

<MQL4 ; 基礎の確認 (その1) >

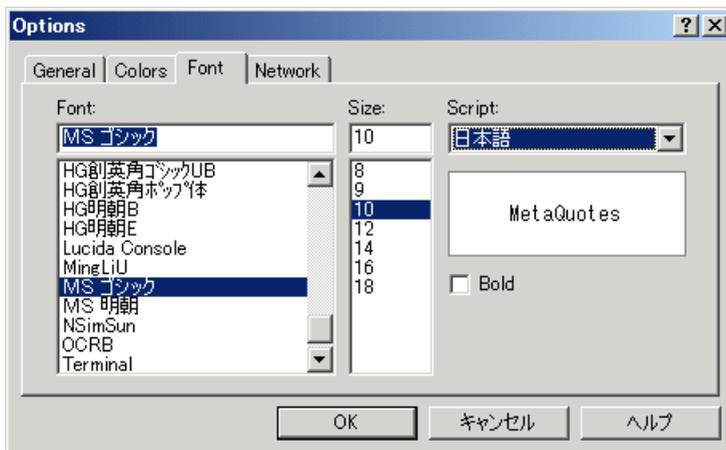
○メタエディタで日本語を使う設定

- ・メタエディタを開き、

[Tools]-[Options]で開く設定窓の[Font]タブで、

①MS ゴシックやMS 明朝などの日本語フォントを選び、

②次に[Script:]で「日本語」を選択する。



○スクリプトを他のテキストエディタで書く場合の注意

- ・文字コードは「SHIFT-JIS」を使うこと。

メタエディタは「SHIFT-JIS」を使っているため、

「UTF-8N」などの設定のままで使い、MT4 側に持ってくると、メタエディタで開いたとき文字化けする。

○EA ファイル (.mq4) を外部のエディタで作成後に、

experts ホルダに入れたが、MT4 のメタエディタを開くと、navigator ウィンドウに表示されない。

- ・MT4 を立上げ直すと、navigator ウィンドウに表示された。

○スクリプトの動作確認

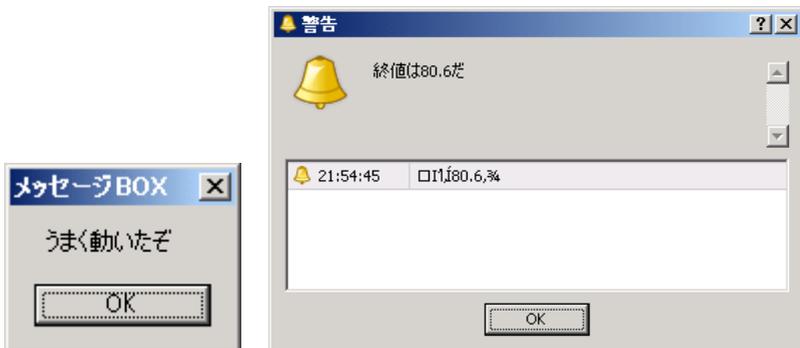
1. コード「my\_script\_01.mq4」

```
//確認する関数
//Comment,Print,Alert,MessageBox
//FileName=script_01.mq4
//
#include <WinUser32.mqh>           //MessageBox 用
#property show_confirm           //最初に1回止める

int start()
{
    Print("Hellow World\n");
    //Alert("終値は",Close[0],"だ");
    Alert("close=",Close[0],"・!!");
    Comment("現在のサーバー時間は
",TimeToStr(TimeCurrent()),"\n","現在の日本時間は
",TimeToStr(TimeLocal()));
    MessageBox("うまく動いたぞ","メッセージ
BOX",MB_OK);

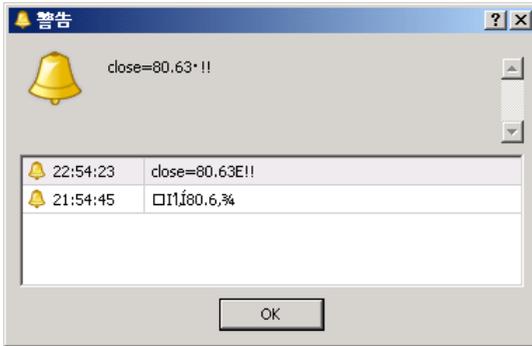
    return(0);
}
```

2. 動作画面



※「警告」中の文字が化けた？

< 要、調査 >



※「警告」では、枠内のデータ表示の日本語は化ける。



時間	メッセージ
2011.05.09 23:04:15	my_script_01 USDJPYFXF.M1: omit reason 0
2011.05.09 23:04:13	my_script_01 USDJPYFXF.M1: Alert: close=80.68* !!
2011.05.09 23:04:13	my_script_01 USDJPYFXF.M1: Hello World
2011.05.09 23:04:10	my_script_01 USDJPYFXF.M1: loaded successfully
2011.05.09 22:54:46	my_script_01 USDJPYFXF.M1: removed

○インディケータの動作確認

```
1. コード「my_draw_ema_01.mq4」
//
//      移動平均を2本引く：長短（EMA）・
//
//
//
#property show_inputs //設定時にインプット・タブを表示する

#property indicator_chart_window//チャート上に描画する
#property indicator_buffers 2 //インディケータは2個
#property indicator_color1 Magenta //長 ema はピンク
#property indicator_color2 Blue //単 ema は青

//指標バッファ
double Buf0[];
double Buf1[];

//外部パラメータ
extern int Long_Period = 13;
extern int Short_Period = 8;
extern int shift_ES = 0; //移動平均の表示「右シフト」数

//初期化・
int init()
{
//指標バッファ割り当て
SetIndexBuffer(0,Buf0);
SetIndexBuffer(1,Buf1);

//表示する指標数
IndicatorBuffers(2); //いくつ表示するかを指定

return(0);
}

//スタート関数
int start()
```

```
{  
    int limit = Bars-IndicatorCounted();  
  
    for(int i=limit-1;i>=0;i--)  
    {  
        Buf0[i]=iMA(NULL,0,Long_Period,0,MODE_EMA,PR  
ICE_CLOSE,i); //長 ema の計算  
        Buf1[i]=iMA(NULL,0,Short_Period,0,MODE_EMA,PR  
ICE_CLOSE,i); //単 ema の計算  
    }  
  
    return(0);  
}
```

## 2. 動作画面



○EA 動作確認 (バックテスト)

```
1. コード「my_cross_Up_Down_01.mq4」
//
//      移動平均のクロスにより売買する
//
//
//
#define magic_buy 1930276
#define magic_sell 1930277

#property show_inputs    //設定時にインプット・タブを表示する

//外部パラメータ
extern double Lots=0.1;
extern int Slippage=3;
//エントリー
extern int F_period=8;           //短期 ema
extern int S_period=13;         //長期 ema
//
int tiket=-1;                   //チケット No

//・
bool CrossUp(double line1, double line2)
{
    static bool last_1=0;        //false=0,true=1
    static bool last_2=0;
    static bool current_=0;     //

    if(line1>line2)current_+=1;  //
    if(line1<line2)current_-=1;

    if(current_ != last_1)
    {
        last_2=last_1;
        last_1= current_;
        if(last_2==-1 && last_1==+1)
        {
            return(true);
        }
    }
}
```

```
//ここに、コメントか、アラートを入れる
//Alert("Cross_Up");
//Comment("Cross_Up\n");
}
else
{
    return(false);
}
}
else
{
    return(false);
}
}
//
bool CrossDown(double line1, double line2)
{
    static bool last_1=0;    //false=0,true=1
    static bool last_2=0;
    static bool current_=0; //

    if(line1>line2)current_+=1;    //
    if(line1<line2)current_-=1;

    if(current_ != last_1)
    {
        last_2=last_1;
        last_1= current_;
        if(last_2==+1 && last_1==-1)
        {
            return(true);
            //ここに、コメントか、アラートを入れる
            //Alert("Cross_Down");
            //Comment("Cross_Down\n");
        }
        else
        {
            return(false);
        }
    }
}
```

```
    }
    else
    {
        return(false);
    }
}
//
void buy_0
{
    int
ticket=OrderSend(Symbol0,OP_BUY,Lots,Ask,3,0,0,NULL,ma
gic_buy,0,Green);
    //return(ticket);
}
//
void sell_0
{
    int
ticket=OrderSend(Symbol0,OP_SELL,Lots,Bid,3,0,0,NULL,m
agic_sell,0,Red);
    //return(ticket);
}

//スタート関数-----
int start()
{
    if(Bars<100)
    {
        Print("bars less than 100");
        return(0);
    }

    double
shortEma=iMA(NULL,0,8,0,MODE_EMA,PRICE_CLOSE,0);
    double
longEma=iMA(NULL,0,13,0,MODE_EMA,PRICE_CLOSE,0);

    int total=OrdersTotal();
```

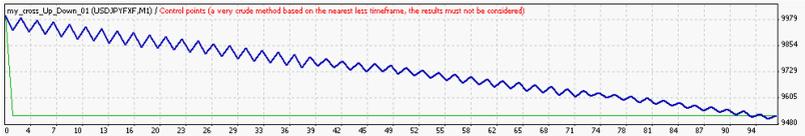
```
//-----  
  
if(CrossUp(shortEma,longEma)  
{  
    buy_0;  
}  
//  
if(CrossDown(shortEma,longEma)  
{  
    sell_0;  
}  
  
return(0);  
}
```

## 2. バックテストの結果画面

チャート；



資産カーブ；



※資産が減る！

リアルトレード（デモ）は、資産カーブを改良した EA で、実施のこと。

○関数化・・・原理確認用「超簡単コード」

1. 「1本」にまとめた場合

(1) コード

```
//
//
//     関数開発の確認用
//
//     現在の終値を表示する簡単なコードを「関数化」する
//
//     1 本版

void myComment(string com)
{
    Comment(com);
}
//スタート
int start()
{
    string com="Now! close="+Close[0];
    myComment(com);

    return(0);
}
```

(2) 結果画面

－ 1 : 設定

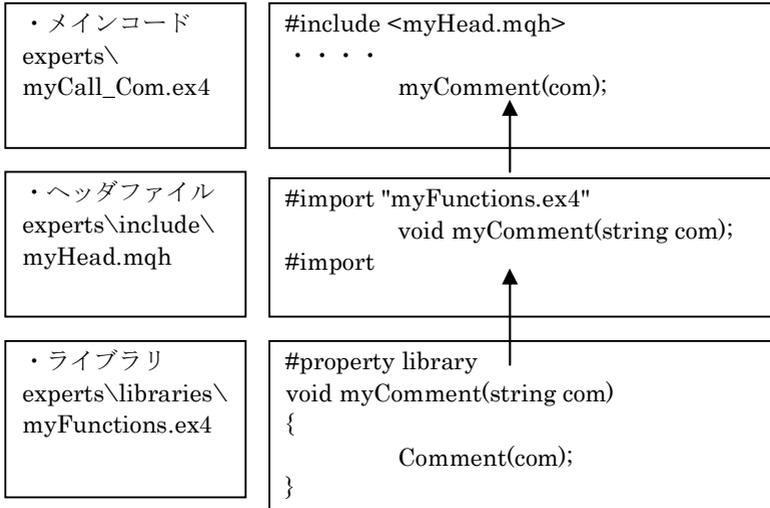


— 2 : 結果



## 2. 「分割版」の確認

### (1) ファイル構成



### (2) コード

#### ー 1. メインコード

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

```
#include <myHead.mqh>
```

```
//スタート
int start()
{
```

```
    string com="Now! close="+Close[0];
    myComment(com);
```

```
        return(0);  
    }
```

－ 2. ヘッダファイル

```
//  
//     『 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);  
#import
```

－ 3. ライブラリ (ファイル)

```
//  
//  
//     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);  
  
}
```

(3) 結果の画面



## ○CSV ファイルへのデータ出力 (EA)

### 1. コード

```
//
//      CSV ファイルへのデータ書き出し
//
//      ・ポイント;      上書モード [ FILE_WRITE ]
//                        追記モード
// [ FILE_READ|FILE_WRITE ]
//
//初期化
int init()
{
}

//スタート関数-----
int start()
{
    int handle;
    handle=FileOpen("myTest.csv",
FILE_CSV|FILE_READ|FILE_WRITE,",");
```

```
if(handle<1)
{
    Print("can't open file error-",GetLastError());
    return(-1);
}
if(handle>0)
{
    FileSeek(handle, 0, SEEK_END);
    //strText = strText + "\r\n";
    //FileWriteString(handle, strText,
StringLen(strText));

    //FileWrite(handle,TimeToStr(TimeMinute(TimeCurrent())),Open[0],High[0],Low[0],Close[0]);//①

    FileWrite(handle,TimeMinute(TimeCurrent()),Open[0],High[0],Low[0],Close[0]);//②
    FileClose(handle);
}
return(0);
}
//
int deinit()
{
}
```

## 2. 結果

### (1) 作成されるファイル



### (2) ファイルの内容 (書き出し結果)

1970.01.01 00:00	80.63	80.63	80.63	80.63
1970.01.01 00:00	80.63	80.64	80.63	80.64
1970.01.01 00:00	80.63	80.64	80.63	80.64
1970.01.01 00:00	80.63	80.64	80.63	80.64
1970.01.01 00:00	80.65	80.65	80.65	80.65
1970.01.01 00:00	80.65	80.65	80.65	80.65
1970.01.01 00:00	80.65	80.65	80.65	80.65
34	80.63	80.64	80.63	80.64
34	80.63	80.64	80.63	80.63
35	80.62	80.62	80.62	80.62
35	80.62	80.62	80.62	80.62
36	80.61	80.61	80.61	80.61
36	80.61	80.61	80.61	80.61
36	80.61	80.61	80.61	80.61

上段=コード①、下段=コード②を使用時

## ○TXT ファイルへのデータ出力 (EA)

## 1. コード

```

//
//      TXT ファイルへのデータ書き出し
//
//      ・ポイント;      上書モード [ FILE_WRITE ]
//                        追記モード
// [ FILE_READ | FILE_WRITE ]
//

//スタート関数-----
int start()
{
    int handle;
    handle=FileOpen("myTest.txt",
FILE_CSV | FILE_READ | FILE_WRITE, ",");
    if(handle<1)
    {
        Print("can't open file error-",GetLastError());
        return(-1);
    }
    if(handle>0)
    {
        FileSeek(handle, 0, SEEK_END);

        FileWrite(handle,TimeToStr(TimeCurrent()),Open[0],
High[0],Low[0],Close[0]);
        FileClose(handle);
        Comment("現在のサーバー時間は
",TimeToStr(TimeCurrent()),"\n","現在の日本時間は
",TimeToStr(TimeLocal()));
    }
    return(0);
}

```

## 2. 結果

### (1) 作成されるファイル



### (2) ファイルの内容 (書き出し結果)

2011.05.10 23:53,80.65,80.66,80.65,80.66  
2011.05.10 23:53,80.65,80.66,80.65,80.66  
2011.05.10 23:53,80.65,80.67,80.65,80.67  
2011.05.10 23:53,80.65,80.67,80.65,80.66  
2011.05.10 23:53,80.65,80.67,80.65,80.66  
2011.05.10 23:54,80.65,80.65,80.65,80.65  
2011.05.10 23:54,80.65,80.66,80.65,80.66  
2011.05.10 23:54,80.65,80.66,80.65,80.65  
2011.05.10 23:54,80.65,80.66,80.65,80.66  
2011.05.10 23:54,80.65,80.66,80.65,80.65  
2011.05.10 23:54,80.65,80.66,80.64,80.64  
2011.05.10 23:54,80.65,80.66,80.64,80.65

(3) チャート



以上