

移動イベント

移動とサイズ変更に対応する

コントロールを、特にメインウィンドウをユーザーが移動させたりサイズを変更した場合
それにに応じて何らかの対応をしなければならないというケースがしばしばあります
ウィンドウを移動すると **Control.OnMove()** メソッドが呼び出されます

```
protected virtual void OnMove(EventArgs e);
```

移動イベントなので、コントロールの位置を示す情報を引数に渡してもよい気しますが
コントロールの座標はプロパティから直接得られるため
このメソッドは独自のイベント引数クラスは持たず、EventArgs を受けます

移動イベントをイベントメンバで処理したい場合は **Control.Move** を使います

```
public event EventHandler Move;
```

これらの機能を使うことによって、ウィンドウの移動を感知することができます

```
using System.Windows.Forms;
using System.Drawing;

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }

    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        g.DrawString(Location.ToString(), Font, Brushes.Black, 0, 0);
    }

    override protected void OnMove(System.EventArgs e) {
        Invalidate();
    }
}
```



このプログラムは、ウィンドウがユーザーによって移動させられると
OnMove() メソッドが呼び出され、クライアント領域を再描画します
この機能によって、ウィンドウのクライアント領域は
常にリアルタイムでウィンドウの左上隅のスクリーン座標を描画します

同様に、コントロールのサイズ変更に対応するには
Control.OnResize() メソッドをオーバーライドします

```
protected virtual void OnResize(EventArgs e);
```

e には、EventArgs オブジェクトが渡されます
イベントメンバでこのイベントを処理するには **Control.Resize** を使います

```
public event EventHandler Resize;
```

以前、サイズ変更時に再描画も行うウィンドウの実現方法としてスタイルを使用しましたが
スタイルを設定しなくても、直接イベントを拡張して実現することができます

```
using System.Windows.Forms;
using System.Drawing;

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }

    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        g.DrawString(Size.ToString(), Font, Brushes.Black, 0, 0);
    }

    override protected void OnResize(System.EventArgs e) {
        Invalidate();
    }
}
```

このプログラムは、「スタイル」の章のプログラムを改良したものです
動作結果はまったく同じですが、イベントを直接オーバーライドして再描画しています

