

<MQL4 ; 基礎の確認 (その2) >

○テキスト (オブジェクト) を表示する ; OBJ_TEXT を使う

1. コード

```
//
//      テキストを表示する
// OBJ_TEXT は&%示位置の指定に「日時」と「価格」を使う。

int init()
{
    ObjectCreate("myObj_1", OBJ_TEXT, 0, 0, 0);
    return(0);
}

int deinit()
{
    ObjectDelete("myObj_1");
    return(0);
}

//スタート関数-----
int start()
{
    //int と double は自動で文字に変換される、
    DoubleToStr(Close[0])はエラーになる
    string st="最新の価格は、"+Close[0]+" です。";
    //ObjectSetText("myObj_1", "表示できたか?", 12, "MS
    S ゴシック", Black);
    ObjectSetText("myObj_1", st, 24, "MS ゴシック
    ", Black);
    ObjectMove("myObj_1", 0, Time[10], (High[10]+0.05));

    return(0);
}
```

2. 結果の画面



○テキスト（オブジェクト）を表示する；OBJ_LABELを使う

1. コード

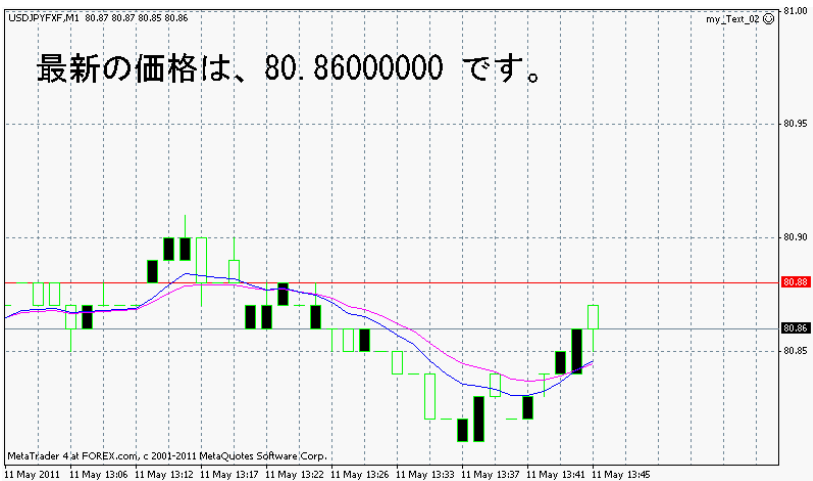
```
//  
//      テキストを表示する  
// OBJ_LABEL は表示位置の指定に「XY座標（ピクセル）；左上  
// が原点」を使う。
```

```
int init()  
{  
    ObjectCreate("myObj_1", OBJ_LABEL, 0, 0, 0);  
    return(0);  
}
```

```
int deinit()  
{  
    ObjectDelete("myObj_1");  
    return(0);  
}
```

```
//スタート関数-----  
int start()  
{  
    //int と double は自動で文字に変換される、  
    DoubleToStr(Close[0]) はエラーになる  
    string st="最新の価格は、"+Close[0]+" です。";  
    ObjectSetText("myObj_1", st, 24, "MS ゴシック", Black);  
  
    ObjectSet("myObj_1", OBJPROP_XDISTANCE, 30);  
    ObjectSet("myObj_1", OBJPROP_YDISTANCE, 40);  
  
    return(0);  
}
```

2. 結果の画面



○関数化・・・原理確認用「超簡単な配列の授受」

・メインから関数に配列を渡し、処理結果の配列を受け取る。

1. 1本化コード

(1) コード

```
//
//      関数開発の確認用
//      配列を参照渡しする
//
//      1 本版 Script で確認
#property show_confirm          //最初に1回止める

int init()
{
    ObjectCreate("myObj_1",OBJ_LABEL,0,0,0);
    return(0);
}

int deinit()
{
    ObjectDelete("myObj_1");
    return(0);
}

void adSeries(int n,int& series1[],int& series2[],int& result[])
{
    int i;
    for (i = 0 ; i < n ; i++) {
        result[i] = series1[i] + series2[i]; //結果を
result に格納
    }
}
//スタート
int start()
{
    int a[4]={10,20,30,40};
    int b[4]={50,60,70,80};
    int c[4];
```

```
adSeries(4,a,b,c);

Comment("Result:c[0]=",c[0]," :c[1]=",c[1]," :c[2]=",c[2]," :
c[3]=",c[3]);//文字サイズ固定

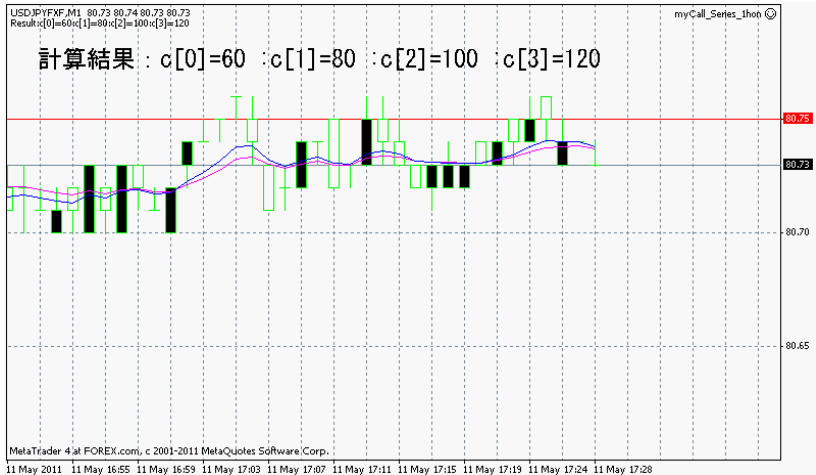
string st="計算結果 :
c[0]="+c[0]+" :c[1]="+c[1]+" :c[2]="+c[2]+" :c[3]="+c[3];
ObjectSetText("myObj_1",st,18,"MS ゴシック
",Black);

ObjectSet("myObj_1",OBJPROP_XDISTANCE,30);
ObjectSet("myObj_1",OBJPROP_YDISTANCE,40);

return(0);

}
```

(2) 結果のチャート



2. 分割コード (関数化)

(1) コード

ー 1. メインコード 「myCall_Series.mq4」

```
//  
//      関数開発の確認用  
//      配列を参照渡しする  
//  
//      メイン; 分割版  
#include <myHead.mqh>  
#property show_confirm           //最初に 1 回止める  
  
int init()  
{  
    ObjectCreate("myObj_1", OBJ_LABEL, 0, 0, 0);  
    return(0);  
}  
  
int deinit()  
{  
    ObjectDelete("myObj_1");  
    return(0);  
}  
  
//スタート  
int start()  
{  
    int a[4]={10, 20, 30, 40};  
    int b[4]={50, 60, 70, 80};  
    int c[4];  
  
    adSeries(4, a, b, c);  
  
    Comment("Result:c[0]=", c[0], ":c[1]=", c[1], ":c[2]="  
    , c[2], ":c[3]=", c[3]); //文字サイズ固定  
  
    string st="関数化の計算結果 :  
c[0]="+c[0]+" :c[1]="+c[1]+" :c[2]="+c[2]+" :c[3]="+c[3];  
    ObjectSetText("myObj_1", st, 18, "MS ゴシック
```

“,Black);

```
ObjectSet("myObj_1", OBJPROP_XDISTANCE, 30);
ObjectSet("myObj_1", OBJPROP_YDISTANCE, 40);

return(0);
```

}

－ 2. ヘッダファイル「myHead.mqh」

```
//
//      『 myHead 』
//      確認用ヘッダファイル
//      ¥experts¥include¥ に置くこと
//
//
#import "myFunctions.ex4"
void myComment(string com);
void adSeries(int n, int& series1[], int&
series2[], int& result[]);
#import
```

－ 3. ライブラリ（ファイル）「myFunctions.mq4」

```
//
//
//      関数開発の確認用
//
//      まず、現在の終値を表示する簡単なコードを「関数化」する
//
#property library

void myComment(string com)
{

    Comment(com);

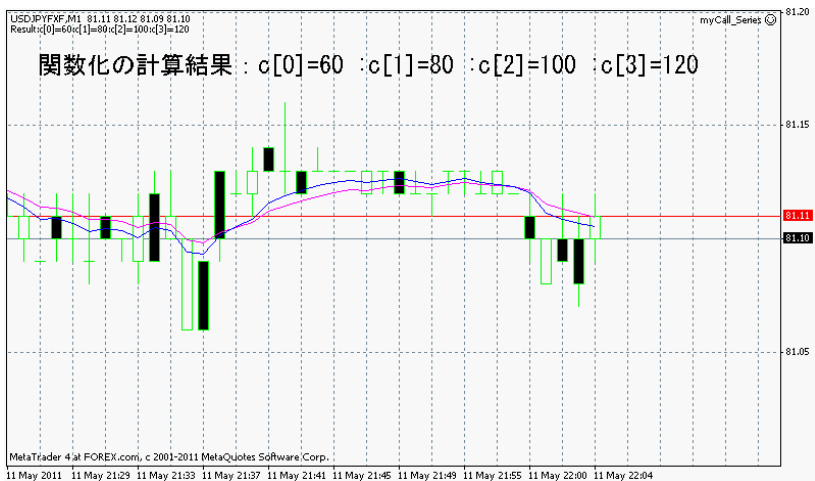
}
```

```
void adSeries(int n, int& series1[], int& series2[], int&
```

```
result[]
{
    int i;
    for (i = 0 ; i < n ; i++) {
        result[i] = series1[i] + series2[i]; //結果
    }
}
```

を result に格納

(2) 結果の画面



○ema の関数化

- ・移動平均線 ; ema をライブラリ (関数) を使い Simple 化する。

<判明事項>

- ・「SetIndexBuffer0」などのインディケータの設定が、
 - ①ヘッダファイルに記述すると動く。OK
 - ②ライブラリ (関数) に記述すると動かない。NG

1. コード

(1) ヘッダファイル

```
//
//      『 myHead.mqh 』
//      ヘッダファイル
//      \experts\include\ に置くこと
//
//
#property indicator_buffers 7 //SetIndexBuffer とは異なる役割？

#import "myFunctions.ex4"
    void ema(double& line[],int PRICE,int period);
    void ema_(double& line[],double& source[],int period);
    int line4(double& line[],string st,int style,int colo,int w);
#import
//
//---- buffers
double line0[2000];
double line1[2000];//不思議なことに、line_1[]はダメで、
line_1[2000]などとするとOKのことあり
double line2[2000];
double line3[2000];
double line4[2000];
double line5[2000];
double histogram[2000];

static double source[2000];
//
```

```
//-----
void line0(string st,int style,int colo,int w)// 「stylw,colo,w」は全
て int で受けること！
{
    SetIndexBuffer(0,line0);
    SetIndexLabel(0,st);
    SetIndexStyle(0, DRAW_LINE,style,w,colo);
    IndicatorDigits(Digits+2);
    SetIndexDrawBegin(0, 0);
    return;
}
void line1(string st,int style,int colo,int w)// 「stylw,colo,w」は全
て int で受けること！
{
    SetIndexBuffer(1,line1);
    SetIndexLabel(1,st);
    SetIndexStyle(1, DRAW_LINE,style,w,colo);
    IndicatorDigits(Digits+2);
    SetIndexDrawBegin(1, 0);
    return;
}
void line2(string st,int style,int colo,int w)// 「stylw,colo,w」は全
て int で受けること！
{
    SetIndexBuffer(2,line2);
    SetIndexLabel(2,st);
    SetIndexStyle(2, DRAW_LINE,style,w,colo);
    IndicatorDigits(Digits+2);
    SetIndexDrawBegin(2, 0);
    return;
}
void histogram(string st,int style,int colo,int w)// 「stylw,colo,w」
は全て int で受けること！
{
    SetIndexBuffer(6,histogram);
    SetIndexLabel(6,st);
    SetIndexStyle(6, DRAW_HISTOGRAM,style,w,colo);
    //IndicatorDigits(Digits+2);
    SetIndexDrawBegin(6, 0);
}
```

```

    return;
}

```

(2) ライブラリ ; 関数

```

//
//   ライブラリ 『myFunctions.mq4』
//
#property library

//
void ema(double& line[],int PRICE,int period)
{
    static double ema[2000];

    for(int j=0;j<1000;j++)
    {
        ema[j]=iMA(NULL, 0,
period,0,MODE_EMA,PRICE, j);
    }

    ArrayCopy(line,ema);

    return;
}
//
void ema_(double& line[],double& source[],int period)
{
    static double ema_[2000];
        //ArraySetAsSeries(source,true);

    for(int j=0;j<1000;j++)
    {
        ema_[j] = iMAOnArray(source, 0, period, 0,
MODE_EMA, j);
    }

    ArrayCopy(line,ema_);
}

```

```

        return;
    }
    /*
    //再度試してみることに
    int line4(double& line4[],string st,int style,int colo,int w)//
    「stylw,colo,w」は全て int で受けること！
    {
        IndicatorBuffers(2);//ライン計算が1個なら「1」でよい
        SetIndexBuffer(0,line4);

        SetIndexLabel(0,st);
        SetIndexStyle(0, DRAW_LINE,style,w,colo);
        IndicatorDigits(Digits+2);
        SetIndexDrawBegin(0, 0);

        int err=GetLastError();
        return(err);
    }
    */
    //-----

```

(3) メインコード return_Array_02_2.mq4

```

//
//|          移動平均 『return_Array_02_2.mq4』
//+-----+
#include <myHead.mqh>
//#include <stdlib.mqh>//エラー解析用
#property indicator_chart_window

extern int period = 15;

int init()
{
    //--- indicators
        IndicatorBuffers(10);
        line0("myIMA_0",STYLE_DASH,Red,1);
        line1("myIMA_1",STYLE_SOLID,Blue,1);
    //---

```

```

        ObjectCreate("myObj_1",OBJ_LABEL,0,0,0);
        return(0);
    }
//+-----+
int deinit()
{
    //----
        ObjectDelete("myObj_1");
    //----
    return(0);
}
//+-----+
int start()
{
    int counted_bars=IndicatorCounted();
    //----

    ema(line0,PRICE_CLOSE,period);

    ArraySetAsSeries(source,true);
    for(int i=0;i<200;i++)
    {
        source[i]=Low[i];
    }

    ema_(line1,source,period);

        string st_=" : line0[30]="+line0[30]+" :
line1[30]="+line1[30];
        ObjectSetText("myObj_1",st_,10,"MS ゴシック
",Black);

        ObjectSet("myObj_1",OBJPROP_XDISTANCE,20);
        ObjectSet("myObj_1",OBJPROP_YDISTANCE,40);
    //----
    return(0);
}
//+-----+

```

2. 結果の画像



○縦線を連続して描く

- ・条件が合致した場合に「縦線」を描く方法

1. コード

```
//V_Line.mq4 |
//
#property indicator_chart_window
//
int init()
{
    for(int i=0;i<=500;i=i+100)
    {
        string ad=i;
        ObjectCreate("V_Line"+ad,OBJ_VLINE,0,Time[i],0);
    }

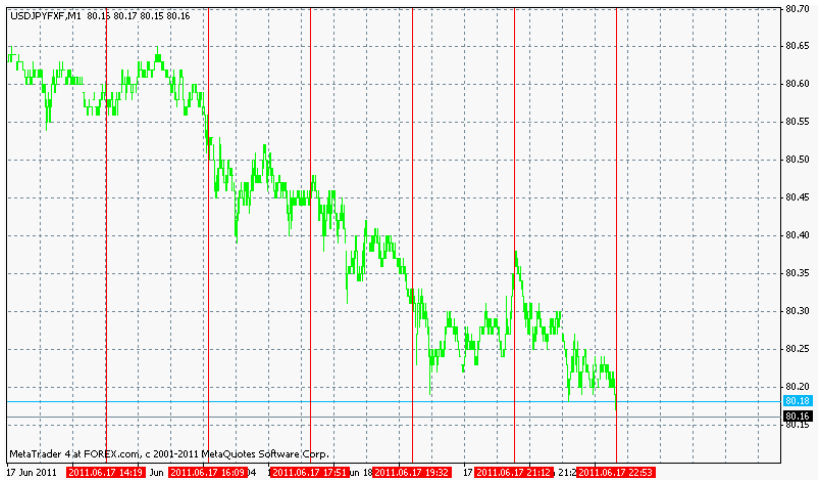
    return(0);
}
//
int deinit()
{
    for(int i=0;i<=500;i=i+100)
    {
        string ad=i;
        ObjectDelete("V_Line"+ad);
    }

    return(0);
}

//+-----+
int start()
{
    int    counted_bars=IndicatorCounted();
//----
    for(int i=0;i<=500;i=i+100)
    {
        string ad=i;
        ObjectSet("V_Line"+ad,OBJPROP_TIME1,Time[i]);
    }
}
```

```
}  
//----  
    return(0);  
}  
//+-----+-----+-----+-----+-----+-----+-----+
```

2. 結果のチャート



以 上