

再描画

描画イベントを発生させる

OnPaint() イベントは、クライアント領域に無効リージョンが作られると発生します
ウィンドウが他のウィンドウに隠れたり、最小化されたりすると
その部分は画面から消えるため、情報を失います

そして、またウィンドウが表示されると隠された部分は無効な状態になっているため
無効リージョンを再び描画する必要がありますので、OnPaint() が呼び出されるのです
因みに、Windows の仕様でマウスカーソルに隠れる部分はメモリに保存されます
つまり、マウスカーソルに隠された部分は無効になることはありません

しかし、それ以外にも明示的に再描画してほしいという場合があります
例えばイベントによって情報が変化した、クライアント領域を再描画するなどです
再描画には **Control.Invalidate()** メソッドを使います

```
public void Invalidate();  
public void Invalidate(bool invalidateChildren);  
public void Invalidate(Rectangle rc);  
public void Invalidate(Region region);  
public void Invalidate(Rectangle rc, bool invalidateChildren);  
public void Invalidate(Region region, bool invalidateChildren);
```

このメソッドは、強制的に向こうリージョンを作りだして再描画します
invalidateChildren は true ならば子コントロールも無効化することを意味します
false を指定すれば、子コントロールは無効化されません

rc は無効化するリージョンを示す長方形を region はリージョンを指定します
再描画しなければいけない範囲がわかっているのであれば、指定するべきです
特に高速な描画を求める場合、無効領域は少ない方が速く動作します

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    int x = 0, y = 0;  
    public static void Main(string[] args) {  
        Application.Run(new WinMain());  
    }  
    override protected void OnPaint(PaintEventArgs e) {  
        Graphics g = e.Graphics;  
        g.DrawString("X = " + x + " : Y = " + y, Font, Brushes.Black, 0, 0);  
    }  
  
    override protected void OnMouseMove(MouseEventArgs e) {  
        x = e.X; y = e.Y;  
        Invalidate();  
    }  
}
```



このプログラムは、マウスがクライアント領域上を移動すると
インスタンスメンバ変数の x と y を現在のマウスカーソルの座標に更新し
Invalidate() メソッドを呼び出して無効リージョンを作り、OnPaint() を呼び出します

また、コントロールと子コントロールを無効化して再描画する
Control.Refresh() メソッドも定義されています

```
public virtual void Refresh();
```

子コントロールの情報を最新のものに更新する時には
このメソッドを使うなどして、子コントロールを再描画する必要があります

[前のページへ](#)

[戻る](#)

[次のページへ](#)