

イベントの概念

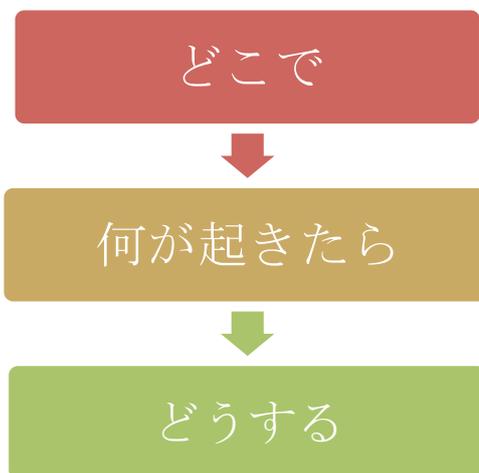
ActionScript3.0 を学ぶ上で、イベント処理は非常に重要な概念になります。イベント処理とは「何かが起こったことをきっかけに、何かを実行する」ということです。例えばマウスで「ボタン」をクリックしたらステージの「自動車」が動きだす。といったことです。

主なイベントの種類には、マウス、キーボード入力などがあります。

ActionScript3.0 ではイベントリスナーという仕組みを使ってイベント処理を行います。

イベントリスナーの追加

「インスタンス `box_mc` をクリックしたら `test()` を実行する」といったインタラクティブ（対話式）な処理を考えたとき、この中には「1 どこで、2 何が起きたら、3 どうする」の3つの要素が含まれています。



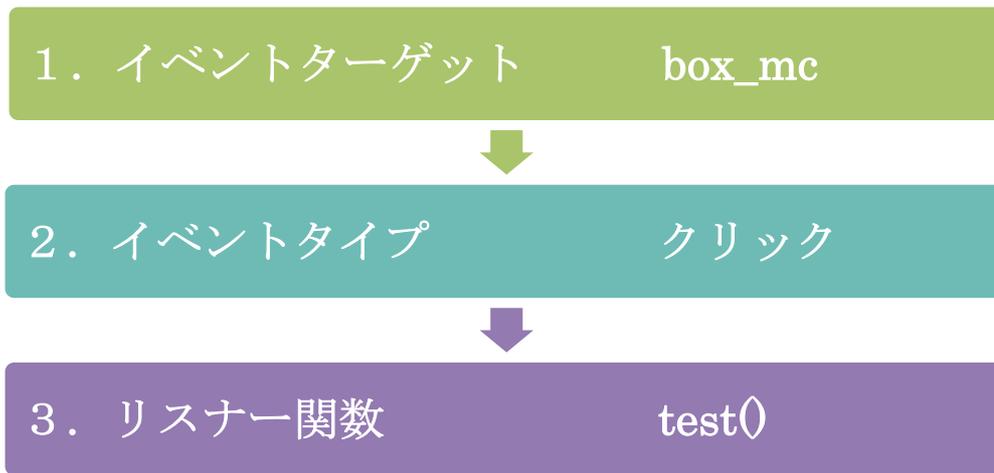
これをイベント処理の用語で言い換えると次のようになります。

1. イベントターゲット イベントが発生するオブジェクト

2. イベントタイプ 発生するイベントの種類

3. リスナー関数 イベント発生で実行する関数

先ほどの「インスタンス `box_mc` をクリックしたら `test()` を実行する」をこれに当てはめると次のようになります。



このイベント処理をスクリプトでは `addEventListener()` というメソッドを使って設定します。書式は次のようになります。イベントが発生するオブジェクトに対して、イベントタイプとリスナー関数をペアにしてイベントリスナーとして追加します。

```
イベントターゲット.addEventListener(イベントタイプ,リスナー関数)
```

✚ イベントの発信元を知るには

```
//リスナー関数を作る

function eventTest(evt:MouseEvent):void{

    evt_target.rotation+=10;

}

//イベントリスナでボタンと関数を結び付ける

btn_mc.addEventListener(MouseEvent.CLICK,eventTest);
```

🚩 複数の関数を実行させる

```
//リスナー関数を作る

function eventTest1(evt:MouseEvent):void{

    trace("eventTest1");

}

function eventTest2(evt:MouseEvent):void{

    trace("eventTest2");

}

//イベントリスナでボタンと関数を結び付ける

btn_mc.addEventListener(MouseEvent.CLICK,eventTest1);

btn_mc.addEventListener(MouseEvent.CLICK,eventTest2);
```

🚩 イベントリスナを削除する

```
btn_mc.removeEventListener(MouseEvent.CLICK,eventTest1);
```

🚩 イベントの種類

イベント	イベントが発生するタイミング
MouseEvent.MOUSE_DOWN	マウスボタンが押されたら
MouseEvent.MOUSE_UP	マウスボタンが離れたら
MouseEvent.CLICK	マウスボタンがクリックされたら (押して離す)
MouseEvent.DOUBLE.CLICK	マウスボタンがダブルクリックされたら
MouseEvent.MOUSE_MOVE	マウスカーソルが動いたら
KeyboardEvent.KEY_DOWN	キーが押されたら
Event.ENTER_FRAME	再生ヘッドがフレームに来たら
Event.COMPLETE	外部ファイルの読み込みが完了したら

ENTER_FRAME について

ENTER_FRAME は繰り返し発生するイベントです。フレームレートに合わせて繰り返し発生するフレーム再生イベントは、アニメーションで利用する繰り返しイベントとして最適なイベントです。デフォルトのフレームレートは毎秒 12 フレームなので、デフォルトの ENTER_FRAME の頻度は毎秒 12 回になります。

```
//リスナー関数を作る
```

```
function eventTest(evt:MouseEvent):void{
```

```
    evt_target.rotation+=10;
```

```
}
```

```
//イベントリスナで ENTER_FRAME の設定をする
```

```
evt_target.addEventListener(Event.ENTER_FRAME,eventTest);
```

例題 四角形を描き、四角形をマウスでクリックすると45度回転させてみましょう。四角形は以下のとおり描きます。

解答 Event プロジェクト Onclick2.as

```
var rect:Sprite = new Sprite();
rect.graphics.beginFill(0x00FF00);
rect.graphics.drawRect(-75, -50, 150, 100);
rect.graphics.endFill();
rect.x = rect.y = 100;      // 四角形を x 座標 y 座標 100 px に置く
addChild(rect);
```

```
public function Onclick():void {
    var rect:Sprite = new Sprite();
    rect.graphics.beginFill(0x00FF00);
    rect.graphics.drawRect(-75, -50, 150, 100);
    rect.graphics.endFill();
    rect.x = rect.y = 100;
    addChild(rect);
    rect.addEventListener(MouseEvent.CLICK, test);
    //クリックで実行されるリスナー関数
    function test(eventObj:MouseEvent):void {
        //box_mc を 45 度回転させる
        rect.rotation += 45;
    }
}
```

課題 ステージをマウスでクリックすると四角形が現れ、クリックするたびに色が変わるようにしてみよう。

参考 ランダムメソッドで色を変えます。使用方法は以下

Math.random()*0xFFFFFFFF

ステージのイベントを取得するには次のイベントイベントリスナーを使う。

```
stage.addEventListener(MouseEvent.CLICK, Onclick);
```

解答 Event プロジェクト Onclick2.as

```
public function Onclick2():void
{
    stage.addEventListener(MouseEvent.CLICK, Onclick);
    function Onclick(evt:MouseEvent):void {
        var mc:Sprite = new Sprite();
        var fillcolor:uint = 0xFFFFFFFF * Math.random();
        mc.graphics.beginFill(fillcolor);
        mc.graphics.drawRect(0, 0, 100, 150);
        mc.graphics.endFill();
        mc.x = 100;
        mc.y = 100;
        addChild(mc);
    }
}
```

課題 Event.ENTER_FRAME を使って、四角形を連続回転してみよう。

次に四角形の中心点を次のように x,y 座標を 0 にして回転してみよう。

```
rect.graphics.drawRect(0, 0, 150, 100);
```

解答 Event プロジェクト **Renzoku.as**

```
public function Renzoku():void
{
    var rect:Sprite = new Sprite();
    rect.graphics.beginFill(0x00FF00);
    rect.graphics.drawRect(-75, -50, 150, 100);
    rect.graphics.endFill();
    rect.x = rect.y = 100;
    addChild(rect);

    rect.addEventListener(Event.ENTER_FRAME, test);
    //クリックで実行されるリスナー関数
    function test(eventObj:Event):void {
        //box_mc を 45 度回転させる
        rect.rotation += 45;
    }
}
```

課題 Event.ENTER_FRAME を使って、四角形を右側に等速運動させましょう。

解答 Event プロジェクト **Renzoku2.as**

```
public function Renzoku2():void
{
    var rect:Sprite = new Sprite();
    rect.graphics.beginFill(0x00FF00);
    rect.graphics.drawRect(-75, -50, 150, 100);
    rect.graphics.endFill();
    rect.x = rect.y = 100;
    addChild(rect);

    rect.addEventListener(Event.ENTER_FRAME, test);
    //クリックで実行されるリスナー関数
    function test(eventObj:Event):void {
        //box_mc を 45 度回転させる
        rect.x += 5;
    }
}
```