

ActionScript

ブラウザで無料ではじめる [ActionScript 3.0](#) It's a [wonderfl](#) world

コンポーネントリファレンス

- http://livedocs.adobe.com/flash/9.0_jp/ActionScriptLangRefV3/

Flash CS3 Reference

- http://livedocs.adobe.com/flash/9.0_jp/main/wwhelp/wwhimpl/js/html/wwhelp.htm

IDE

FlashDevelop

- [FlashDevelop](#)

wonderfl

- [wonderfl](#)

型

変数の型

型	概要
int	整数
Number	整数、符号なし整数、浮動小数点数
uint	符号なし整数
Boolean	真偽値 (true または false)
String	文字列
Array	配列
Date	日付

変数

宣言

```
var 変数名 : 型 = 値 ;
```

オブジェクト指向

継承

extends

```
public class SubSprite extends Sprite {  
}
```

API / コンポーネント

グラフィック

表示リストの基本的要素 Sprite

パッケージ

- ・ flash.display

概要

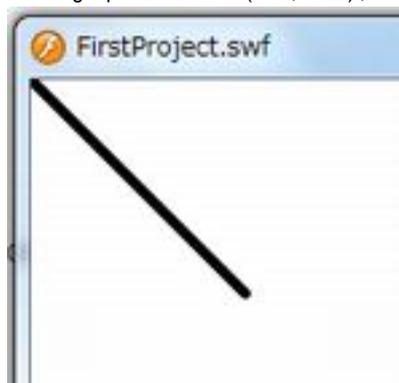
- ・ 表示リストの基本的要素
- ・ グラフィックを表示でき、子を持つこともできる表示リストノード
- ・ ムービークリップと似ていますが、タイムラインを持ちません。
- ・ タイムラインを必要としないオブジェクトに適した基本クラスです

線を引く

- ・ flash.display.Graphics

lineStyle(太さ , 色)、 lineTo(x,y)、 moveTo(x,y)

```
graphics.lineStyle(5, 0x000000);  
graphics.lineTo(100, 100);
```



前の線を引き終わった終点が次の線の始点となる。線を引かずに位置を移動する場合は `moveTo` を利用する

円を書く

- ・ flash.display.Graphics

drawCircle(x,y, 半径)

```
graphics.lineStyle(10, 0x000000);
graphics.drawCircle(100, 100, 50);
```

描画本体に別インスタンスを追加

addChild(インスタンス)

```
public class Main extends Sprite {
    public function Main():void {
        var h:Hoge = new Hoge();
        addChild(h); // 本体の描画処理にインスタンスを追加
        h.drawSomething();
    }
}
class Hoge() extends Sprite {
    public drawSomething():void {
        // 描画処理
    }
}
```

画像を読み込む

Loader

UI

テキストフィールド、イベントハンドラ、URL 送信、XML

TextField

- ・テキストの表示と入力用の表示オブジェクトを作成
- ・SWF ファイルのダイナミックテキストフィールドおよびテキスト入力フィールドは、すべて TextField クラスのインスタンス

addEventListener(イベント名, イベントリスナー)

- ・イベントリスナーオブジェクトを登録し、リスナーがイベントの通知を受け取るようにします

URLLoader

- ・指定した URL からテキスト、バイナリデータ、または URL エンコード形式の変数をダウンロードする際に使用

XML

- ・XML オブジェクトを操作するためのメソッドとプロパティを含む

例

```
package
{
```

```

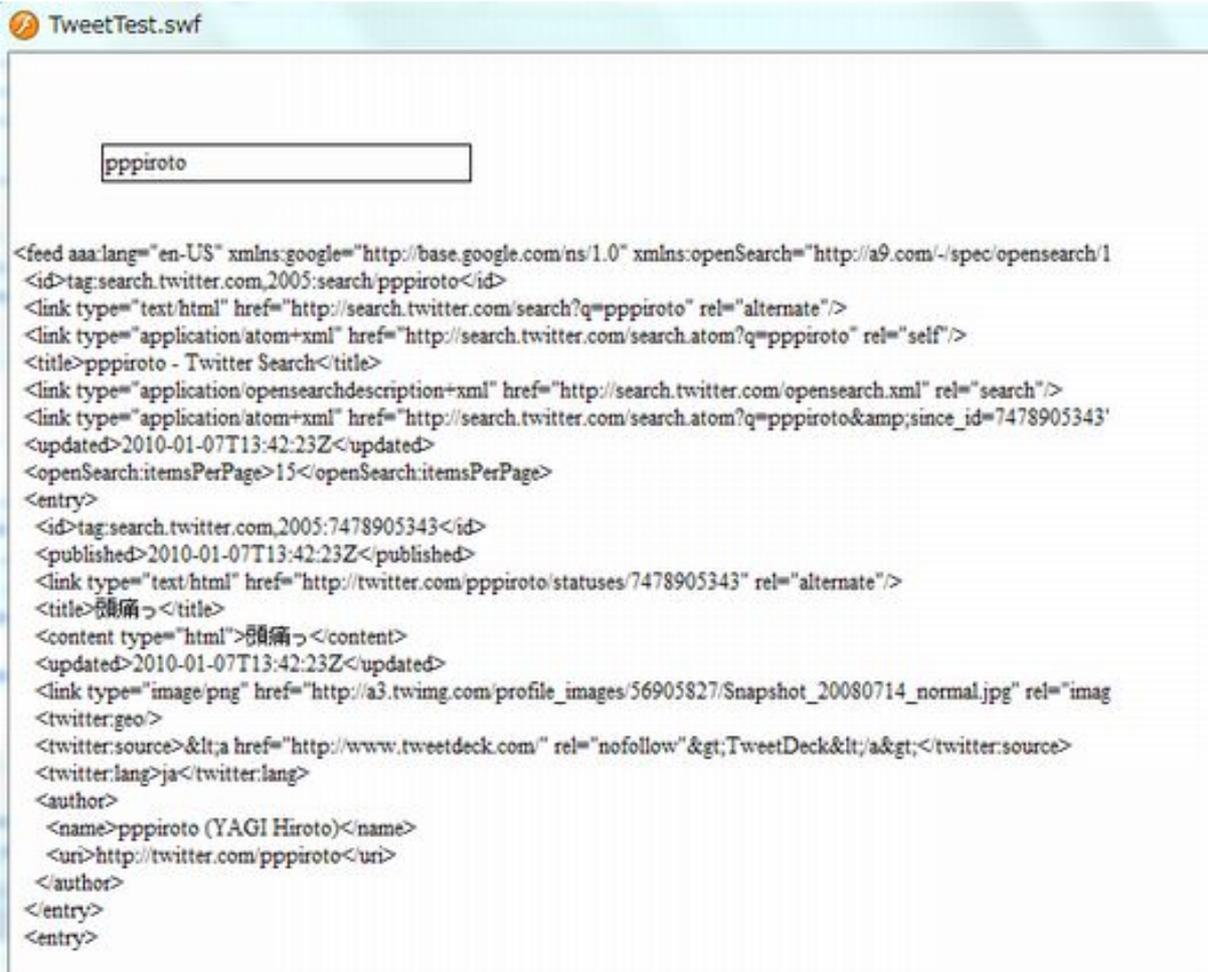
import flash.display.Sprite;
import flash.events.Event;
import flash.events.KeyboardEvent;
import flash.events.TextEvent;
import flash.net.URLLoader;
import flash.net.URLRequest;
import flash.text.*;

public class Main extends Sprite
{
    private var txt_keyword:TextField;
    private var txt_result:TextField;
    public function Main():void
    {
        // テキストフィールドを入力可能とする
        txt_keyword = new TextField();
        txt_keyword.type = TextFieldType.INPUT;
        txt_keyword.x = 50;
        txt_keyword.y = 50;
        txt_keyword.width = 200;
        txt_keyword.height = 20;
        txt_keyword.border = true;
        addChild(txt_keyword);
        // キーダウンイベントハンドラ
        txt_keyword.addEventListener(KeyboardEvent.KEY_DOWN, txt_keyword_onKeyDown);

        // 結果を表示するテキストフィールド
        txt_result = new TextField();
        txt_result.width = 600;
        txt_result.height = 400;
        txt_result.y = 100;
        addChild(txt_result);
    }
    private function txt_keyword_onKeyDown(e:KeyboardEvent):void
    {
        var twitter_url:String = "http://search.twitter.com/search.atom?q=";
        if (e.keyCode == 13) {
            // URL を送信し、結果を得る
            twitter_url += e.currentTarget.text;
            var url_loader:URLLoader = new URLLoader();
            // URL の結果を処理するイベントハンドラを登録
            url_loader.addEventListener(Event.COMPLETE, xml_loaded);

            // 送信
            url_loader.load(new URLRequest(twitter_url));
        }
    }
    // 結果を処理するイベントハンドラ
    private function xml_loaded(e:Event):void
    {
        txt_result.text = "";
        var twitter_xml:XML = new XML(e.currentTarget.data);
        txt_result.text = twitter_xml;
    }
}
}

```



テキスト処理

URI エンコード

encodeURIComponent(文字列)

- ・グローバル関数 (任意の箇所やユーザー定義クラスで使用できる)
- ・文字列を有効な URI (Uniform Resource Identifier) にエンコード

encodeURIComponentComponent(文字列)

- ・グローバル関数 (任意の箇所やユーザー定義クラスで使用できる)
- ・文字列有効な URI コンポーネントにエンコード

encodeURIComponent では、"&" や "+" や "=" などの文字をエンコードしないので、GET や POST メソッドで利用するときは、encodeURIComponent を使用する。こちらは、英数以外は、"-_!.~*" のみがエスケープされない。

XML

default xml namespace ディレクティブ

- ・ XML オブジェクトに使用するデフォルトの名前空間を設定

```
default xml namespace = new Namespace("http://www.w3.org/2005/Atom");
```

XML オブジェクト

- ・ 1つの XML オブジェクトは、1つの XML エlement、属性、コメント、処理命令、またはテキストElementを表します。
- ・ "単純内容"を持つものと"複合内容"を持つものに分類されます。
- ・ 複合内容の XML オブジェクトとは、子ノードを持つ XML オブジェクトです。
- ・ 単純内容の XML オブジェクトとは、属性、コメント、処理命令、テキストノードのいずれか1つを含んだ XML オブジェクトです。

XML 操作クラス

- ・ XML 構造化情報を操作するためのクラスがいくつか含まれている。
- ・ 2つのメインクラス
- ・ XML: 単一の XML Element。複数の子Elementまたはドキュメント内に単一値のElementを持つ XML ドキュメント。
- ・ XMLList: XML Elementセット。兄弟Element (XML ドキュメント階層で同じレベルにあり、同じ親に含まれているElement)である複数の XML Elementが存在する場合、XMLList オブジェクトが使用される。

階層構造内の移動

デバッグ

trace(任意の数のカンマ区切りの引数)

- ・ デバッグ中に式を表示、またはログファイルに書き込み
- ・ [FlashDevelop](#) で 利用するには

サンプル

アニメーション

- ・ [ActionScript アニメーション](#)