

# EasyLanguage语法

- 1.程序化交易平台（ MultiCharts、 TradeBlazer... ）应用技术支持
- 2.为客户实现交易策略提供：培训、策略编程咨询、技术答疑



联系我们：

电话：0571-83788082

0571-82820093

客服QQ：100590571

E-Mail: [quant@cnzsqh.com](mailto:quant@cnzsqh.com)

传真：0571-83788077

# EasyLanguage



- 1.基本关键字部分**
- 2.流程控制语句**
- 3.多图表数据调用**
- 4.全局变量**
- 5.趋势线画图**
- 6.部分实盘注意点分享**

# 基本关键字

<i>Data Word</i>	<i>Abbreviation</i>	<i>Description</i>
Open	O	First available price for the bar
High	H	Highest price within the bar
Low	L	Lowest price within the bar
Close	C	Last available price for the bar
Date	D	Date of the last trade within a bar
Time	T	Time of the last trade within a bar (in 24 hour format)
Volume	V	Total volume of trades within the bar
OpenInt	I	(Open Interest) Total number of open contracts

# 买卖指令



**Buy 买入开仓**



**Sell 卖出平仓**

**Sellshort 卖出开仓**



**Buytocover 买入平仓**

## Stop / limit的含義和區別

**Stop**表示下單價格的停止點，例如：

Buy next bar 3000 stop;表示價格3000以上才買；

Sellshort next bar 3000 stop;表示價格3000以下才賣；

Sell next bar 3000 stop;表示價格3000以下平多頭倉位，多用於止損；

Buytocover next bar 3000 stop;表示價格3000以上平空頭倉位，多用於止損；

**Limit**表示達到限制價格就限制下單，例如：

Buy next bar 3000 limit;表示價格超過3000就不買，即3000以下才買；

Sellshort next bar 3000 limit;表示價格低於3000就不賣，即3000以上才賣；

Sell next bar 3000 limit;表示價格3000以上平多頭倉位；

Buytocover next bar 3000 limit;表示價格3000以下平空頭倉位；

# 回溯调用

N bars ago数据的写法：

**XXX[N]**

例：

high of 5 bars ago ( **REF(HIGH,5)** )

↪ **high[5] 或者 h[5]**

# 流程控制语句

**1. IF 语句**

**2. Begin..end语句**

**3. Once语句**

**4. For循环**

**5. While循环**

**6. Switch...case语句**

# If语句

**If {conditions<sub>s</sub>} then {EL instruction} ;**

**If {conditions<sub>s</sub>} then begin  
    {EL instructions<sub>s</sub>};  
End;**



# If...then &Begin...end;

```
var: CountTheBuys(0) ;
```

```
if (Close>High[1]+Range[1] and Low>Low[1]) then begin  
    Buy next bar at Close stop ;  
    CountTheBuys = CountTheBuys + 1 ;  
end ;
```

# 忽略字

显示为红色字体的

用于阅读，在程序里面不起任何语句功能

A An At Based By Does From  
Is Of On Place Than The Was

# If...else...

If {conditions} then  
    {EL instructions}

else

    {EL instructions};

```
if Close > Close[1] then  
    Buy 20 shares next bar at market  
else  
    sellshort 10 shares next bar at market ;
```

# If...else...&begin...end;

if {conditions} then **begin**  
    {EL instructions};  
**end**

Else **begin**  
    {EL instructions};  
**end;**

```
if Close > High[1] then begin
    Value1 = 10 ;
    value2 = true ;
end
else begin
    Value1 = 20 ;
    value2 = false ;
end ;
```

# Once语句

**EL的程序是每个K棒执行一次，Once语句能使指定陈述式只执行一次**

**once {EL instruction };**

**Once begin  
    {EL instructions};  
End;**

# Once语句

**当第一次满足条件时，执行陈述式**

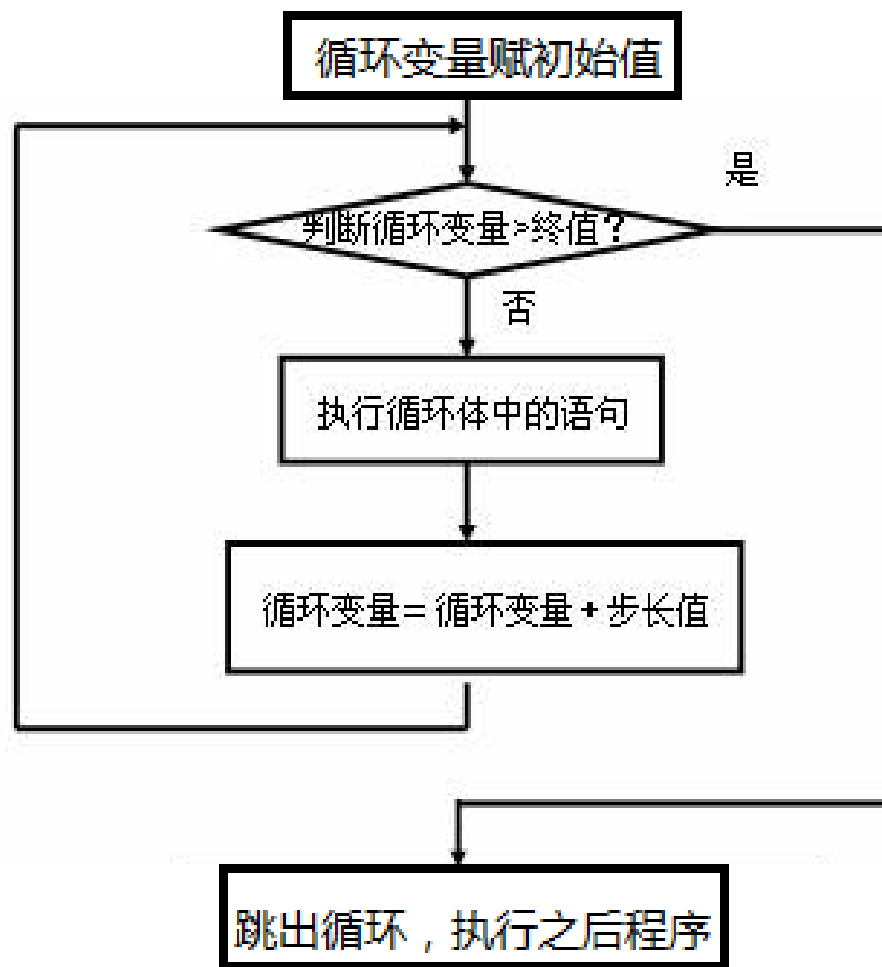
```
Once {conditions} begin  
    {EL instructions};  
End;
```

# Once语句

```
once Close<Open begin  
    value1=close;  
end;
```

```
if value1=0 and Close<Open then  
    value1=Close;
```

# For循环





# For循环

```
For {循环变量} = {起始值} to {终值} begin  
    { EL instructions } ;  
End;
```

```
For {循环变量} = {起始值} downto {终值} begin  
    { EL instructions };  
End;
```

# For循环

求5天价格之和：

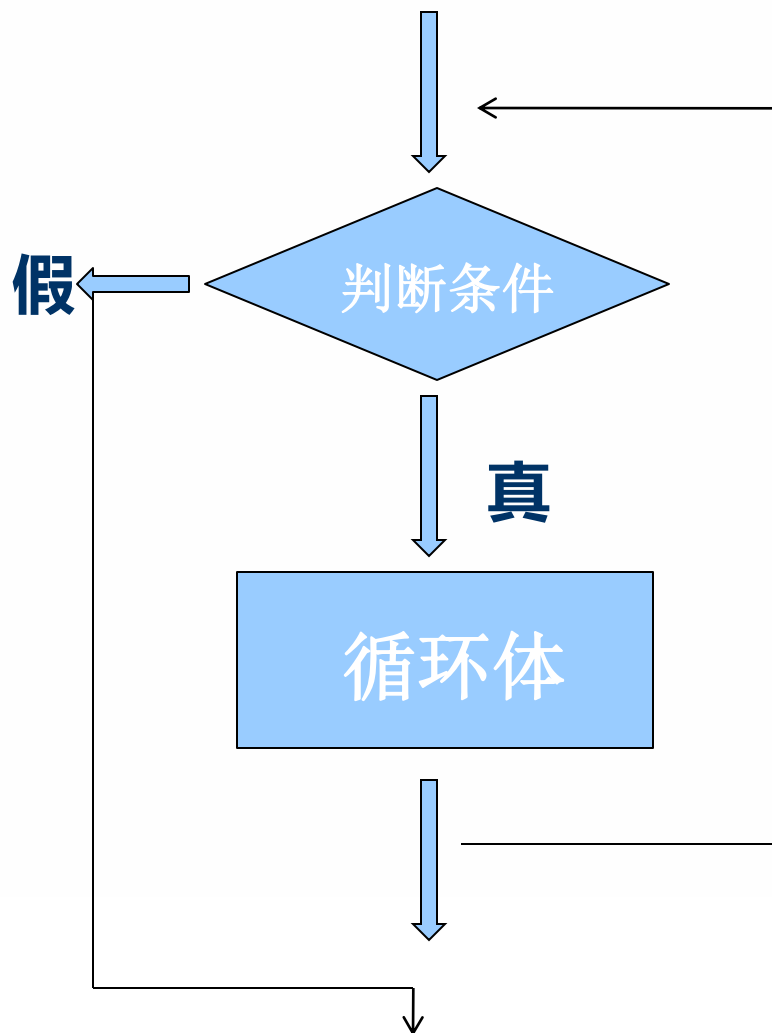
```
Inputs: price(close);  
vars: N(0), var0(0), Sum_5Days(0);  
  
for N= 0 to 4 begin  
    var0=var0+price[N];  
end;  
  
Sum_5Days=var0;
```

# for循环

## 中间需要终止循环的：

```
Inputs: price(close);  
vars: N(0), var0(0), Sum_5Days(0);  
  
for N= 0 to 4 begin  
    var0=var0+price[N];  
    if var0>10000 then N=5;  
end;  
  
Sum_5Days=var0;
```

# While循环



# While循环

```
While {conditions} begin  
    {EL instructions};  
end ;
```

# While循环

求5天价格之和：

```
Inputs: price(close);  
vars: N(0), Sum_5Days(0);  
  
while N<=4 begin  
    Sum_5Days=Sum_5Days+price[N];  
    N=N+1;  
end;
```

# While循环

## 防止形成无限循环：

```
vars: N(0), SumOfPrice(0);  
  
while Vol[N]<100000 begin  
    SumOfPrice=SumOfPrice+Close[N];  
    N=N+1;  
    if N>60 then Vol[N]=999999;  
end;
```

# Switch...case

对同一个变量或者表达式进行多重判断时，可以利用 Switch...case语句，简化于繁琐的if...else多重嵌套结构

语法结构：

```
Switch(变量或者表达式) begin
    case {数值或者字符串}: { EL instruction };
    (case...case...case...)
    default:{EL instruction};
End;
```



# Switch...case

**例：**

**图表是分钟线：画 5MA**

**图表是日线：画 20MA**

**图表是周线：画 50MA**

# Switch...case

## 用if...else结构：

```
if bartype=1 then begin  
    Plot1(Average(close,5));  
end else if bartype=2 then begin  
    Plot1(Average(close,20));  
end else if bartype=3 then begin  
    Plot1(Average(close,50));  
end;
```

# Switch...case

## 用Switch...case结构:

```
switch (bartype) begin
    case 1 :Plot1(Average(close,5));
    case 2 :Plot1(Average(close,20));
    case 3 :Plot1(Average(close,50));
end;
```

# 多图表数据调用

- 1.不同商品之间数据
- 2.同种商品不同时间周期的数据
- 3.需要用到图表外数据的其他情形

语法：

{数值变量} of dataN (N值最大到99)

# Data1...dataN

分析豆油的同时，需要同时比较大豆走势：

Data1:豆油

```
value1=Close of data1;
```

Data2:大豆

```
value2=Close of data2;
```

```
if value2>value2[1] then buy this bar on close;
```

（跨时间周期调用类似，order只能下在data1商品上）

# dataN的注意事项

```
value1=RSI(close of data2);  
value2=RSI(close) of data2;
```

- 1.函数仅有一个参数资料数列时，二种写法效果相同
- 2.value1直接针对变量指定资料数列
- 3.value2把变量对应的资料数列指定为新资料数列

**若函数里面变量有指定资料数列，外加指定数列变换时，仍以变量自身设定数列为主**

```
function(High,Low) of data2
```

```
Function(High of data2,Low of data2)
```

```
function(High of data1,Low) of data2
```

```
Function(High of data1,Low of data2 )
```

```
function(High of data1,Low of data2) of data3
```

```
Function(High of data1 ,Low of data2)
```

**跨图表数据之间联系：在同一工作窗口中打开多个图表，  
用全局变量串联之间某些变量**

**GVSetNamedDouble( “变量名” ,设置值)**

**GVGetNamedDouble ( “变量名” ,缺失值)**

**GVSetNamedInt ( “变量名” ,设置值)**

**GVGetNamedInt ( “变量名” ,缺失值)**





## 小麦和玉米，任何其一有持仓，则不再开仓

```
vars:WS_Pos(0);  
if marketposition=0 then  
    GVSetNamedDouble("C_position",1);
```

玉米策略

```
WS_Pos=GVGetNamedDouble("WS_Position",0);  
if marketposition=0 and WS_Pos=1 {and {condition}} then  
    buy("buy_Corn") next bar {price};
```

```
vars:Corn_Pos(0);  
if marketposition=0 then  
    GVSetNamedDouble("WS_position",1);
```

小麦策略

```
Corn_Pos=GVGetNamedDouble("C_position",0);  
if marketposition=0 and Corn_Pos=1 {and {condition}} then  
    buy("buy_WS")next bar {price};
```

# 趋势线画图

支持的5种画线类型：

线条(line)

柱状图 ( Histogram )

点(Point)

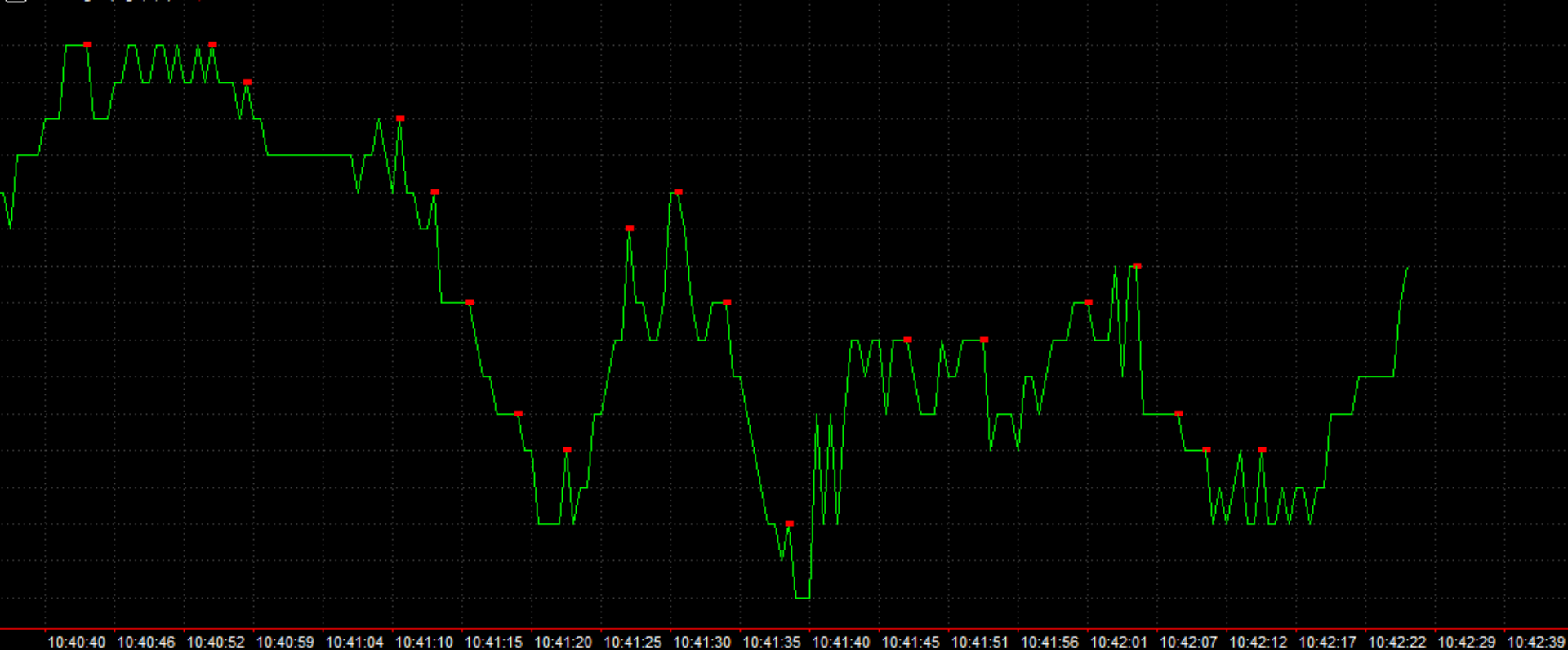
十字(Cross)

K棒 ( Bar High、 Bar low、 left tick、 right tick )

# Right tick



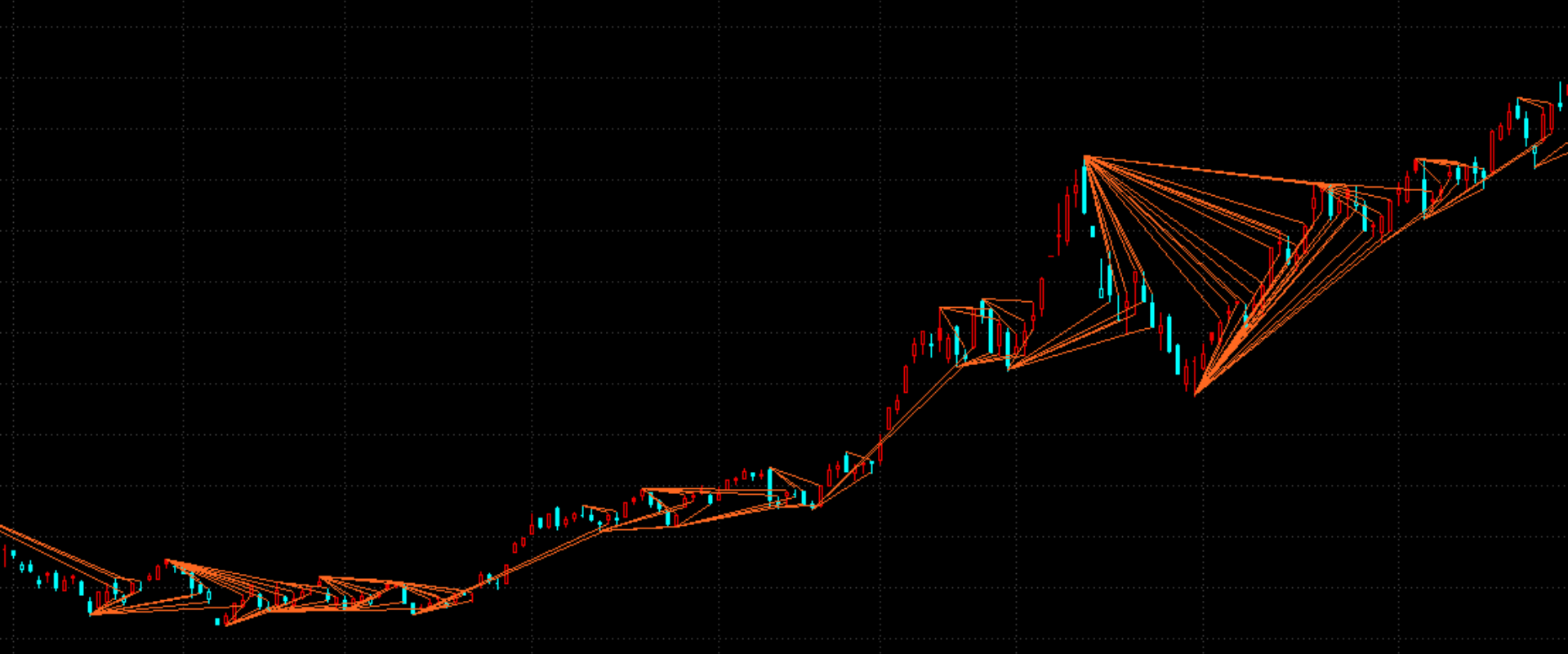
SA CTP.SHFE.ru.201105 天胶1105 1 Tick Bar 成交价 sfe TOUCHANCE 期货 2011/2/21 10:42:25 买=40,430 卖=40,435 开=40,690 高=40,760 低=40,085 收=40,435 最新=40,435 -690 -1.68% 成交里=253,328  
X Pivot High (high,3,3) 40,410



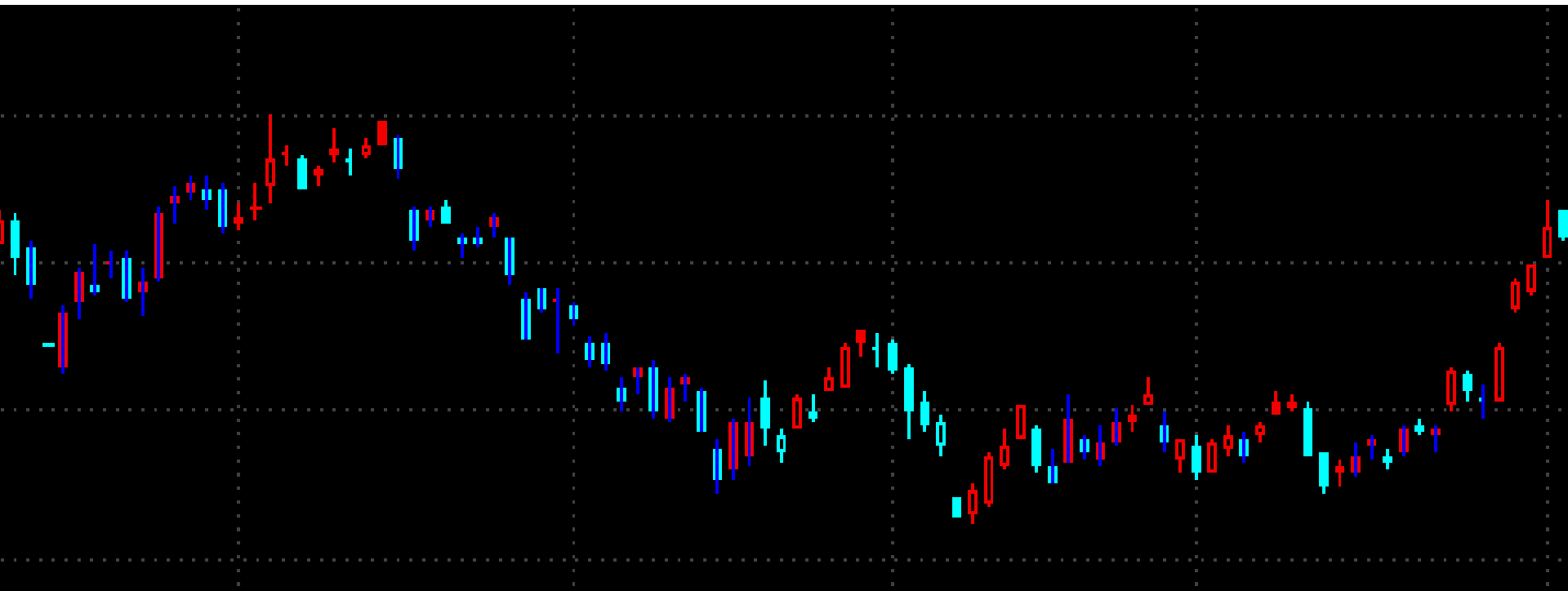
# line



SA CTP.SHFE.ru.201105 天胶1105 1日 R 成交价 sfe TOUCHANCE 期货 2011/2/21 10:49:08 买=40,345 卖=40,350 开=40,690 高=40,760 低=40,085 收=40,350 最新=40,350 -775 -1.88% 成交量=267,426



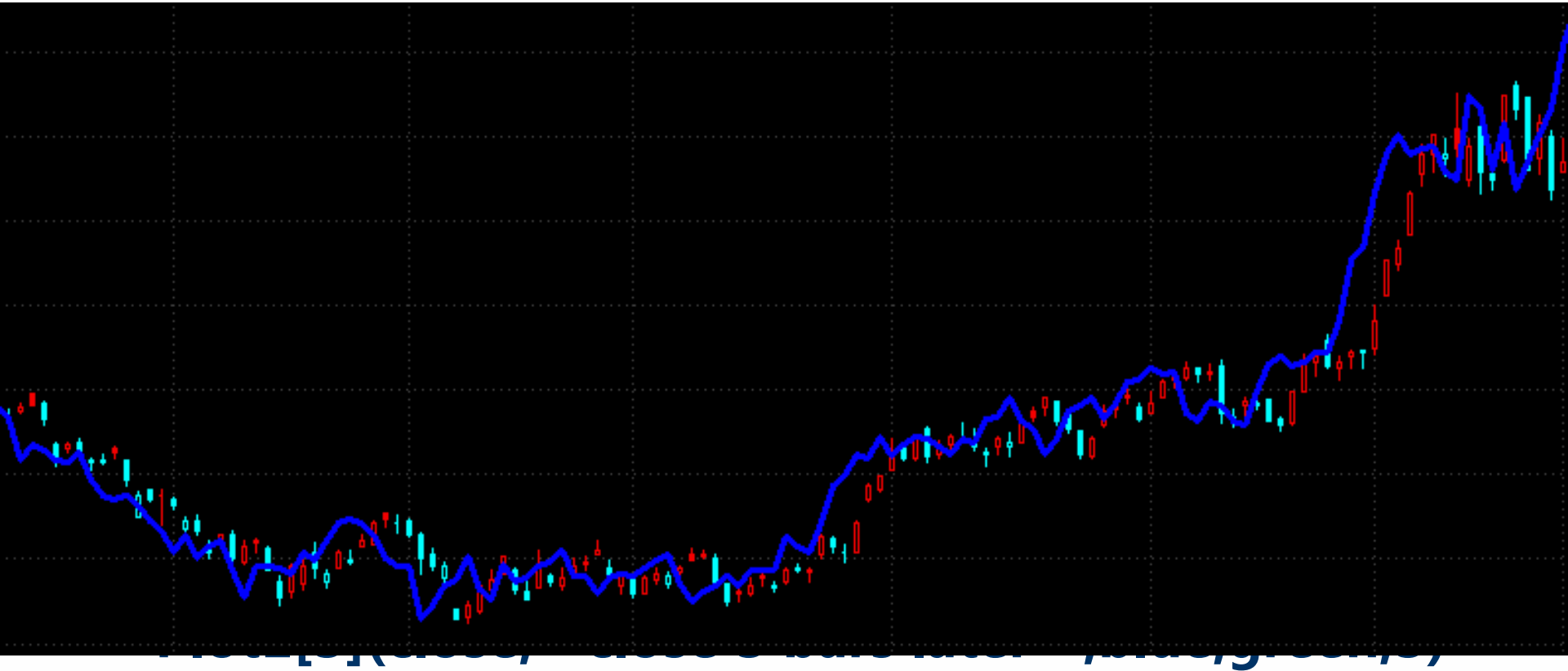
# Bar high & bar low



# cross



# plot





**指标或者信号中，绘制一条指定起点和终点的趋势线;并  
传回画线编号的数值**

**语法：**

**Tl\_new(起始日期，起始时间，起始价格，终点日期，终  
点时间，终点价格)**

**日期格式YYYYMMDD 时间格式HHMM**

# Tl\_new



**绘制从昨天开盘9点00分时开盘价开始，到今天15点00时的close价格结束的趋势线**

```
tl_new(date[1], 900, open(1), date, 1500, close);
```

# Text\_new

**在图表上指定位置标示一个特定文字的显示**

**语法：**

**Text\_new(日期，时间，价格，“显示文字”)**

# Text\_new

例：

当连续三根K线的开盘价都持续升高，则在K线最高点的位置标示 “Apex”

```
if Open>Open[1] and Open[1]>Open[2] then  
    text_new(date,time,high,"apex");
```

# 实盘注意点

Order写法的格式固定：

(buy...) this bar **at** close;

(buy...) next bar **at** XXXX stop (or limit);

# 条件满足就买

## 1.开启bar内交易：

**[IntrabarOrderGeneration =true]**

**可以结合barstatus**

## 2.多图表，结合全局变量，大周期条件判断，小周期发order

# 快10秒下单

```
vars:
    intrabarpersist waitingForBuy(false),
    intrabarpersist mySec(0),
    intrabarpersist myEntrySec(0);

if barstatus=0 then begin
    mySec = currenttime_s;
    myEntrySec = currenttime_s+barinterval*60-10;
end else begin
    mySec = mySec[1];
    myEntrySec = myEntrySec[1];
end;

if {your entry condition} then begin
    waitingForBuy = true;
end;

if waitingForBuy and currenttime_s>=myEntrySec then begin
    buy next bar at open;
    waitingForBuy =false;
end;
```

# 多时间框架

用data1、data2的时候，涉及到一些如求highest的函数，如果data1和data2的周期大小不一样，那要注意时间截止的周期问题：

Data1: 5 mins   data2: 20 mins

Highest(h,4) of data2



**如有具体的程序问题，欢迎联系我们！**

**—— Dannie**