

[概要](#)
[プログラム](#)
[検証用コード](#)

概要

BitmapData#dispose()でBitmapDataのメモリを解放してみましょう。

プログラム

```
public function Main()  
{  
    var bd:BitmapData = new BitmapData(300, 300, false, 0xED1A3D);  
}
```

まず何でもいのでBitmapDataオブジェクトを用意します。

このBitmapDataで使用されているメモリを解放対象にするには、

ローカル変数でかつBitmapに関連付けられていないので、Main()を抜けると自動削除
nullを入れる
の二つです。

しかし、この方法ではガベコレの対象にはなるのですが、いつ解放されるのかはガベコレのタイミング次第になります。
どうしても今すぐ解放したいという場合の為にBitmapData#dispose()というメソッドが用意されています。

```
bd.dispose();
```

BitmapDataの中身(ピクセル)が削除、つまりメモリ解放されます。

```
bd.dispose();  
trace(bd == null); // false
```

解放といってもBitmapDataオブジェクト自体がnullになるわけではありません。
しかし、もうプロパティやメソッドにアクセスすることが出来ないので事実上nullを入れたのと同じような状態になっています。

検証用コード

```
package  
{  
    import flash.display.Bitmap;  
    import flash.display.BitmapData;  
    import flash.display.Sprite;
```

```
import flash.events.MouseEvent;

public class Main extends Sprite
{
    private var bd:BitmapData;
    private var bitmap:Bitmap;

    public function Main()
    {
        bd = new BitmapData(2000, 2000, false, 0xED1A3D);

        var bitmap:Bitmap = new Bitmap(bd);
        addChild(bitmap);

        stage.addEventListener(MouseEvent.CLICK, onMouseDown);
    }

    private function onMouseDown(event:MouseEvent):void
    {
        bd.dispose();
        trace(bd == null); // false
    }
}
```