

[概要](#)
[プログラム](#)
[検証用コード](#)

概要

ベジェトゥイーンを作成してみましょう。

普通のトゥイーンでは開始位置と終了位置を指定するだけでしたが、ベジェトゥイーンでは制御位置という新しいポイントも指定することになります。

ベジェのアルゴリズムについては、[curveToで曲線を描く](#)を参照してください。

では、実際にベジェトゥイーンを作成してみましょう。

BetweenAS3.bezier()を使用します。
第四引数

controlPoint:Object

コントロールポイント(制御位置)を指定します。

bezier()の使い方はtween()とほとんど同じです。

第四引数にcontrolPointが追加されただけです。コントロールポイントの意味は上で説明しましたね、引数の指定の仕方はfrom, toと同じで、プロパティ名:値を中括弧(波括弧)で括ります。

プログラム

```
var rect:Sprite = new Sprite();
rect.graphics.beginFill(0xFF0000);
rect.graphics.drawRect(0, 0, 50, 50);
rect.graphics.endFill();
addChild(rect);
```

```
BetweenAS3.tween(rect, { x:300, y:200 }).play();
```

[BetweenAS3でオブジェクトを動かす](#)をもとにベジェトゥイーンを作成します。

```
// BetweenAS3.tween(rect, { x:300, y:200 }).play();
BetweenAS3.bezier(rect, { x:300, y:0 }, null, { x:150, y:600 }, 5.0).play();
```

tween()を消してbezier()を新しく加えましょう。

目標値は(300, 0)に変更しておきます。

コントロールポイントは(150, 600)なので、実際に通るポイントは(150, 300)ですね。

実行してみるとrectオブジェクトがカーブを描くようにトゥイーンします。

検証用コード

```
package
{
import flash.display.Sprite;
import org.libspark.betweenas3.BetweenAS3;

public class Main extends Sprite
{
public function Main()
{
var rect:Sprite = new Sprite();
rect.graphics.beginFill(0xFF0000);
rect.graphics.drawRect(0, 0, 50, 50);
rect.graphics.endFill();
addChild(rect);

BetweenAS3.bezier(rect, { x:300, y:0 }, null, { x:150, y:600 }, 5.0).play();
}
}
}
```