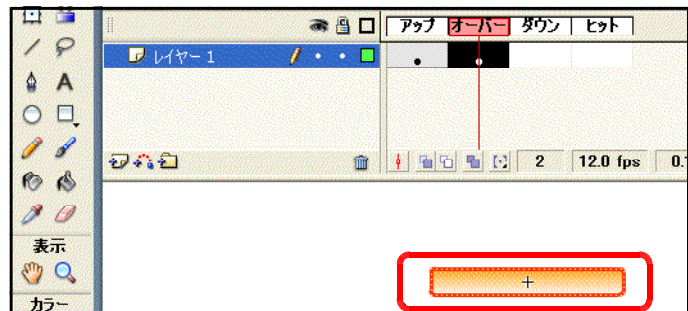
- (d) [レイヤー1] の [アップ] をクリックして、矩形ツールを用いて、ボタンを作成します。適宜、線のカラーや塗りのカラーを変更します。



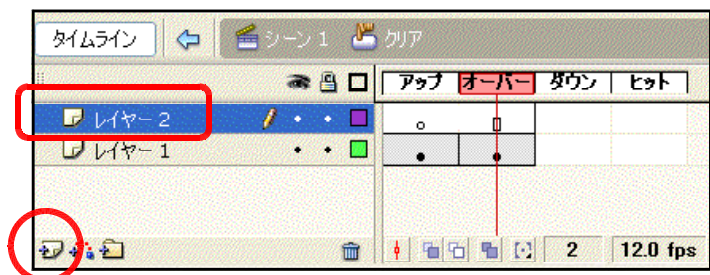
- (e) [レイヤー1] の [オーバー] 上で右クリックから、[キーフレームの挿入] をクリックします。




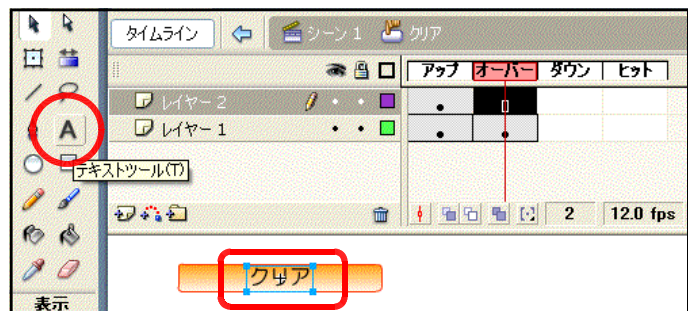
- (f) 適宜、線のカラーや塗りのカラーを変更します。これで、通常時のボタンの配色とマウスのカーソルが重なった時で色が変化ようになります。



- (g) [レイヤーの追加] ボタンをクリックして、[レイヤー1] 上に [レイヤー2] を作成します。



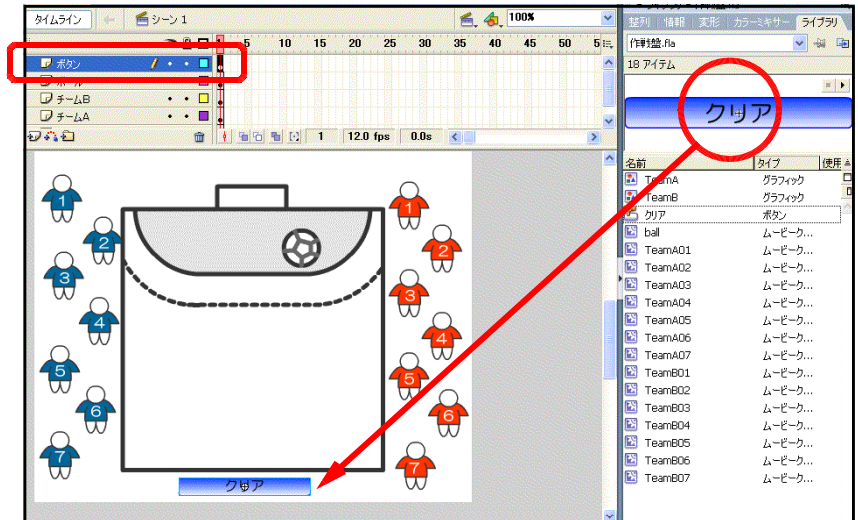
- (h) [テキストツール] を用いて、ボタン上に [クリア] の文字を入力します。



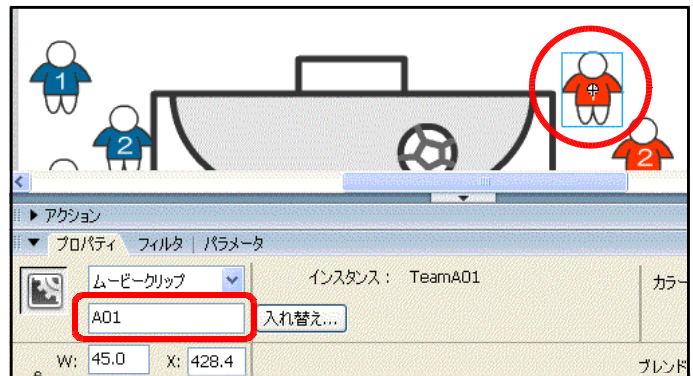
- (i) [シーン1] をクリックして、ドキュメントに戻ります。



- (j) [ボタン] レイヤーをクリックして、ライブラリパネルから [ボタン] シンボルをステージ上にドラッグして配置させます。

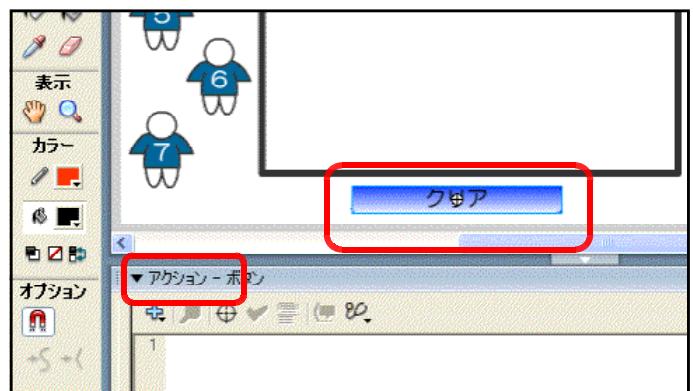


- (k) チームAの背番号1の選手アイコンをクリックして選択します。プロパティインスペクタで名前を [A01] と入力します。これで、チームAの背番号1のアイコンの名前を **A01** と付けたことになります。



- (l) 同様にチームAの背番号2を[A02]、・・・背番号7を [A07]、チームBの背番号1を [B01]、・・・、背番号7を [B07]、ボールを [ball] というように、それぞれ名前を付けます。

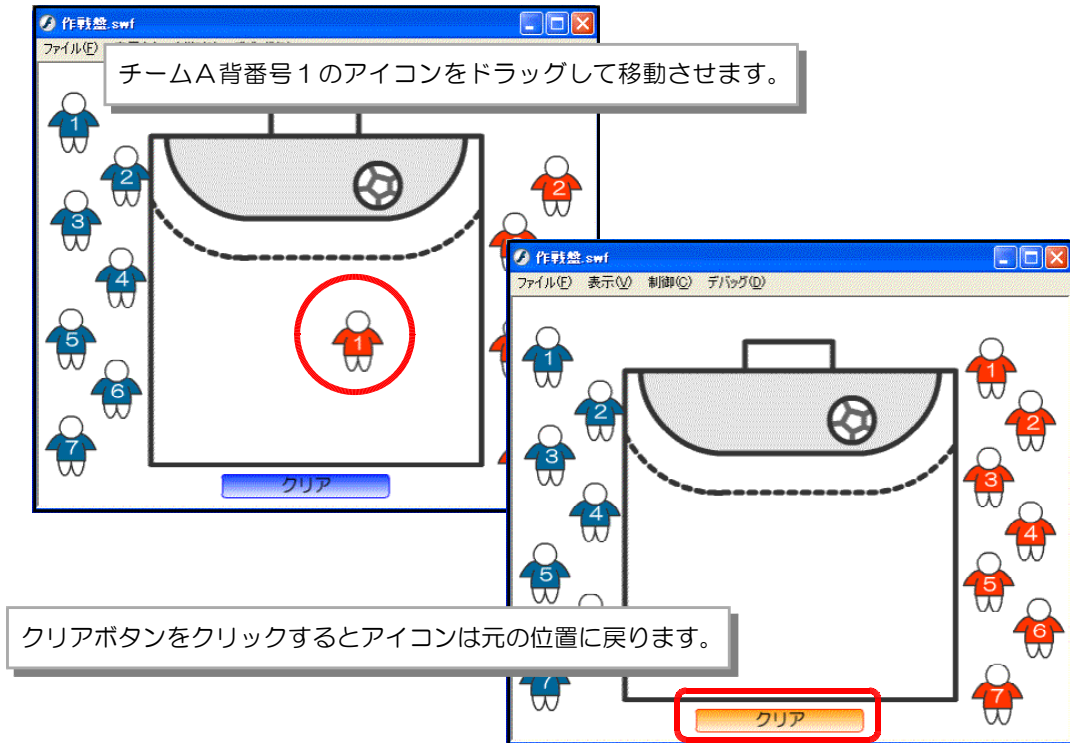
- (m) ステージ上のボタンをクリックして選択し、[アクション] をクリックして、アクションパネルを表示させます。



(n) 以下のスクリプトを入力します。

| | |
|--|--|
| <pre>on (press) { _root.A01._x = 430; _root.A02._y = 65; }</pre> | <p>※ ボタンが押されらならば、 インスタンス a01 の X 座標は 630 へ、 Y 座標は 65 へ 移動させる。</p> |
|--|--|

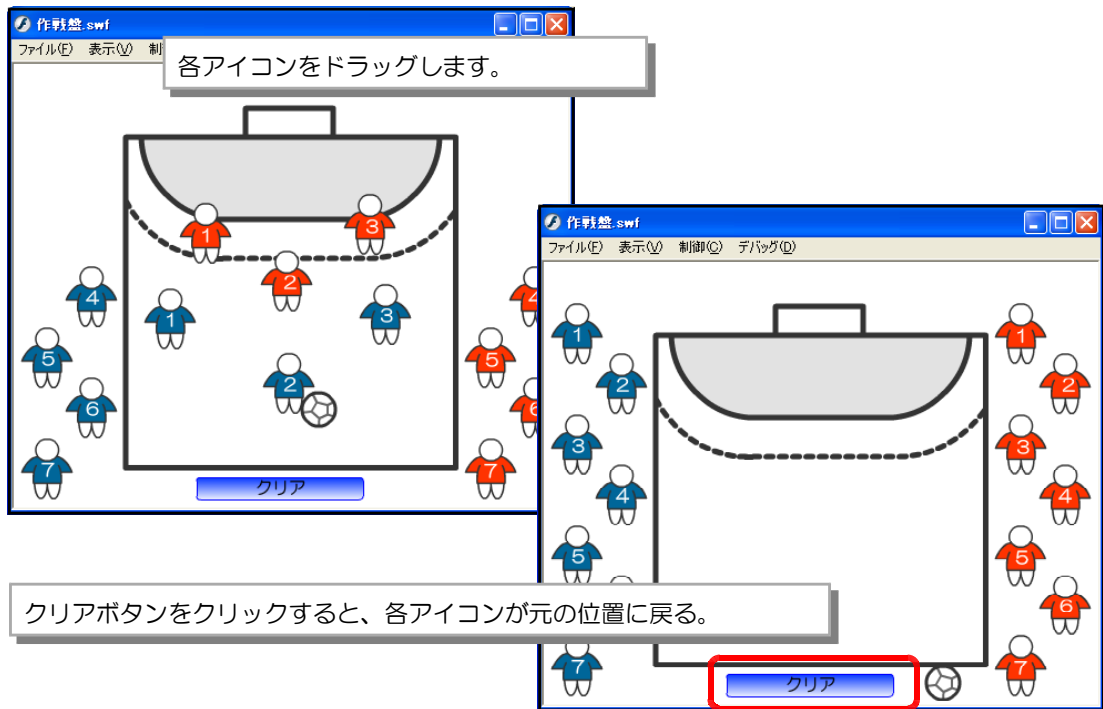
(o) プレビューで動作を確認してみましょう。[Ctrl] キーと [Enter] キーを同時に押し、プレビューさせます。



(p) 他のアイコンについても、元の位置に戻るためのスクリプトを以下のようにボタンへ追加します。

| | | |
|--|--|--|
| <pre>on (press) { _root.A01._x = 430; _root.A01._y = 65; _root.A02._x = 470; _root.A02._y = 110; _root.A03._x = 430; _root.A03._y = 165; _root.A04._x = 470; _root.A04._y = 210; _root.A05._x = 430; _root.A05._y = 265; _root.A06._x = 470; _root.A06._y = 310; _root.A07._x = 430;</pre> | <pre>_root.A07._y = 365; _root.B01._x = 30; _root.B01._y = 65; _root.B02._x = 70; _root.B02._y = 110; _root.B03._x = 30; _root.B03._y = 165; _root.B04._x = 70; _root.B04._y = 210; _root.B05._x = 30; _root.B05._y = 265; _root.B06._x = 70; _root.B06._y = 310; _root.B07._x = 30;</pre> | <pre>_root.B07._y = 365; _root.ball._x = 360; _root.ball._y = 380; }</pre> |
|--|--|--|

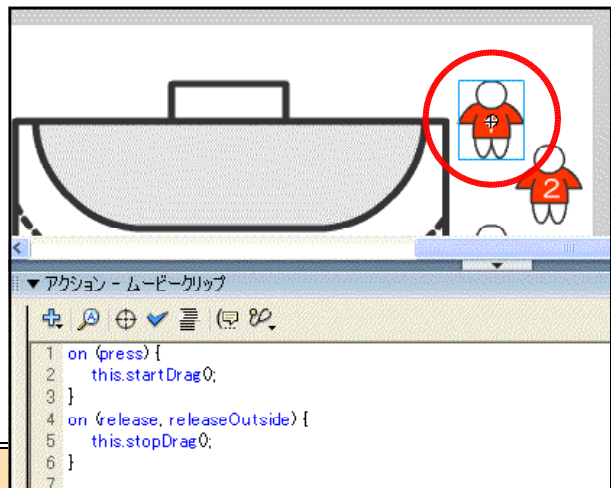
- (q) 以上でスクリプトの設定が完了しましたので、動作確認をしてみます。[Ctrl] キーと [Enter] キーを同時に押して、プレビューさせます。



【追加機能3】 ~ アイコンの重ね順を変えれば！！

基本的に各インスタンスは後から作成したものが前面に配置されます。ここでは意図的に重ね順を変えてみましょう。具体的にはクリックしたものが最前面に配置されるようにしてみます。

- (a) チームAの背番号1のアイコンをクリックして選択します。



- (b) すでにスクリプトが設定されていますが、**on(press) { ~ }**内に以下のスクリプトを入力します。

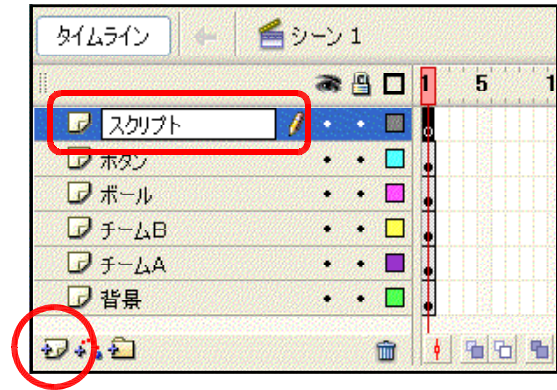
```

on (press) {
  this.swapDepths(_root.depth++); ← 追加部分
  this.startDrag();
}

```

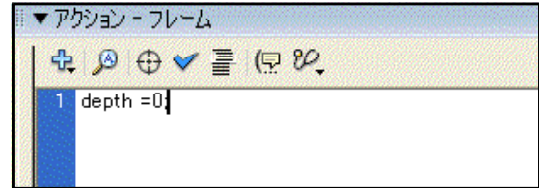
- (c) 同様に他のアイコンについても、上記のスクリプトを追加します。

- (d) [レイヤーの追加] ボタンをクリックして、
[ボタン] レイヤーの上に [スクリプト] レイヤーを追加します。

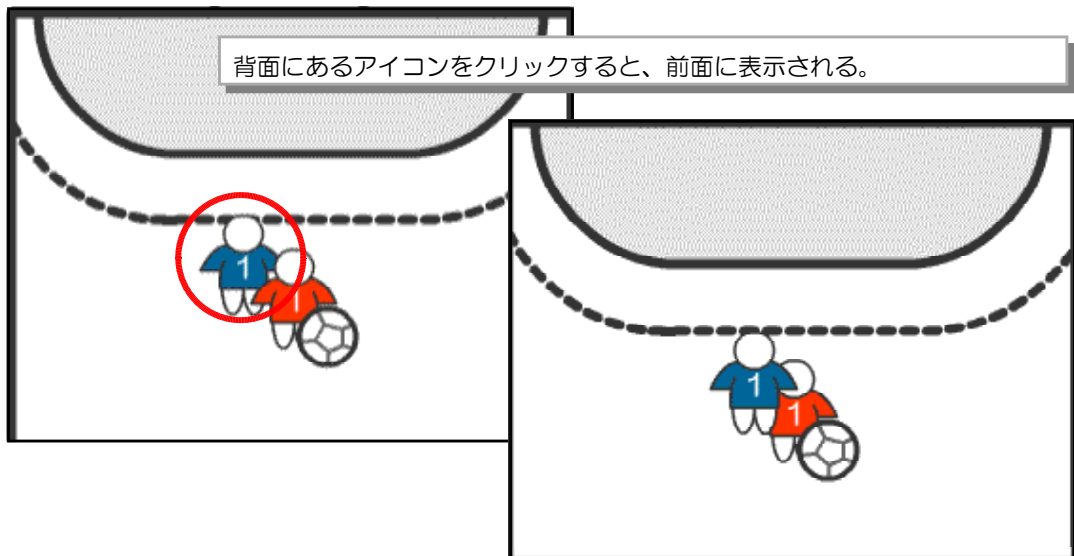


- (e) [アクション] パネルを表示させ、以下のスクリプトを入力します。

```
depth = 0;
```



- (f) 以上で設定終了です。プレビューで動作確認をしてみましょう。[Ctrl] キーと [Enter] キーを同時に押します。



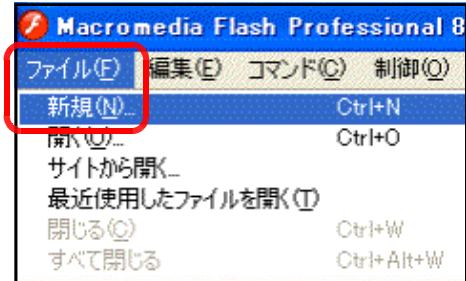
<memo>

第3章 外部ファイル読込の活用

1 外部ファイルからデータを 読み込むムービーの作成

(Ⅰ) テキストフィールドの作成

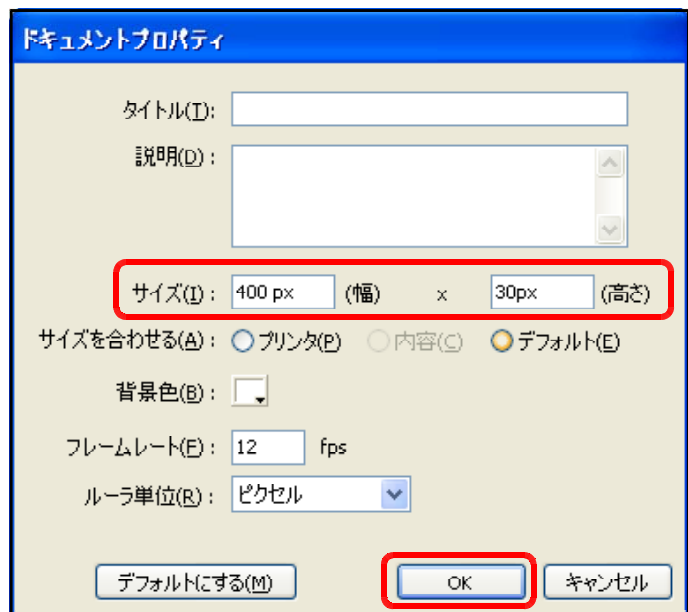
(a) **Flash8** を起動し、メニューから [ファイル] - [新規] をクリックして、新規ドキュメントを作成します。



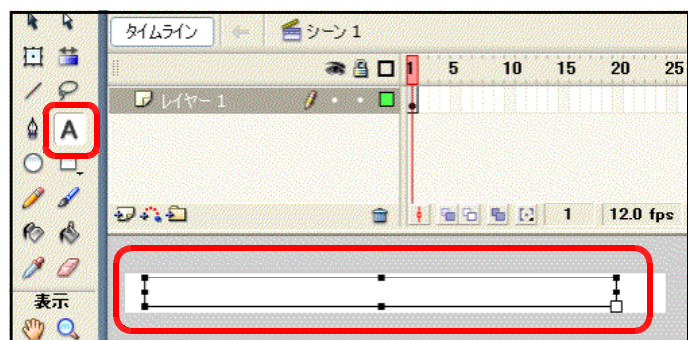
(b) [プロパティ] をクリックして、プロパティパネルを表示させます。ドキュメントのサイズを変更するために [550 × 400 ピクセル] ボタンをクリックします。



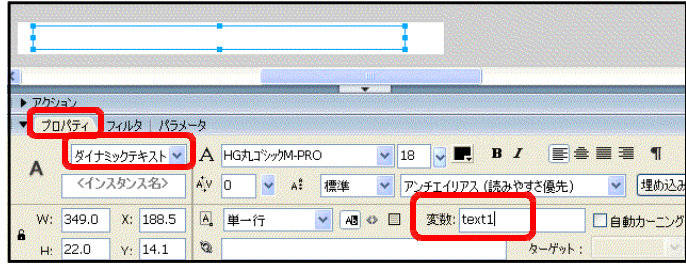
(c) [ドキュメントプロパティ] ダイアログが表示されますので、サイズを [400px] (幅) × [30px] (高さ) に変更して [OK] ボタンをクリックします。



(d) [テキストツール] **A** をクリックして、ドキュメント上にドラッグして、任意のサイズでテキストフィールドを作成します。

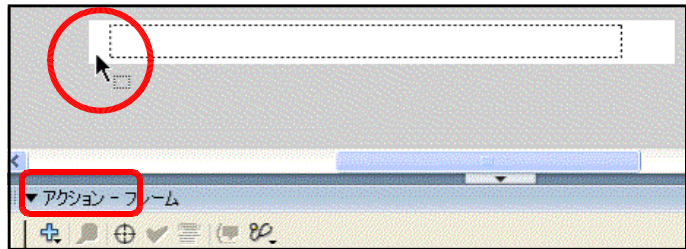


- (e) [プロパティ] をクリックしてプロパティパネルを表示させます。[ダイナミックテキスト] を選択して、変数を [text1] と入力して設定します。

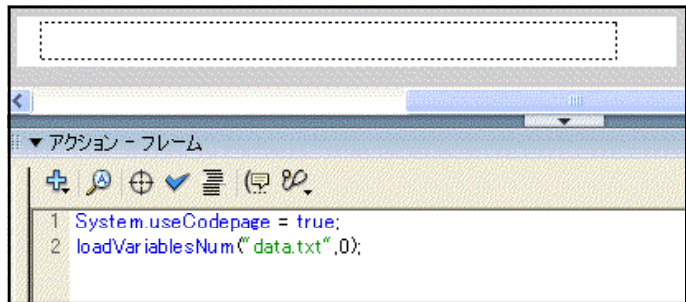


(2) ActionScript の設定

- (a) ドキュメントをクリックしてから [アクション] をクリックし、アクションパネルを表示させます。

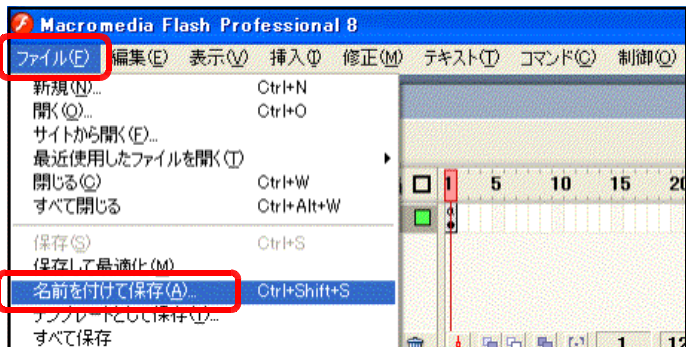


- (b) 外部ファイルを読み込むためのスクリプトを入力します。

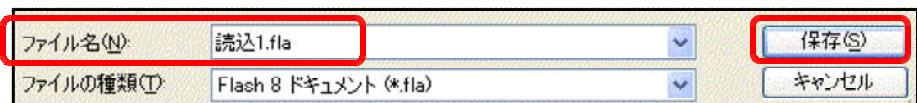


| | |
|--|----------------------|
| System.useCodepage = true; | 文字コードを Shift-JIS にする |
| loadVariablesNum("data.txt",0); | data.txt から変数を読み込む |

- (c) メニューから [ファイル] - [名前を付けて保存] をクリックします。



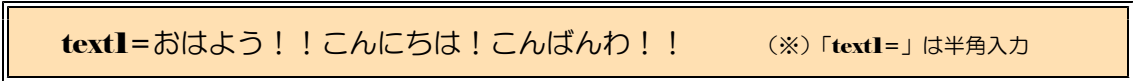
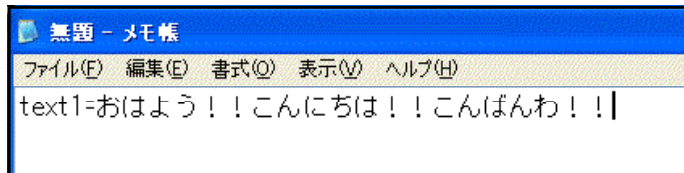
- (d) ファイル名を [読込1.fla] と入力し、保存するディレクトリを指定して、[保存] ボタンをクリックします。



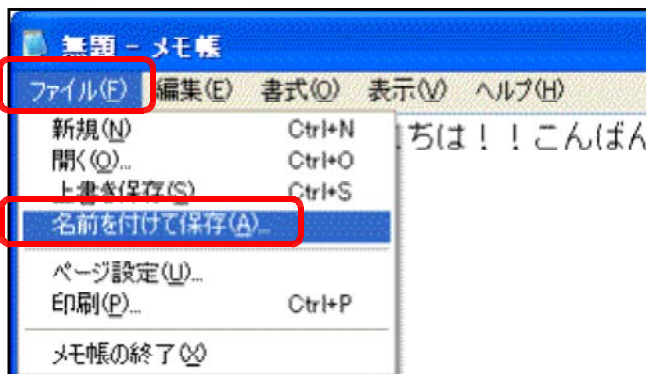
(3) 外部ファイルの作成

(a) デスクトップの [スタート] から [すべてのプログラム] - [アクセサリ] - [メモ帳] をクリックします。

(b) メモ帳に以下の文字列を入力します。



(c) メニューから [ファイル] - [名前を付けて保存] をクリックします。

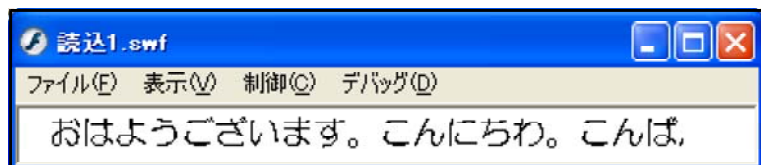


(d) ファイル名を [data.txt] として、保存ディレクトリを設定したら、[保存] ボタンをクリックして、データファイルを保存します。ただし、この data.txt ファイルは、先に作成した [読込1.flc] と同じ階層のディレクトリに保存してください。

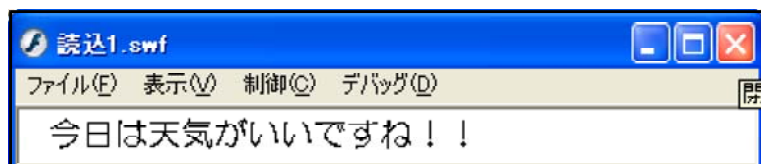


(4) 動作確認

(a) [読込1.flc] の編集画面から、[Ctrl] キーと [Enter] キーを同時に押して、プレビュー画面を表示させます。文字列が表示されるかを確認します。



テキストファイル (data.txt) の内容を変更して、再度プレビューすると反映されます。



2 応用：「リンク集」の作成

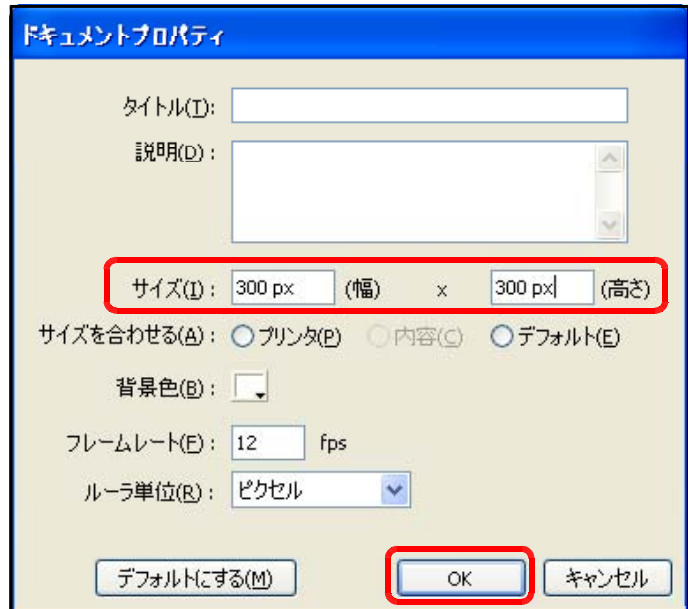
(1) ドキュメントのサイズ変更

(a) [プロパティ] をクリックして、[プロパティ] パネルを表示させます。



(b) サイズの [550px × 400px] ボタンをクリックします。

(c) サイズを [300px] (幅)、[300px] (高さ) に設定して、[OK] ボタンをクリックして、ドキュメントサイズを変更します。



(2) ボタンシンボルの作成

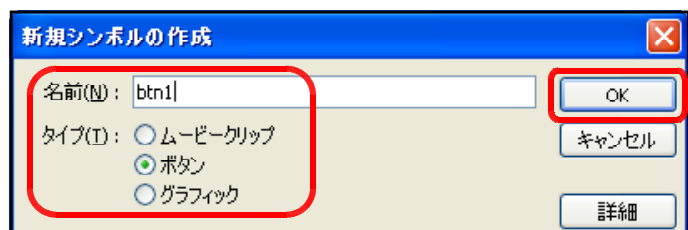
(a) **Flash8** を起動して、メニューから [ファイル] - [新規] をクリックして、新規のドキュメントを開きます。



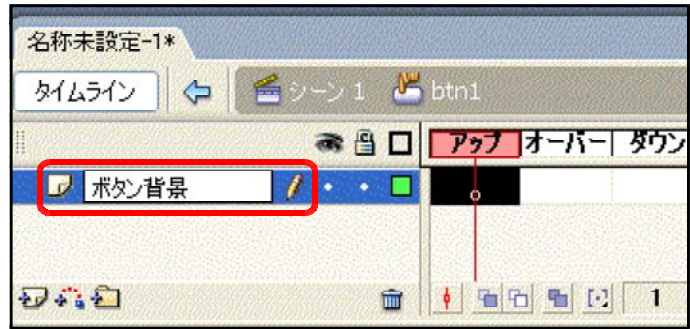
(b) メニューから [挿入] - [新規シンボル] をクリックします。




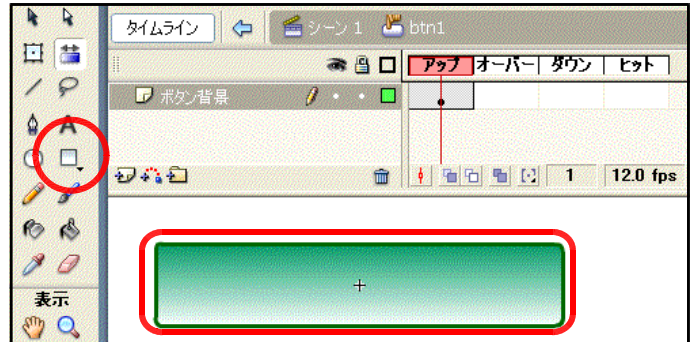
(c) タイプを [ボタン]、名前を [btn1] として、[OK] ボタンをクリックします。



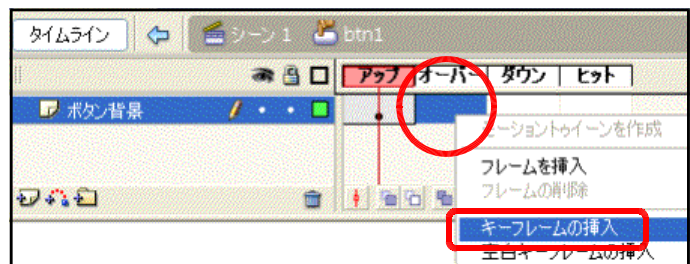
- (d) レイヤー名 [レイヤー1] をダブルクリックして、名称を [ボタン背景] に変更します。



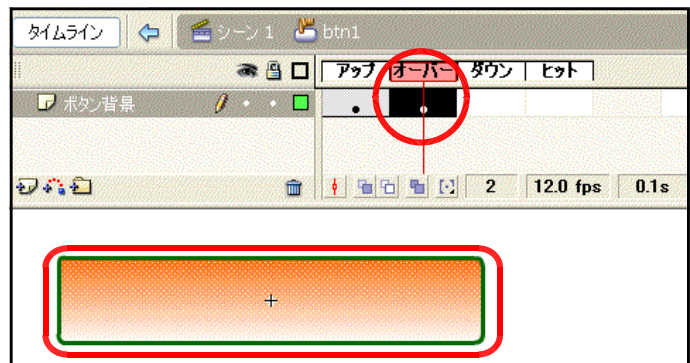
- (e) 矩形ツール  を用いて、長方形を描画します。適宜、線のカラーと塗りのカラーを設定します。なお、長方形のサイズは、[300px] (幅) × [60px] とします。




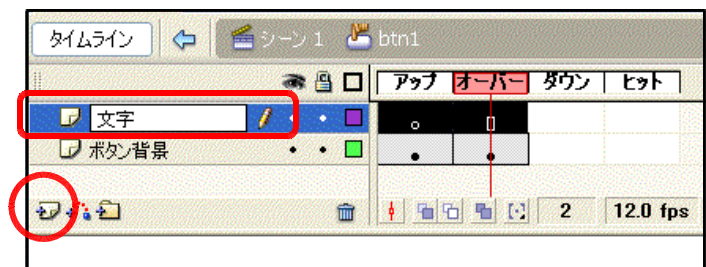
- (f) [ボタン背景] レイヤーの [オーバー] 上で右クリックから、[キーフレームの挿入] をクリックします。




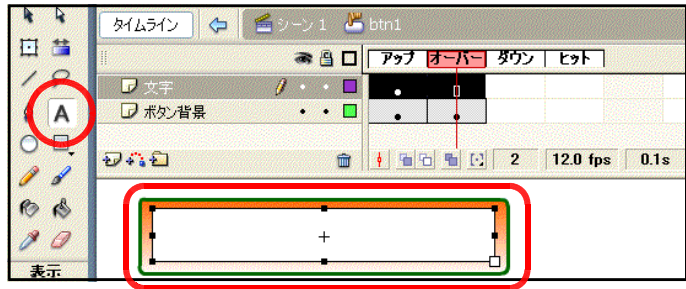
- (g) [ボタン背景] レイヤーの [オーバー] 上でクリックし、適宜、線のカラーや塗りのカラーを変更します。このような設定をすることで、このボタン上にマウスのカーソルが重なった時に、色が変わるアクションが可能になります。



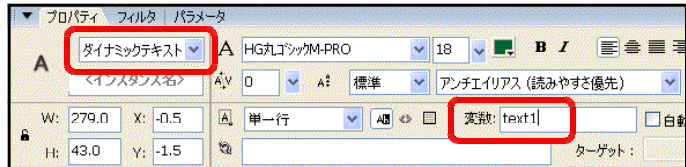
- (h) [レイヤーの追加]  ボタンをクリックして、[文字] レイヤーを追加します。



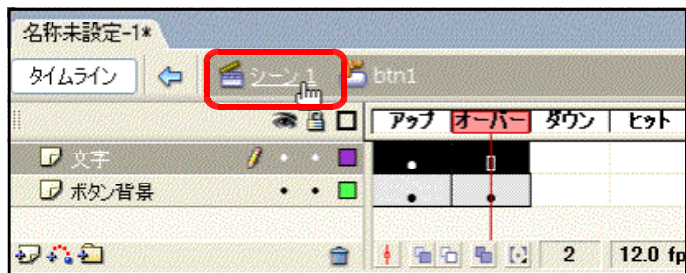
- (i) テキストツール  を用いて、先に描画した長方形の上に、テキストフィールドを配置します。



- (j) [プロパティ] パネルで、[ダイナミックテキスト] を選択し、変数を [text1] と設定します。



- (k) [シーン1] をクリックして、ドキュメントに戻ります。これで、ボタンのシンボルが完成しました。

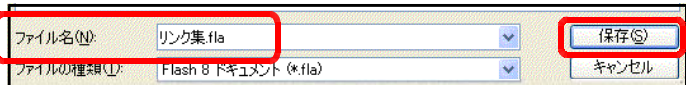


(3) ファイルの保存

- (a) メニューから [ファイル] - [名前を付けて保存] をクリックします。

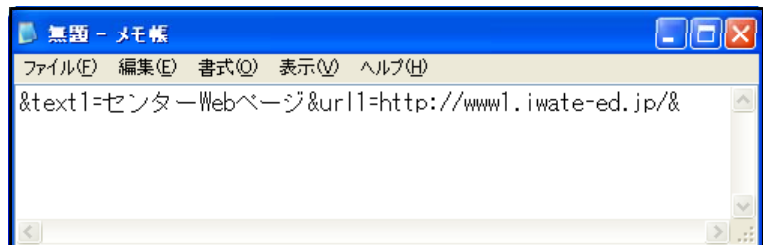


- (b) 適宜、保存するフォルダを選択し、ファイル名を [リンク集.flas] として、[保存] ボタンをクリックします。



(4) テキストファイルの作成

- (a) メモ帳を開きます。
 (b) 以下の文字列を入力します。



&text1=センター Web ページ&url1=http://www1.iwate-ed.jp/&

- (c) ファイル名を [linkdata.txt] として保存します。ただし、先の [リンク集.flas] ファイルと同じフォルダに保存します。

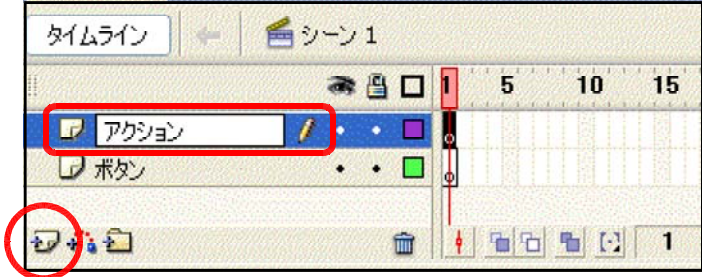
(5) ファイルの読込設定

(a) 先に作成した [リンク集.fla] ファイルを開きます。

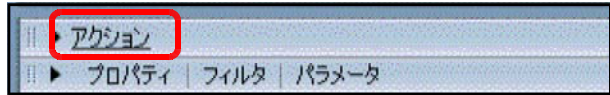


(b) レイヤー名を [ボタン] に変更します。

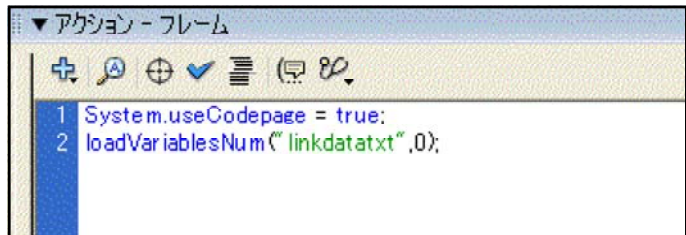
(c) [レイヤーの追加] ボタンをクリックして、[ボタン] レイヤー上に [アクション] レイヤーを作成します。



(d) [アクション] レイヤーの1フレーム目をクリックして選択して、[アクション] をクリックして、[アクション] パネルを表示させます。

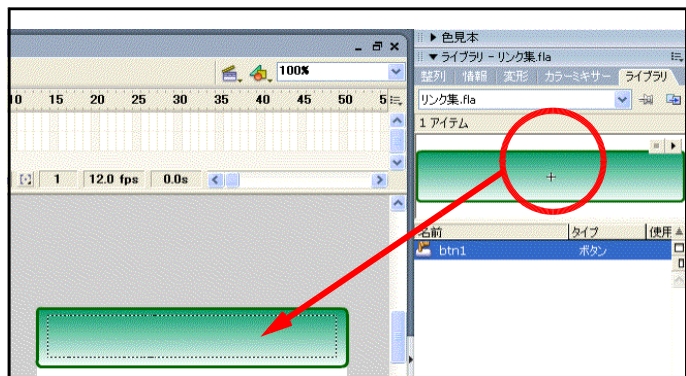


(e) 以下のスクリプトを入力します。

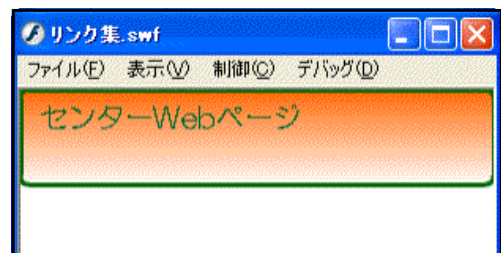


| | |
|--|-------------------------|
| System.useCodepage = true; | ← 文字コードを設定します |
| loadVariablesNum("linkdata.txt",0); | ← linkdata.txt から読み込みます |

(f) ライブラリから [btn1] シンボルをステージ上にドラッグして配置します。



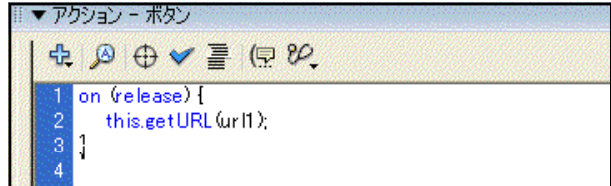
これで、外部のテキストファイルの内容を読み込んでボタンに表示できるようになりましたので、[Ctrl] キーと [Enter] キーを同時に押して、プレビューで動作を確認してみましょう。



(6) リンクの設定

- (a) ステージ上に配置されたボタンをクリックして選択します。
- (b) [アクション] をクリックして、アクションパネルを表示させます。
- (c) 以下のスクリプトを入力します。

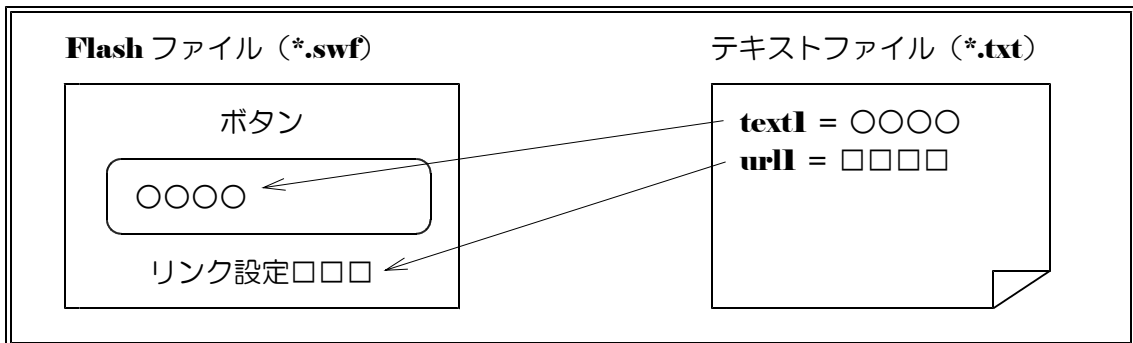
```
on (release) {
    this.getURL(url1);
}
```



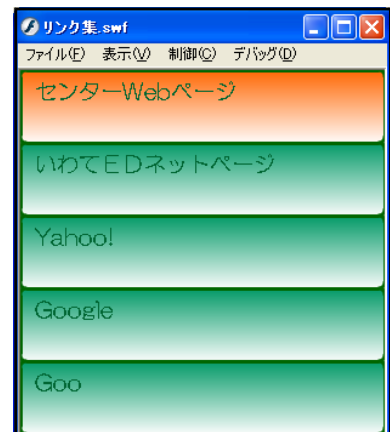
- (d) これで、ボタンに対するリンク設定ができました。[Ctrl] キーと同時に [Enter] キーを押して、プレビュー表示させます。ボタンをクリックして該当のページが表示されるか確認をしてみましょう（表示されない場合は、[参考資料] を参照）。



【ファイル読込のしくみ】



- 【応用】 ボタンの数を増やしてみましょう。
(ヒント) 変数 **text1** と変数 **url1** を追加していく。



参考資料

ActionScript

on (press) { ~ } ■ボタンまたはムービークリップのインスタンスに対して、マウスボタンをクリックしたら「~」でしてされたイベントを実行する

例) **on (press) {**
 x = 1; このムービークリップ上で、マウスボタンが押されたなら、
 } 変数 **x** に **1** を代入しなさい。

on (release) { ~ } ■ボタンまたはムービークリップのインスタンスに対して、マウスボタンが戻ったら「~」で指定されたイベントを実行する

例) **on (release) {**
 x = 1; このムービークリップ上で、マウスボタンが戻ったら、
 } 変数 **x** に **1** を代入しなさい。

on (release,releaseOutside) { ~ } ■ボタンまたはムービークリップのインスタンスに対して、マウスボタンが戻るか、ドキュメント外へカーソルが移動したときに「~」で指定されたイベントを実行する

例) **on (release,releaseOutside) {**
 y = 2; このムービークリップ上で、マウスボタンが戻るか、
 } ドキュメント外にマウスカーソルが移動したら、変数 **y**
 に **2** を代入しなさい。

startDrag() ■ムービークリップのインスタンスに対してドラッグを開始する

例) **this.startDrag()** このムービークリップのドラッグを開始する

stopDrag() ■ムービークリップのインスタンスに対してドラッグを終了する

例) **this.stopDrag()** このムービークリップのドラッグを終了する。

_x ■ムービークリップのX座標を取得する

例) **this._x = 600** このムービークリップの **x** 座標を **600** に設定する

| | |
|------------------------|------------------------------------|
| _y | ■ムービークリップの y 座標を取得する |
| 例 this._y = 400 | このムービークリップの y 座標を 400 に設定する |

| | |
|---------------------|-------------------|
| swapDepths() | ■ムービークリップの深度を変更する |
| | |

| | |
|-------------------------------------|---|
| useCodepage | ■文字コードを変更する |
| 例) System.useCodepage = true | このプロパティを true に設定すると、 Flash layer を実行するオペレーティングシステムの通常のコードページを使用して外部テキストファイルが解釈します。 |

| | |
|--|--|
| loadVariablesNum() | ■外部ファイルの値を読み込み、指定した変数に代入する |
| 例) loadVariablesNum("data.txt") | data.txt ファイルからデータを読み込み、指定した変数に代入する |

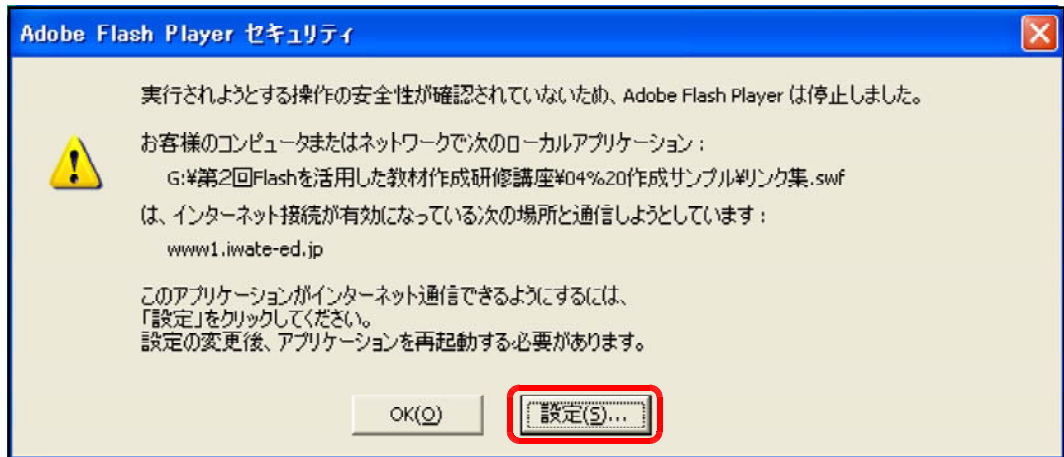
| | |
|---|-------------------------|
| getURL() | ■指定した URL からページを読み込む |
| 例) getURL("http://www.yahoo.co.jp/") | Yahoo! ページを表示させる |

<memo>

【FlashPlayer のセキュリティ設定】

Flash コンテンツから外部の Web サイトへリンク設定をしている場合、下図のようなダイアログが表示される場合があります。以下に外部のサイトへのアクセスを許可する設定を行います。ただし、インターネットへの接続ができる環境でのみ設定が可能です。

- [Adobe Flash Player セキュリティ] ダイアログ上で、[設定] ボタンをクリックします。



- Web 上の [FlashPlayer 設定マネージャー] ページが表示されますので、[グローバルセキュリティ設定パネル] をクリックします。



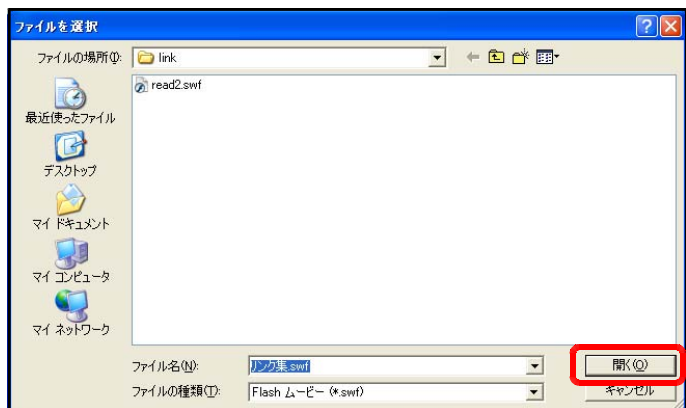
- [グローバルセキュリティ] パネルが表示されますので、[追加] をクリックします。



- [ファイルを参照] ボタンをクリックします。



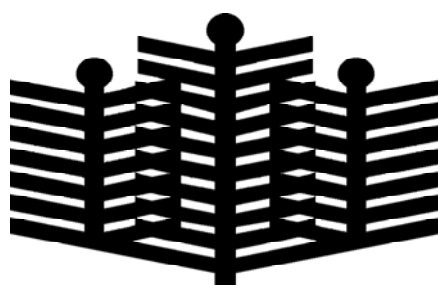
- **Flash** ファイル (*.swf) の保存先を指定して、ファイルを選択し、[開く] ボタンをクリックします。以上で設定完了です。



- [FlashPlayer 設定マネージャー] ページを閉じ、**Flash** コンテンツを再起動すると、外部の **Web** ページへのアクセスができるようになります。



1. 「WindowsXP」は、株式会社マイクロソフトの著作物であり、「WindowsXP」にかかる著作権その他の権利は、株式会社マイクロソフト及び各権利者に帰属します。
2. 「Flash8」は、株式会社アドビの著作物であり、「Flash8」にかかる著作権その他の権利は、株式会社アドビ及び各権利者に帰属します。
3. このテキストは、岩手県立総合教育センターで作成したものであり、ここに掲載されている内容について株式会社マイクロソフト並びに株式会社アドビは関与していません。
4. このテキストに関するご質問等は、岩手県立総合教育センター情報教育室(joho@center.iwate-ed.jp)までお問い合わせ下さい。



岩手県立総合教育センター
情報教育室
平成19年2月1日発行