
1. 예스랭귀지란?	4
2. 예스랭귀지 편집기	4
1) 예스 랭귀지 편집기 실행	4
2) 예스랭귀지 편집기 기능설명	5
가) 도구모음	6
나) 사전	9
다) 파일관리자	10
라) 수식편집창	12
마) 오류 확인창	13
바) 디버깅창	14
3) 수식 작성하기	14
가) 새로 만들기	15
나) 시스템 새로 작성	16
다) 시스템 편집창	18
라) 작성	18
마) 검증	20
바) 수식 적용	21
3. 예스랭귀지 문법	23
1) 기본문법	23
가) 수식은 위에서 아래로 읽어 내려 간다	23
나) 첫 봉부터 마지막 봉으로 실행	25
다) 대소문자를 구별을 하지 않는다	25
라) 하나의 단어 내에서 띄어쓰기나 줄바꿈은 허용되지 않는다	26
마) 문장의 마침은 세미콜론(:)을 찍는다	26
2) 문장 구조	26
3) 각종 연산자	27
가) 산술 연산자	27
나) 관계연산자	28
다) 논리연산자	28
라) 할당연산자	29
마) 구둣점 및 괄호	29
바) 연산자 우선 순위	31
4) 데이터 및 데이터 정보	31
5) 변수	35
가) 변수란?	35
나) 외부변수	36

다) 내부변수.....	37
라) 내장변수.....	40
마) 변수 이용의 장점.....	40
6) 제어문.....	41
가) if 문.....	41
① if 조건 then 문.....	41
② If 조건 then 그룹 문.....	42
③ if 조건 then 중첩문.....	42
④ if 조건 then 실행1 else 실행2.....	43
⑤ if 조건 then 실행1 else if 실행2 else 실행3 문.....	44
나) 반복 처리문.....	45
① For문.....	45
② While 문.....	49
7) 함수 및 예약어.....	50
가) 주문함수.....	50
① 신호명.....	51
② 신호타입과 조건가격.....	51
③ 진입명.....	54
④ 수량.....	54
⑤ 수량옵션.....	54
나) 주문예약어.....	55
다) 수학함수.....	55
라) 분석함수.....	56
마) 출력함수.....	61
바) 추세선출력함수.....	61
사) 텍스트출력함수.....	62
아) 박스출력함수.....	63
자) 타원출력함수.....	63
차) 그리드출력함수.....	64
카) 잔고함수.....	64
타) 포지션 함수.....	64
파) 전략 성과 함수.....	65
하) 종목검색함수.....	66
9) 사용자함수(외장함수)로 제공되는 기본함수.....	67
9) 이전 값 참조.....	70
4 예스랭귀지 활용.....	71

1) 배열.....	71
2) 디버깅.....	75
3) 데이터 추출.....	78
4) 타종목 참조(Multi Data).....	80
가) 차트 구성.....	81
나) 타종목 참조 수식 작성.....	82
다) 타종목/타주기 유의사항.....	84
5) 사용자함수 만들기.....	89
6) 신호체계.....	94
7) 과거 데이터 움직임 가설.....	96
8) 예비신호.....	98
9) 실전매매와 시뮬레이션의 차이.....	99
가. 강제청산 중 최대수익대비하락(SetStopTrailing) 청산 사용.....	99
나. SetStopEndOfDay.....	100
다. 타종목참조.....	101
라. 일간갭보정 차트 이용.....	102
라. 틱차트 이용시 실시간 매매신호와 재조회 매매신호의 불일치 위험.....	102
마. 기타.....	102
5. 함수설명.....	104
1) 주문함수.....	104
2) 주문예약어.....	112
3) 수학함수.....	113
4) 분석함수.....	120
5) 출력함수.....	161
6) 추세선출력함수.....	167
7) 텍스트출력함수.....	176
8) 박스출력함수.....	181
9) 타원출력함수.....	192
10) 그리드출력함수.....	201
11) 잔고함수.....	201
12) 포지션함수.....	207
13) 전략성과함수.....	221
14) 사용자함수(기본 내장).....	224
15) 선언.....	249
16) 색상예약어.....	252
17) 예약어.....	252

1. 예스랭귀지란?

예스트레이더에는 사용자가 구상한 전략을 시스템화하여 구현될 수 있도록 예스랭귀지를 제공하고 있습니다. 예스랭귀지는 개방형 프로그램 언어로 프로그램에서 제공하는 전략만을 사용할 수 있는 블랙박스 형태의 시스템 트레이딩 툴의 한계를 극복하여 사용자가 무한대로 전략을 확장하여 실전매매에서 이용할 수 있습니다.

원하는 전략을 구현하기 위해서는 랭귀지에 대한 공부와 필요로 하게 됩니다. 예스랭귀지는 기본적으로 시스템언어를 다루어 보지 않은 일반 사용자도 쉽게 배워 사용할 수 있는 언어이며 프로그래밍 언어나 엑셀에서 매크로를 다루어 본 경험이 있다면 조금 더 수월하게 배울 수 있습니다.

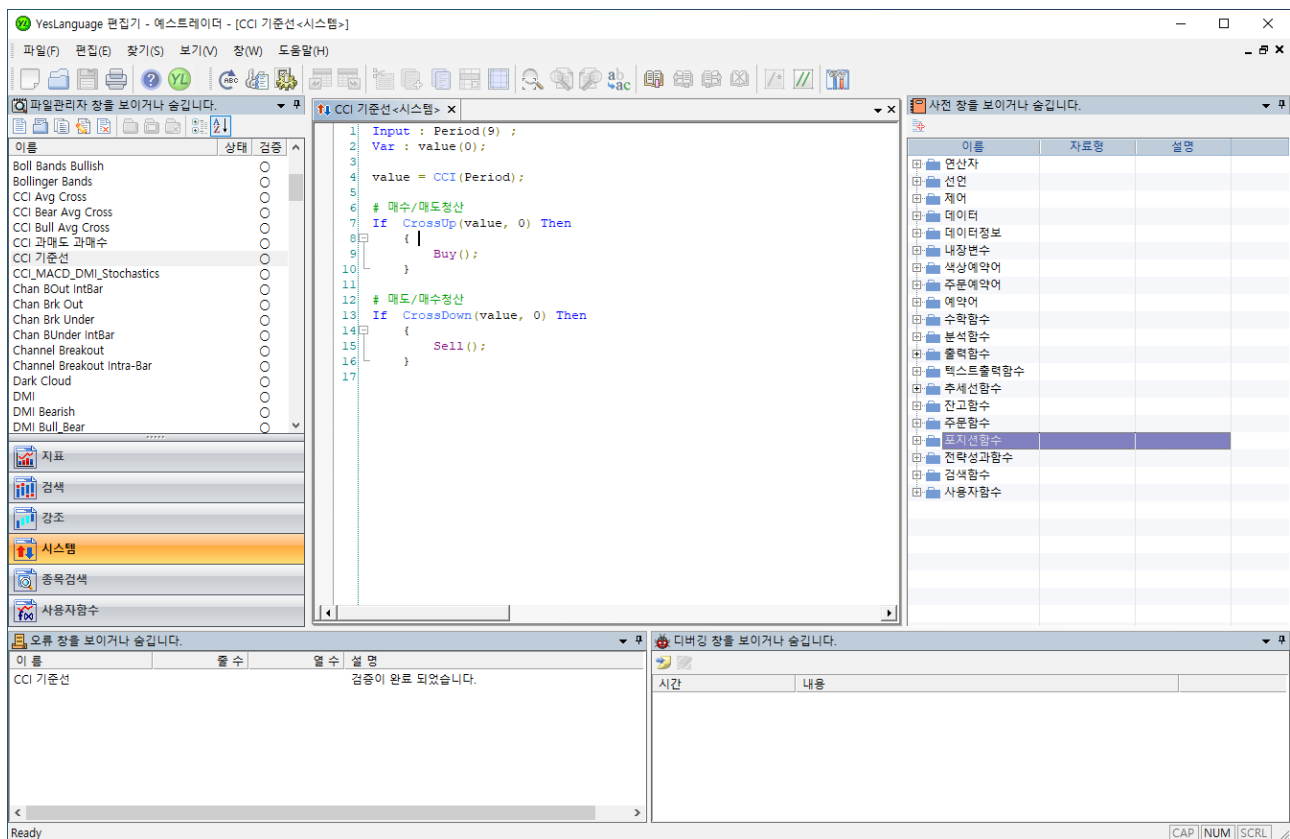
- ※ 예스랭귀지는 차트기반의 언어로 문법, 함수,데이터,체계가 제공되는 모든 프로그램에서 동일합니다. 다만 해외선물과 같은 특정 상품은 매수호가잔량(bids), 매도호가잔량(asks), 상승형거래량(Upvol),하락형거래량(DownVol), 상승형거래건수(UpTicks),하락형거래건수(DownTicks)등이 제공되지 않을 수 있으므로 수식작성시 필요한 데이터가 값을 제공하는지 확인하시기 바랍니다.

2. 예스랭귀지 편집기

1) 예스 랭귀지 편집기 실행

예스랭귀지로 수식을 작성하기 위해서는 먼저 예스랭귀지 편집기를 실행시켜야 합니다.

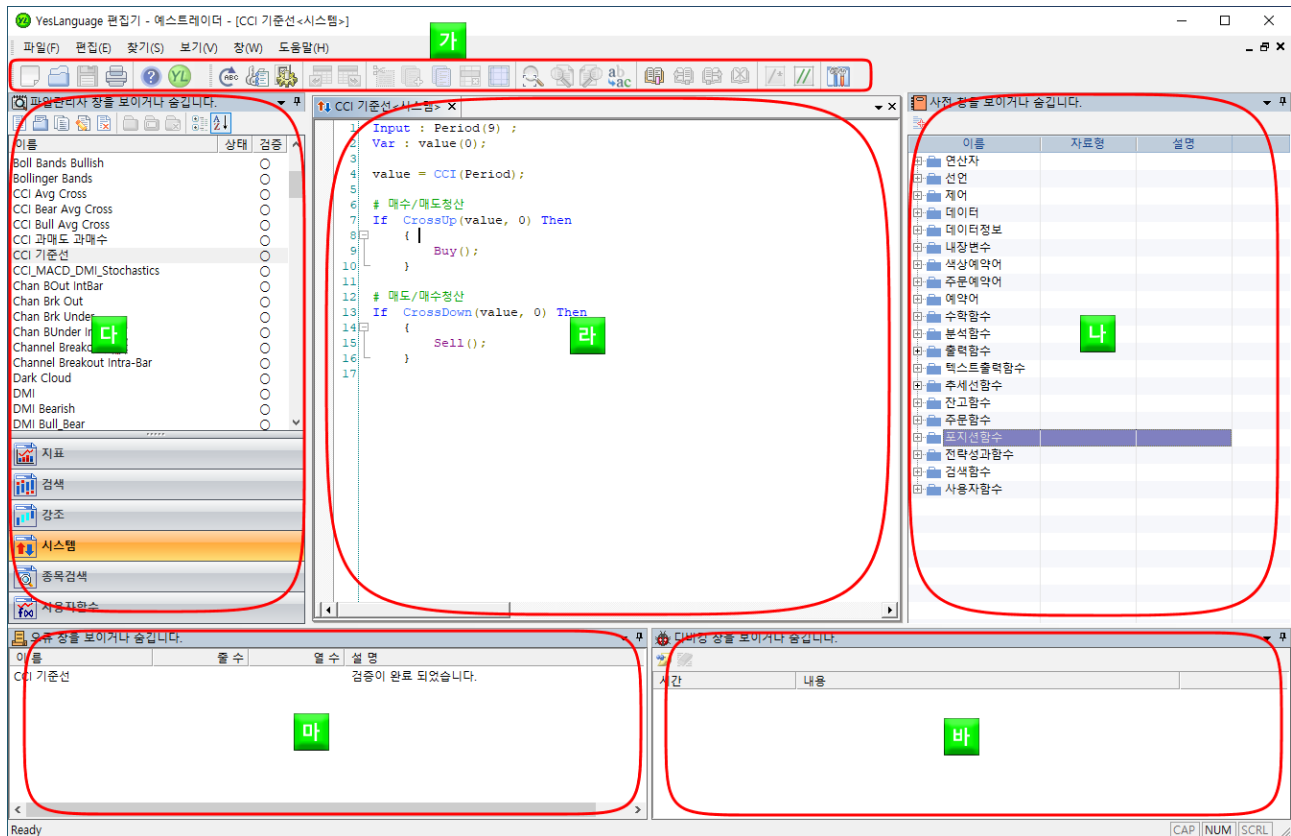
프로그램 상단의 [시스템트레이딩]메뉴에서 [6109] YesLanguage 편집기를 선택하면 랭귀지 편집기가 실행됩니다.



예스랭귀지 편집기는 사전, 파일관리자, 오류창, 디버깅창, 편집창등 5개의 화면으로 구성되어 있습니다. 각 화면의 배치는 사용자의 기호에 따라 자유롭게 구성할 수 있습니다.

2) 예스랭귀지 편집기 기능설명

YesLanguage는 시스템 트레이딩 언어로서 지표, 시스템, 검색식 등을 작성할 수 있는 강력한 도구입니다. YesLanguage는 언어로서의 제약을 완전히 없앤 개방형 프로그래밍 언어입니다. 많은 함수가 내장되어 제공되므로 빠르고 쉽게 매매전략을 작성할 수 있습니다.

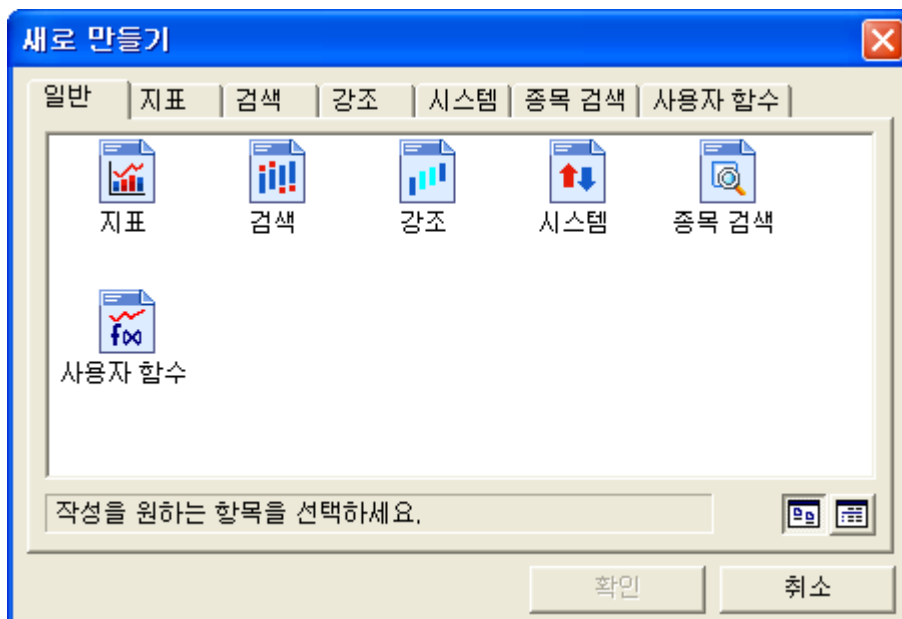



가) 도구모음

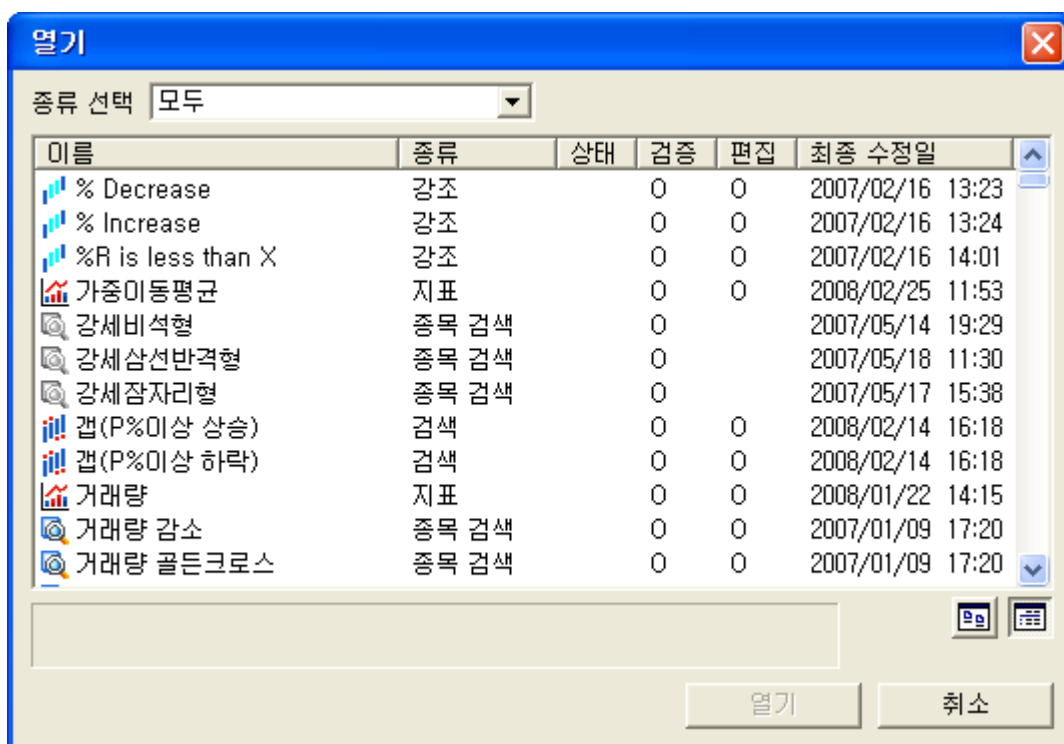
기본도구





새로만들기 : '지표/검색/강조/시스템/종목검색/사용자함수' 중 새로 만들 항목을 선택



 열기 : 지표/검색/강조/시스템/종목검색/사용자함수'를 열 때 사용.



 저장 : 주식 편집창의 내용을 저장.

 프린트 : 주식을 인쇄.



도움말 : 예스랭귀지 웹도움말 실행



정보 : 예스랭귀지 버전정보를 확인

편집도구



검증 : 수식의 문법 검증. 검증된 내용은 오류화면에 출력.



암호설정 : 수식 보안설정



속성 실행 : 수식의 속성(설정)창 실행



되돌리기 : 실행 취소 전의 상태로 복구.



다시하기 : 실행 취소 후의 상태로 복구.



자르기 : 수식 내 선택된 내용을 잘라내기.



복사 : 수식 내 선택된 내용을 복사.



붙여넣기 : 잘라낸 내용 및 복사한 내용을 붙여넣기.



삭제 : 선택된 내용을 삭제





전체선택 : 수식의 모든 내용을 선택




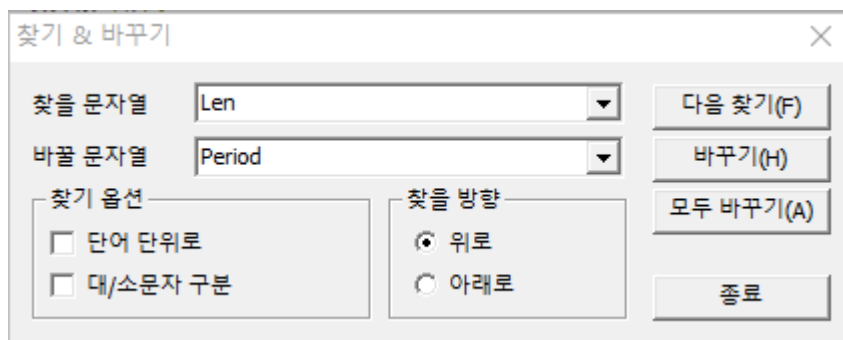
찾기 : 찾기 창을 생성하며, 수식 편집창 내에서 해당 내용을 검색.





 다음찾기 : 찾기 기능 실행 후, 커서 다음 내용 중, 동일 내용 있는 경우, 다음 내용을 검색.


 이전찾기 : 찾기 기능 실행 후, 커서 다음 내용 중, 동일 내용 있는 경우, 이전 내용을 검색.


 바꾸기 : 찾기 기능 실행 후, 커서 다음 내용 중, 동일 내용 있는 경우, 다음 내용을 변경.





 책갈피 상태전환 : 커서가 있는 열에 책갈피를 생성.


 이전 책갈피로 : 현재 커서 이전의 책갈피로 이동.

 다음 책갈피로 : 현재 커서 다음의 책갈피로 이동.

 모든 책갈피 삭제 : 지정된 모든 책갈피를 삭제.

 여러 주석처리/해제 : 수식 편집창의 여러 줄의 주석 처리 및 해제.

 한줄 주석처리/해제 : 수식 편집창의 한 줄 주석 처리 및 해제.

 편집창 설정

나) 사전

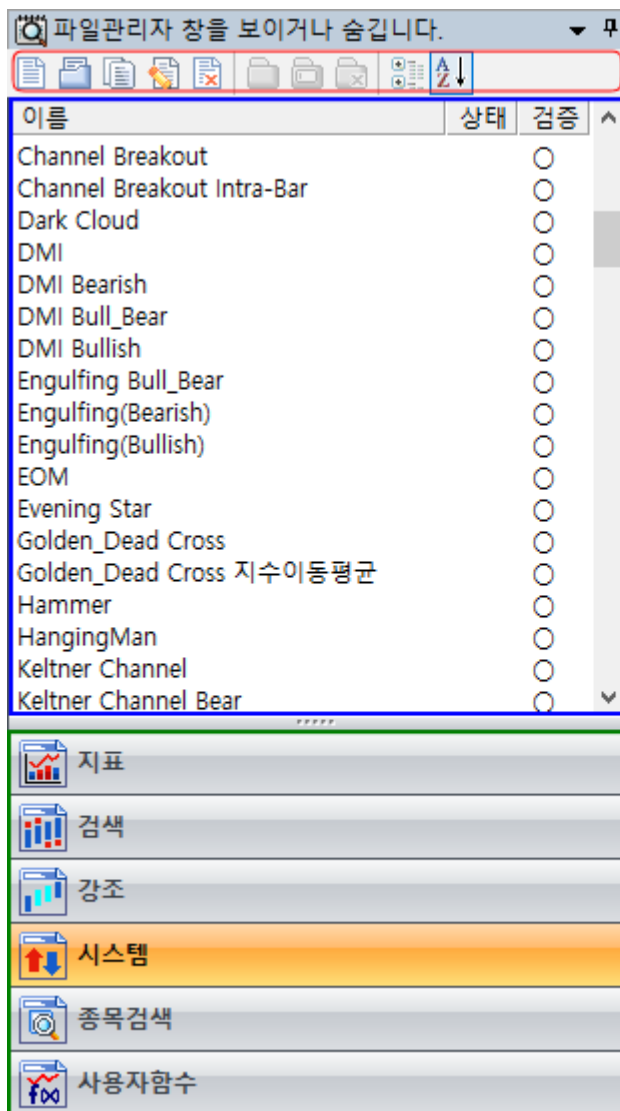
함수와 예약어 등의 사용법을 확인할 수 있는 화면입니다.

카테고리별로 함수나 예약어를 확인할 수 있고 찾기탭에서 직접 텍스트를 입력하여 찾기 가능합니다.

사전 창을 보이거나 숨깁니다. ▼ 무			
이름	자료형	설명	
연산자			
선언			
제어			
데이터			
데이터정보			
내장변수			
색상예약어			
주문예약어			
예약어			
수학함수			
분석함수			
출력함수			
텍스트출력함수			
추세선함수			
잔고함수			
주문함수			
포지션함수			
전략성과함수			
검색함수			
사용자함수			

다) 파일관리자

수식 파일들을 관리할 수 있는 화면입니다.



파일관리도구

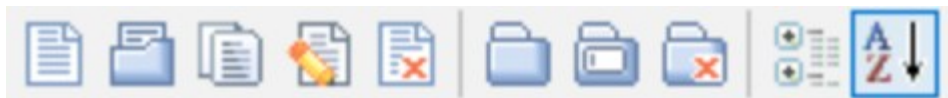
수식리스트

수식종류탭

각 수식은 종류별로 탭으로 분리되어 있습니다.

하단 영역에서 수식종류를 지정하면 상단에 수식들이 리스트 됩니다.

파일관리도구



각 버튼을 클릭하여 수식을 생성 및 열기, 복사, 이름변경, 삭제를 할 수 있습니다.



새로작성 : 선택된 수식 종류의 "새로 작성"화면이 생성.



수식열기 : 선택한 수식의 편집창을 오픈.



수식복사 : 선택한 수식을 복사.



이름변경 : 선택한 수식의 이름을 변경.



수식삭제 : 선택한 수식을 삭제.



새그룹생성 : 선택한 수식탭에 하위로 새로운 그룹을 생성.



그룹변경 : 선택한 그룹의 이름을 변경.



그룹삭제: 선택한 그룹을 삭제.



리스트 그룹별보기



리스트 이름순보기

라) 수식편집창

수식을 작성하는 화면입니다.

```

1 input : 투자금액(100000000), Startdate(20120102);
2 var : investment(0), Start(false), Base(0), Initial(0), BuyAmount(0);
3
4 if MarketPosition == 0 and (sdate == Startdate and TotalTrades == 0) Then{
5     investment = 투자금액+NetProfit;
6     Initial = floor(investment*0.200);
7     BuyAmount = floor(investment*0.025);
8     buy("b", OnClose, def, floor((Initial+BuyAmount)/C));
9     Start = true;
10    Base = C;
11 }
12
13 if MarketPosition == 0 and (MarketPosition[1] != 0 and TotalTrades > 0) Then{
14     investment = 투자금액+NetProfit;
15     Initial = floor(investment*0.200);
16     BuyAmount = floor(investment*0.025);
17     buy("bb", OnClose, def, floor((Initial+BuyAmount)/C));
18     Start = true;
19     Base = ExitPrice(1);
20 }
21
22 if MarketPosition == 1 Then
23 {
24     if CurrentContracts > CurrentContracts[1] and LatestEntryName(0) == "b11" Then
25         BuyAmount = BuyAmount*0.95;
26     if CurrentContracts > CurrentContracts[1] and LatestEntryName(0) == "b12" Then
27         BuyAmount = BuyAmount*1.05;
28     if CurrentContracts > CurrentContracts[1] and LatestEntryName(0) == "b21" Then
29         BuyAmount = BuyAmount*0.98;
30     if CurrentContracts > CurrentContracts[1] and LatestEntryName(0) == "b22" Then
31         BuyAmount = BuyAmount*1.02;
32
33     if LatestEntryPrice(0)*1.015 > Base Then
34         buy("b11", AtStop, LatestEntryPrice(0)*1.015, floor((BuyAmount*0.95)/C));

```

마) 오류 확인창

문법 검증 결과를 표시하는 창이며 문법상 오류가 없으면 [검증이 완료 되었습니다.]라는 메시지가 나타납니다. 문법상 오류가 있으면 오류설명과 함께 줄수와 열수를 표시가 되고 해당 오류메시지를 클릭하면 해당 위치로 커서가 이동하게 됩니다.

오류 창을 보이거나 숨깁니다.			
이름	줄 수	열 수	설명
CCI 기준선			검증이 완료 되었습니다.

바) 디버깅창

선택한 주식의 디버깅 된 내역을 조회할 수 있습니다.

디버깅 창에 내용을 표시하기 위해서는 MessageLog라는 함수를 이용해야 한다.

디버깅 창을 보이거나 숨깁니다.		
시간	내용	
2020-06-26 17:00:00	100000000 -21557 4000000	
2020-06-29 17:00:00	100000000 -58865 4000000	
2020-06-30 17:00:00	100000000 -39186 4000000	
2020-07-01 17:00:00	100000000 -37956 4000000	
2020-07-02 17:00:00	100000000 -8848 4000000	
2020-07-03 17:00:00	100000000 10421 4000000	
2020-07-06 17:00:00	100000000 46499 4000000	
2020-07-07 17:00:00	100000000 17801 4000000	

3) 주식 작성하기

예스랭귀지는 목적별로 6가지의 편집창이 존재하고 사용자가 구현하고자 하는 목적에 맞는 편집창을 열어 수식을 작성하고 사용하여야 합니다.

매매 아이디어를 구현하고 싶다	시스템 편집창
차트에 선이나 막대 등으로 값을 표시하고 싶다	지표 편집창
특정 조건에 만족하는 종목을 찾고 싶다	종목검색 편집창
특정 조건이 만족하는 봉에 점으로 표시하고 싶다	검색 편집창
특정 조건이 만족하는 봉에 색을 입히고 싶다	강조 편집창
함수를 만들고 싶다	사용자함수 편집창

편집창의 종류가 다르다고 하여 문법이 다른 것은 아닙니다.

다만 buy와 같은 주문함수는 시스템 편집창에서만 사용할 수 있고 PlotPaintBar는 강조 편집창에서만 사용이 가능한 것과 같이 일부 함수의 경우에는 특정 편집창에서만 사용이 가능하므로 해당 부분만 유의하여 사용하면 됩니다.

주문함수	시스템 편집창
주문예약어	시스템 편집창
포지션 함수	시스템 편집창
전략성과 함수	시스템 편집창
지표출력 함수	지표 편집창, 검색 편집창
종목검색 함수	종목검색 편집창
강조출력 함수	강조 편집창

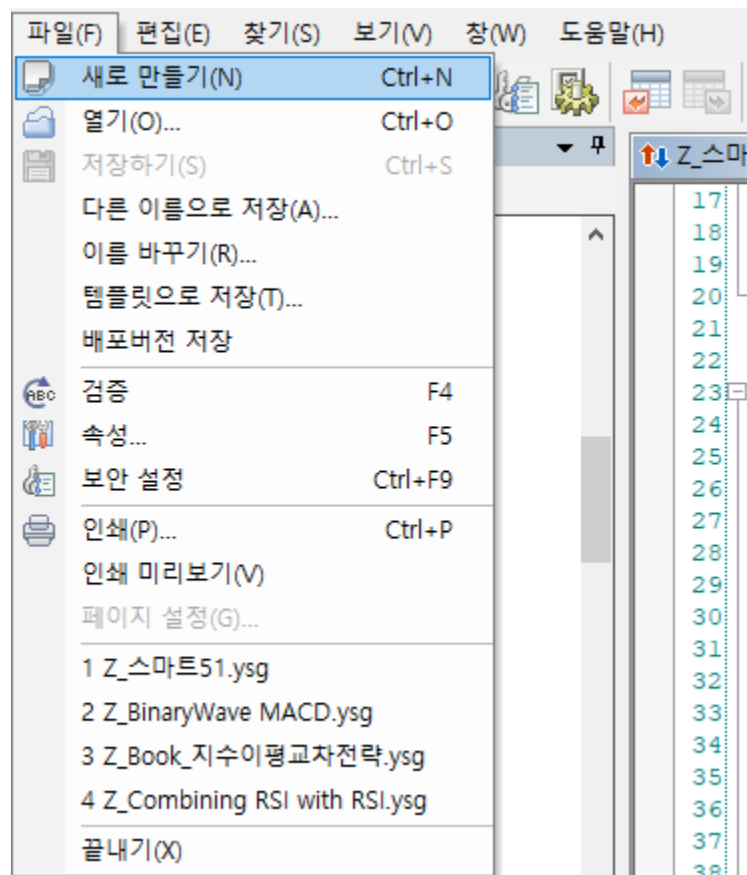
하나의 수식을 만들어 차트에 적용하기 까지는 모두 동일하게 아래의 과정으로 거치게 됩니다.

메뉴 [파일] → 새로만들기 → 일반탭에서 편집창 종류 아이콘 선택
→ 이름과 설명 입력 → 편집창 생성 → 수식작성 → 검증 → 적용

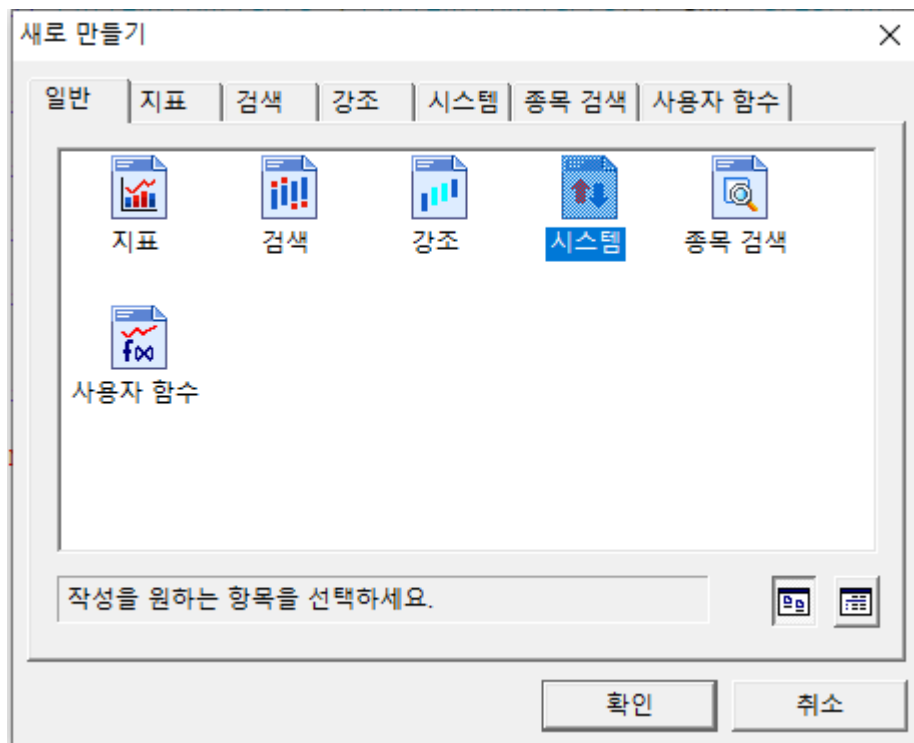
시스템 수식을 예로 수식작성과정을 설명해 보도록 하겠습니다.

가) 새로 만들기

예스랭귀지 상단 메뉴 중 [파일]을 선택하고 목록에서 [새로 만들기]를 선택합니다.

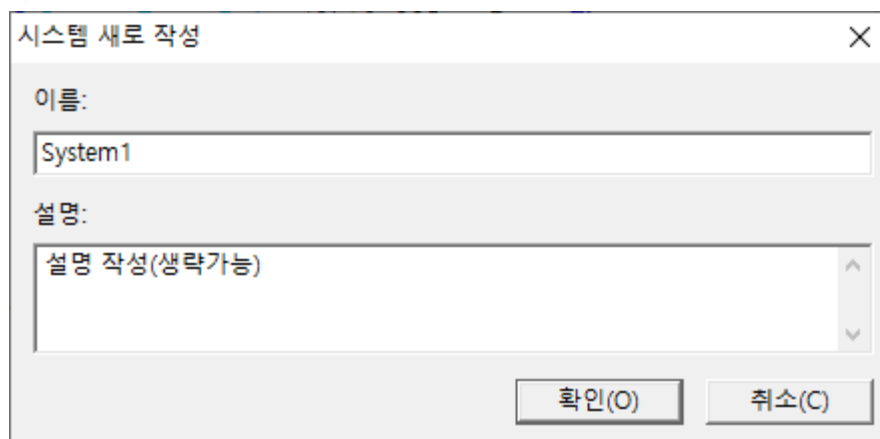


[새로 만들기] 화면은 편집창의 종류를 선택하는 화면입니다.



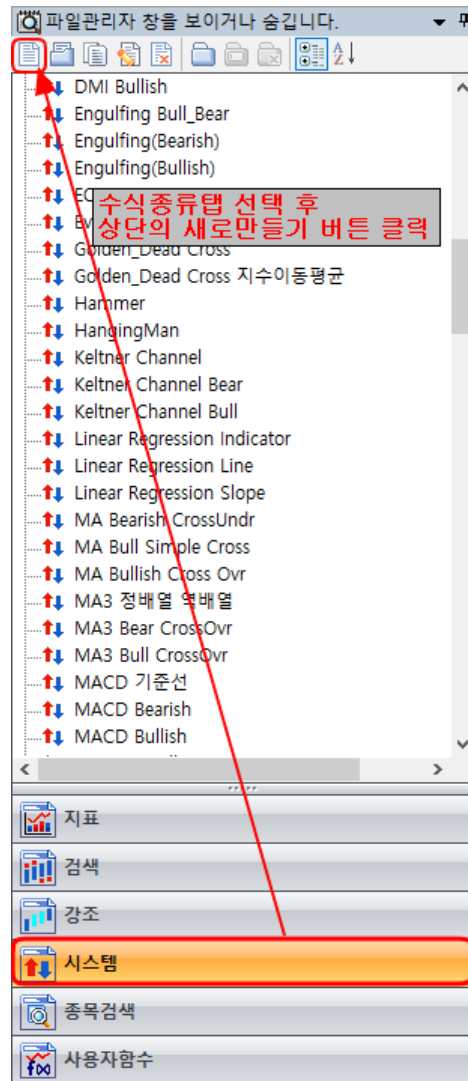
이 화면에서 작성할 수식의 목적에 맞는 수식종류를 선택하고 확인 버튼을 클릭합니다.
시스템식을 작성해야 하므로 시스템을 선택하고 확인 버튼을 클릭합니다.

나) 시스템 새로 작성



[시스템 새로 작성] 창이 나타나면 이름에 작성할 시스템의 이름을 입력하고 확인 버튼을 클릭합니다.
부가적으로 설명을 넣고 싶으면 설명란에 입력을 해도 됩니다.

파일관리자에서 시스템 탭을 선택 후에 [새로 작성] 버튼을 누르면 [새로 만들기] 화면을 건너뛰고 바로
시스템 새로 작성화면이 나타납니다.



다) 시스템 편집창

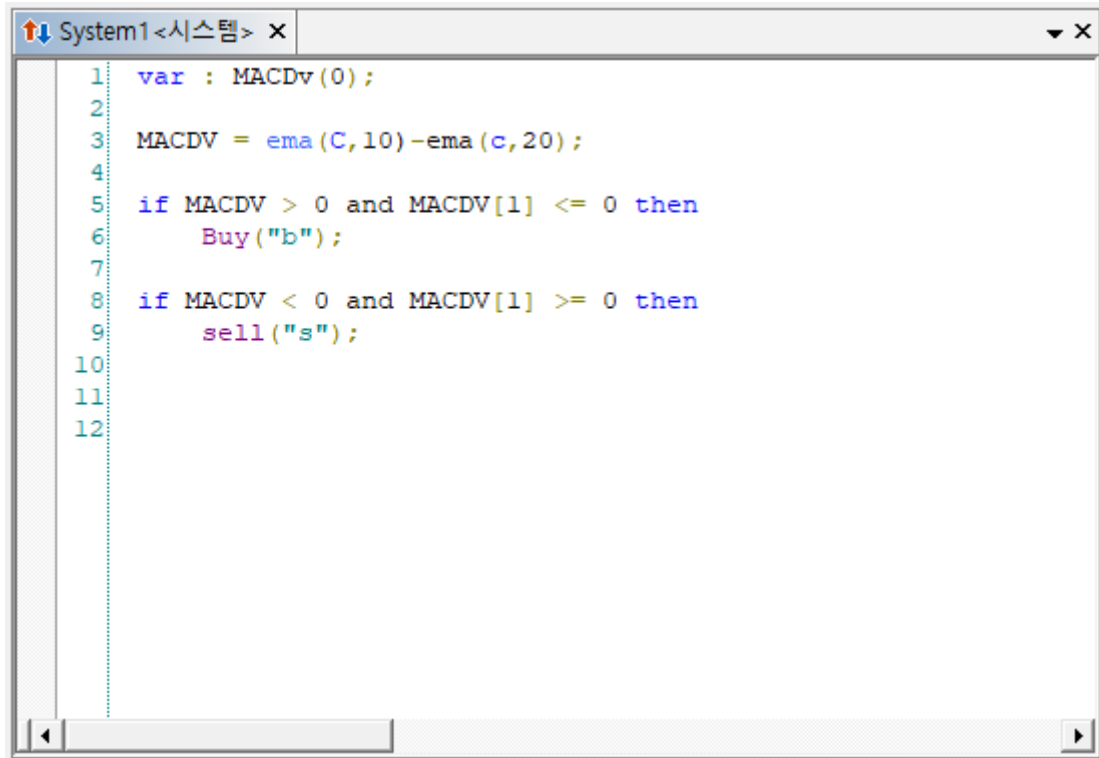


이름을 입력하고 확인버튼을 누르면 시스템 편집창이 생성됩니다.

라) 작성

시스템 편집창에서 구현하고자 하는 매매 아이디어를 문법에 맞게 작성합니다.

간단하게 MACD가 0값을 상향돌파하면 매수하고 MACD가 0값을 하향이탈하면 매도하는 내용으로 식을 작성하면 아래와 같습니다.



```
1  var : MACDV(0);  
2  
3  MACDV = ema(C,10)-ema(c,20);  
4  
5  if MACDV > 0 and MACDV[1] <= 0 then  
6      Buy("b");  
7  
8  if MACDV < 0 and MACDV[1] >= 0 then  
9      sell("s");  
10  
11  
12
```

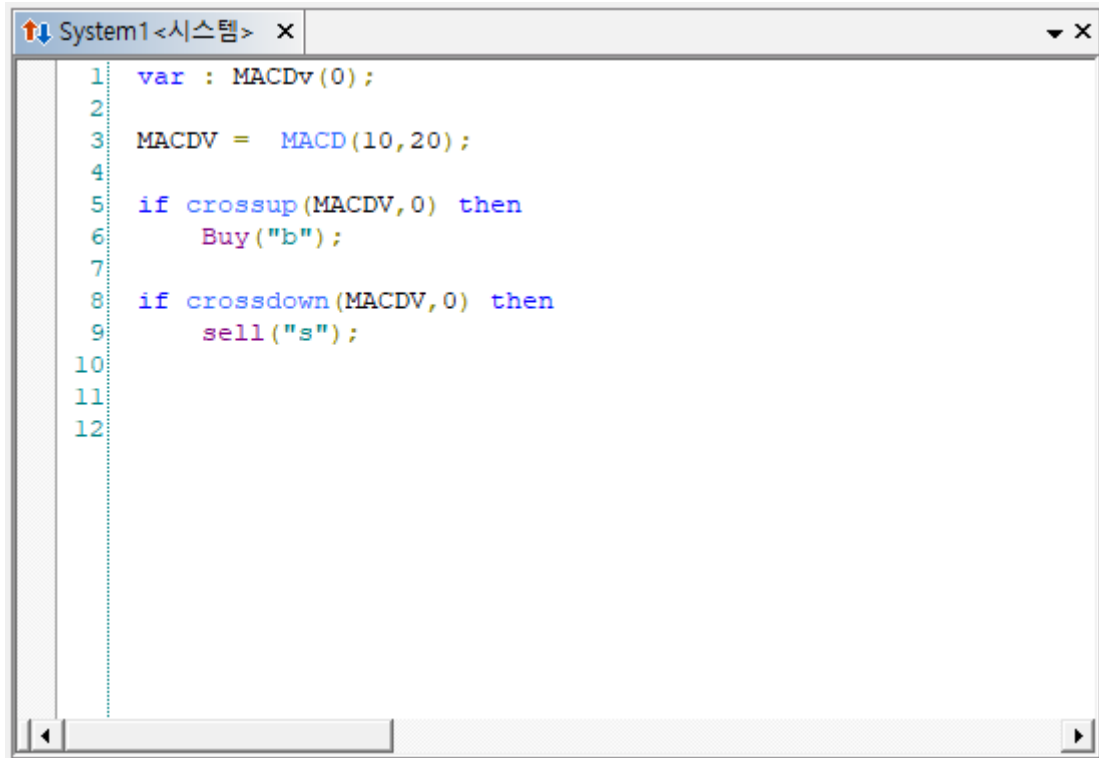
수식을 작성하는 것은 일종의 영어로 작문을 하는 것과 비슷합니다. 영작을 하기 위해서는 문법과 영단어를 알아야 하므로 예스랭귀지의 기본문법과 제공되는 함수등을 먼저 숙지하면 수식을 작성하는게 한결 쉬워 집니다.

위의 식은 MACD가 0선을 상향돌파하면 매수 하향이탈하면 매도하는 식입니다.

MACDV라는 단어를 만들어 단기지수이평에서 장기지수이평을 뺀 값을 저장해 놓고 if조건문에서 현재봉의 MACD상태와 전봉의 상태를 비교하여 상향돌파와 하향이탈이라는 조건문을 만든 식입니다.

랭귀지에 기본적으로 MACD 지표값을 구현하는 함수와 상향돌파와 하향이탈을 구현하는 함수가 있으므로 이런 함수에 대해 알고 있으면 아래와 같이 더 간단히 식을 작성할 수 있습니다.


MACD지표값은 MACD함수, 상향돌파는 Crossup, 하향이탈은 Crossdown 함수를 이용하여 수식을 작성하면 됩니다.



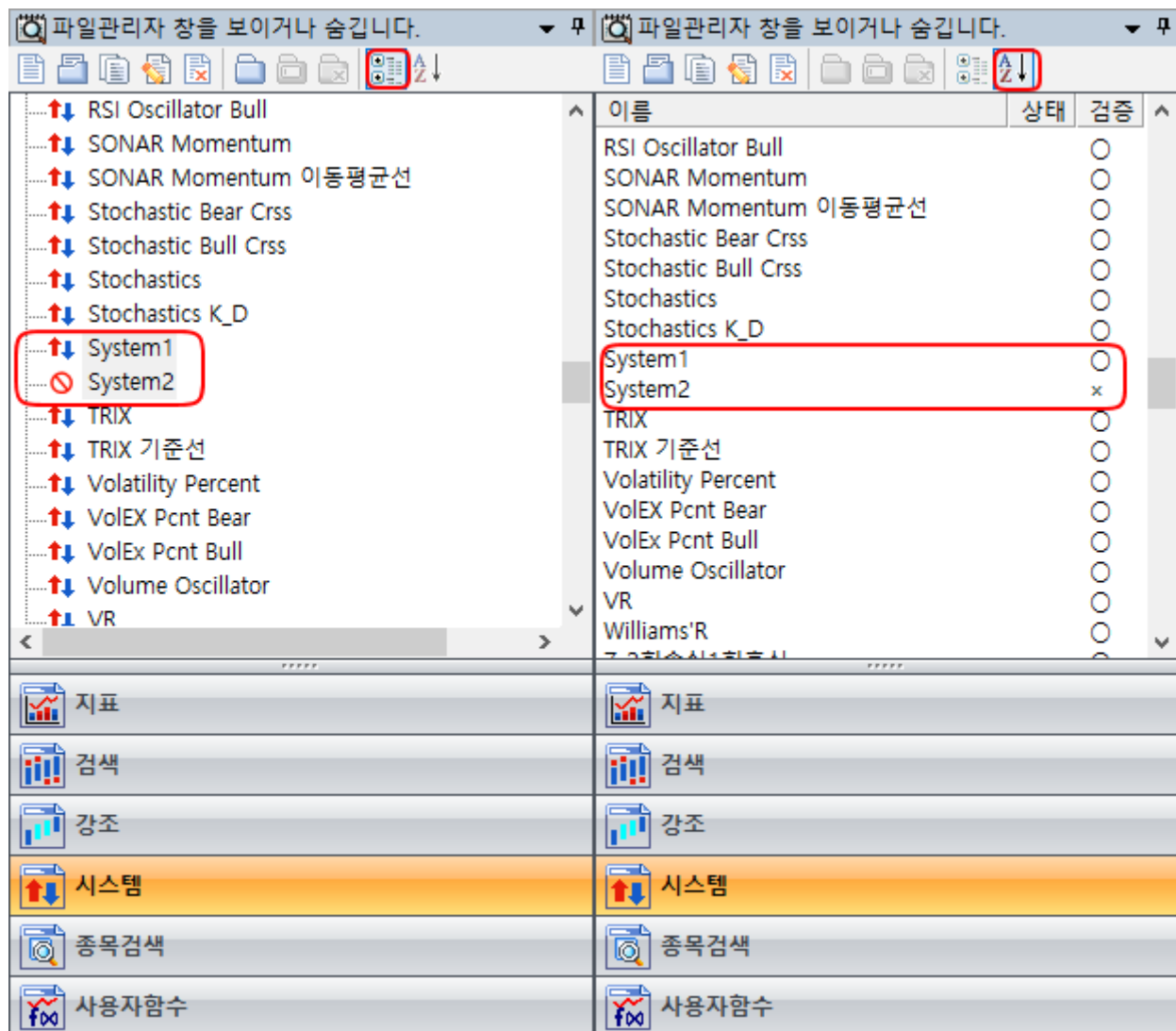
```
1  var : MACDV(0);  
2  
3  MACDV = MACD(10,20);  
4  
5  if crossup(MACDV,0) then  
6      Buy("b");  
7  
8  if crossdown(MACDV,0) then  
9      sell("s");  
10  
11  
12
```

마) 검증

수식은 작성을 모두 마쳤으면 검증 과정을 거쳐야 합니다. 이 검증은 기존의 검증이 완료된 수식이라고 해도 수정을 하였다면 수정과정에서 문법적으로 오류가 있을 수 있으므로 다시 검증을 해주어야 합니다.

수식 검증은 상단의 도구모음에서  를 누르거나 수식 위에서 마우스 오른쪽을 클릭하면 나오는 메뉴에서 상단의 검증을 클릭하거나 F4키를 누르면 됩니다.

검증은 작성된 수식이 문법적으로 문제가 있는지를 검사하는 과정이고 문법적으로 이상이 있다면 오류 창에 그 내용을 표시합니다. 만약 문제가 없다면 “검증이 완료되었습니다.”라는 메시지가 나타나며 해당 시스템식은 자동으로 저장됩니다.



검증을 통과한 시스템은 자동으로 저장되어 파일관리자 시스템 영역에 해당 이름의 수식이 추가가 됩니다. 또한 검증 항목에 O가 표시됩니다. 위에서 작성한 system1이 검증에 O 표시와 함께 시스템 영역에 추가된 것을 확인할 수 있습니다. 하지만 system2와 같이 검증을 완료하지 못하고 저장된 수식은 시스템 영역에 추가는 되지만 검증에 X로 표시되고 X로 표시된 수식은 차트에 적용할 수 없습니다.

검증완료여부는 파일관리자를 리스트 모드로 볼 때만 확인할 수 있습니다.

바) 수식 적용

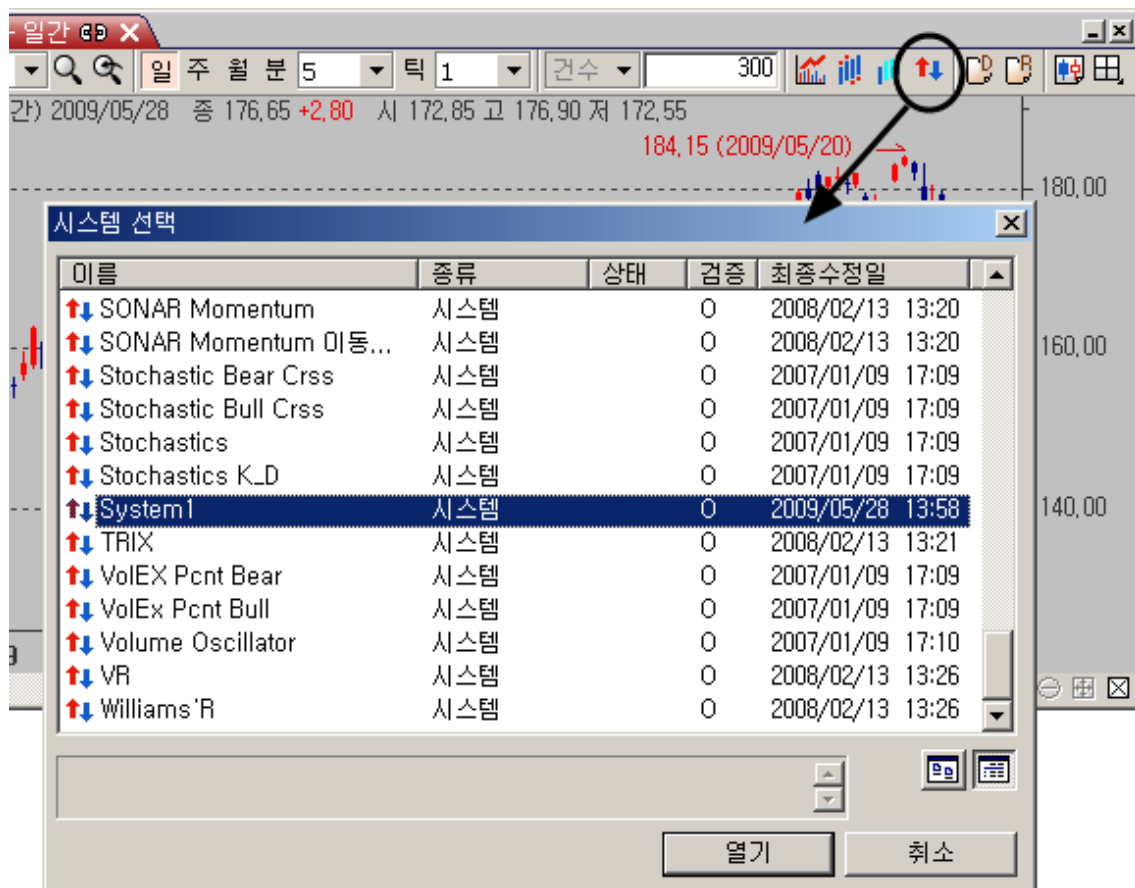
“검증이 완료되었습니다”라는 메시지가 나온다면 작성한 식은 차트에 적용될 수 있습니다.

프로그램에는 두 종류의 차트가 제공되고 있습니다. 하나는 전략실행차트이고 다른 하나는 시뮬레이션 차트입니다. 실시간 차트는 최대 10000개 까지만 과거 데이터를 불러와 작용할 수 있는 차트로 장중에 실시간으로 시세를 받아 실전 매매를 가능하게 하는 차트입니다. 시뮬레이션 차트는 전일 데이터까지만 조회가 가능한 차트로 5000개봉을 초과하는 데이터를 조회하여 수식을 적용할 수 있습니다. 프로그램에서는 KP200선물 분봉의 경우 2000년 7월 18일부터 조회가 가능한데 이 시뮬레이션 차트에서 기간을 설정하여

조회할 수 있습니다.

기본적으로 수식의 적용방법은 전략실행차트나 시뮬레이션이나 모두 동일합니다.

차트에서 시스템 선택창을 열고 목록에서 system1을 찾아 열기를 누르면 됩니다.



3. 예스랭귀지 문법

1) 기본문법

가) 수식은 위에서 아래로 읽어 내려 간다.



예스랭귀지는 작성된 수식을 첫번째 줄부터 마지막 줄로 읽어 들어가며 해석하게 됩니다. 일반적으로 글을 읽는 때 위에서 아래로 읽어 내려가며 줄거리를 파악해 나가는 것과 같이 사용자가 작성한 수식을 위에서부터 아래로 읽어 내려가면서 계산식이 있으면 계산을 하고 문장을 해석 하면서 작성자의 지시가 있으면 해당 지시를 실행하게 됩니다.

수식 작성시 이런 부분을 간과하고 작성하므로 해서 의도와는 다른 결과를 얻는 경우가 종종 발생하기도 합니다. 예를 들면 변수를 사용하여 제어문에 작성을 완료했는데, 제어문에서 사용된 변수에 값의 할당을 제어문 보다 뒤에 한 경우입니다.

아래의 예를 보면 그 차이를 확인 할 수 있습니다.



```

Example1 <시스템>

var1 = ma(C,5);
var2 = ma(C,20);

5 if crossup(var1,var2) Then
    buy("B");
  
```



```

Example2 <시스템>

if crossup(var1,var2) Then
    buy("b");

5 var1 = ma(C,5);
  var2 = ma(C,20);
  
```

Example1 수식은 먼저 변수에 5일 이동평균과 20일 이동평균을 계산하여 할당하고 제어문에서 이를 사용하여 식을 작성했으므로 정상적으로 골드크로스가 난 봉에 신호가 발생했음을 확인할 수 있습니다. 하지만 Example2 수식은 골드크로스가 발생한 음봉이 아니라 그 다음 봉에서 신호가 발생한 것을 볼 수 있습니다. 이것은 제어문에 사용된 변수가 그 제어문 보다 아래에서 값이 할당되기 때문에 제어문을 읽어 들어가는 시점에서는 변수가 아직 현재봉의 값으로 계산되지 않은 상태이기 때문에 전봉의 값으로 계산한 값을 사용하기 때문입니다.



“차 떠나고 손 흔들지 말자.”

이렇게 신호에 필요한 모든 내용이 수식 안에 있지만 배치의 문제로 엉뚱한 신호를 발생시킬 수 있으므로 항상 제어문에 사용할 변수에 값의 할당은 해당 제어문보다 위에서 값을 할당해서 사용하도록 해야 합니다.

나) 첫 봉부터 마지막 봉으로 실행

수식을 차트에 적용하게 되면 차트의 첫 봉부터 마지막 봉 쪽으로 순차적으로 움직이면서 각각의 봉에서 작성한 내용을 계산하여 실행하게 됩니다. 시간의 역순으로 어떤 행위를 할수 없습니다.

다) 대소문자를 구별을 하지 않는다.

예스랭귀지에서는 대소문자를 구별하지 않습니다. 아래와 같이 식을 작성한다고 해도 랭귀지에는 대소문자를 구분하여 값을 저장하지 않습니다.

```
Var : mav(0);

mav = ma(close,5);
MAU = ma(close,20);

If mav > MAU then
    Buy();
```

수식은 위에서 아래로 읽어 내려가므로 처음 mav에 5일 이동평균을 할당하고 바로 20일 이동평균으로 다시 할당하므로 20일 이동평균만 사용하는 결과가 됩니다. 이렇게 동일한 이름으로 대소문자로 구별하여 값을 할당하게 되면 가장 나중에 할당된 값만을 사용하게 되므로 주의해야 합니다.

변수를 만들어 사용할 때 대소문자 구분하지 않고 독해하기 편한 이름으로 만들어 사용하면 됩니다.

라) 하나의 단어 내에서 띄어쓰기나 줄바꿈은 허용되지 않는다.
단어와 단어 사이의 띄어쓰기나 줄바꿈은 제한 없이 가능합니다

```
var1 = ma      (Close,5);
var2 = ma
      (Close.20);
var3 =
      Ma(c,60);
```

하지만 하나의 단어 내에서 띄어쓰기와 줄바꿈을 하게 되면 하나의 단어가 분리되면서 각각 독립된 단어로 인식함으로 해서 에러가 발생하게 됩니다.

```
var1 = ma(Cl    ose,5);
var2 = ma(Cl
      ose,20);
```

Close라는 단어는 종가를 나타내는 데이터 예약어입니다. 이 단어 내에서 띄어쓰기나 줄바꿈하게 되면 랭귀지에서 Cl과 ose로 분리하여 인식하게 되므로 단어와 단어 사이만 띄어쓰기나 줄바꿈을 해야 합니다.

마) 문장의 마침은 세미콜론(;)을 찍는다.

하나의 문장의 작성이 모두 완료되었다면 마지막에 세미콜론(;)을 찍어 마침을 표시해 주어야 합니다. 다만 if 문과 같은 제어문의 경우에는 문장의 마침을 찍지 않습니다.

2) 문장 구조

문장은 크게 선언부, 할당부, 제어부, 실행부 4가지로 나뉜다.

선언부	수식에서 사용될 변수를 선언하는 문장
할당부	변수에 계산값이나 논리판단의 결과값 혹은 문자열을 할당하는 문장
제어부	사용자가 지정한 조건의 만족여부를 판단하는 문장
실행부	제어부가 True일 때, 취해야 할 행동을 지칭하는 문장

아무리 복잡하고 긴 내용의 수식이라고 해도 분류해 보면 위와 같은 4개의 문장들로 구성이 되어 있습니다. 소설과 같은 글이 일반적으로 기승전결의 구조를 가지는 것과 같이 수식도 같은 구조를 가진다고 생각하면 됩니다.

서론(序論) → 설명(說明) → 증명(證明) → 결론(結論)
어떤 것이 있는데 → 그것은 무엇이고 → 그것이 어떻게 되면 → 무엇을 해라.
변수를 선언 → 변수에 값을 할당 → 수행된 조건을 판단 → 참이면 실행

```

input : Short(5),Long(20);    # 선언부(외부변수)
var : Sma(0),Lma(0);        # 선언부(내부변수)

Sma = ma(C,Short);           # 할당부
Lma = ma(C,Long);            # 할당부

if crossup(Sma,Lma) Then     # 제어부
    Buy();                    # 실행부

if crossup(Sma,Lma) Then     # 제어부
    Sell();                    # 실행부

```

위의 문장을 해석하면 아래와 같습니다.

“SMA와 LMA라는 변수가 있는데 각각 Short과 Long 기간으로 계산된 이동평균 값이고 SMA가 LMA를 상향돌파하면 매수하고 하향이탈하면 매도한다.”

3) 각종 연산자

```

input : Period(20),DU(2);
var : EntryHL(0),Hval(0);
Array : BollUP[1](0);

BollUP[0] = BollBandUp(Period,Du);

if (C > 0 and (H==C or L == 0)) and C > BollUP[0] Then
    buy("B", AtStop,C+(H+C)/2,10);

if MarketPosition == 1 Then
{
    Hval = Highest(H,BarsSinceEntry);
    EntryHL = H[BarsSinceEntry]-L[BarsSinceEntry];

    if Hval >= EntryPrice*1.05 Then
        ExitLong("X",AtStop,Hval-EntryHL);
}

```

가) 산술 연산자

기 호	이 름	예문
+	더하기	Entryprice+0.5 (진입가 더하기 0.5)
-	빼기	Entryprice-0.5 (진입가 빼기 0.5)
*	곱하기	Entryprice+PriceScale*2

/	나누기	Entryprice+(H+L)/2
%	나머지	10/3 → 1
^	거듭제곱	10^2 → 100

산술연산자는 흔히 말하는 사칙연산을 할 수 있는 연산자입니다. 예를 들면 봉의 중간 값을 구한다고 할 때 (H+L)/2와 같이 작성할 수 있습니다.

산술연산자의 우선순위는 곱하기,나누기 > 더하기 > 빼기 순입니다.

나) 관계연산자

기 호	이 름	예문
==	같다	Close == Open (종가와 시가가 같다)
!=	같지 않다	Close != Open (종가와 시가가 같지 않다)
!	아니다	!(C > O) (양봉이 아니다)
<>	같지 않다	Close <> Open (종가와 시가가 같지 않다)
<	작다	V < V[1] (거래량이 전봉보다 작다)
>	크다	Oi > Oi[1] (미결제약정이 전봉보다 크다)
<=	작거나 같다	bids <= Asks (매수잔량이 매도잔량 이하이다)
>=	크거나 같다	bids >= Asks (매수잔량이 매도잔량 이상이다)

관계연산자는 두 개의 값을 비교할 때 사용하는 것이며 비교의 결과값은 True(참) 혹은 False(거짓) 입니다. 같다를 표현할 때는 ==(등호 두개)을 사용해야 하며 작거나 같다와 크거나 같다를 표현할 때 항상 부등호 다음에 등호가 오는 것에 유의하며 식을 작성하여야 합니다.

다) 논리연산자

기 호	이 름	예문
And	둘 다 참이면 true	Condition1 and Condition2
Or	둘 중 하나가 참이면 True	Condition1 or Condition2

논리값이란 주어진 명제에서 참과 거짓을 판단한 결과값입니다. 논리연산자 and는 앞과 뒤의 논리값이 모두 참이어야 True가 되고 or 는 앞의 논리값과 뒤의 논리값 중에 하나이상 참이면 True가 됩니다.

예를 들면 C > O and H == C 와 같이 작성하게 되면 종가가 시가보다 커야 하고 고가가 종가와 같아야 한다는 2개의 조건이 모두 만족해야 True가 됩니다. 하지만 C > O Or H == C 와 같이 작성하면 종가가 시가보다 크거나 고가와 종가가 같으면 True가 되게 됩니다.

※ and와 or의 중복

if~ then문 내에서 And와 Or가 동시에 사용될 때 작성자가 의도하는 대로 적절한 순서로

조건을 참과 거짓을 판단하게 하기위해서는 소괄호()를 이용하여 각 조건사이의 관계를 정리해 주어야 할 필요가 있습니다.

예를 들어 아래와 같은 수식을 보자

```
if C > 0 and H == C or L == 0 Then
    buy("B",OnClose,def,1);
```

매수를 진입하기 위한 조건식이다. 3가지의 조건이 And 와 Or를 결합되어 있습니다.

양봉인데 위 꼬리가 없거나 아래 꼬리가 없는 봉에 진입을 하고자 작성했지만 양봉도 아닌 봉에서 신호가 발생하는 것을 확인할 수 있습니다.



예스랭귀지에서는 If ~ then문에 And와 Or가 사용되고 작성자가 특별히 관계를 지정해 주지 않는다면 좌우에서 우로 방향으로 연산하게 됩니다. 그러므로 위 식은 위 꼬리가 없는 양봉이거나 저가가 시가와 같은 봉이면 진입이라고 작성한 것과 같습니다.

그러므로 소괄호를 이용하여 관계를 명확히 구분해 주어야 의도하는 결과를 얻을 수 있습니다. 논리연산자 사이에 괄호가 없는 경우 우선순위는 and > Or 순으로 연산합니다.

```
if C > 0 and (H == C or L == 0) Then
    buy("B",OnClose,def,1);
```

라) 할당연산자

기 호	이 름	예문
=	할당.	Value1 = H-L;

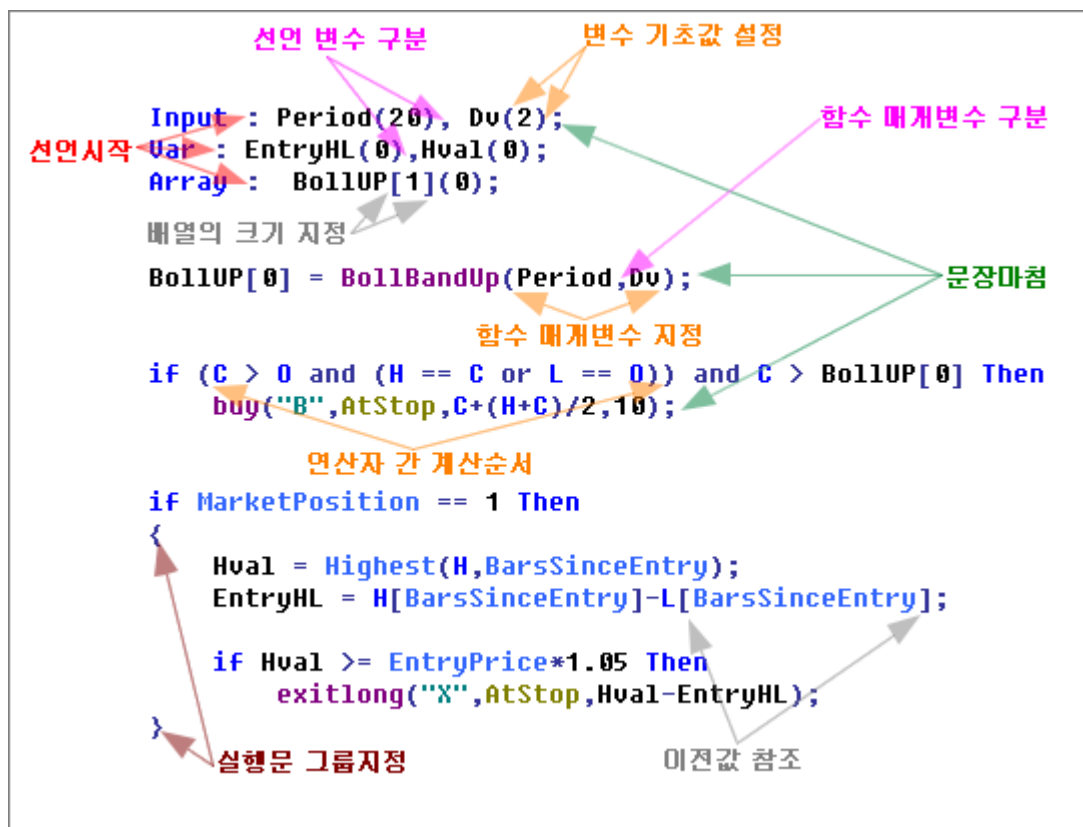
예스랭귀지에서 등호(=)의 의미는 일반적인 수학의 개념과는 다르게 같다라는 의미가 아니라 할당(저장)한다는 의미입니다. 즉 등호 좌우의 비교하는 관계연산자가 아니라 등호 우측의 값을 좌측의 값으로 할당하는 것으로 어떤 계산작업을 통해서 얻은 결과값을 연산자 좌측으로 이동시켜 저장하는 것입니다.

이 부분은 의외로 수식을 작성할 때 틀리기 쉬운 부분입니다. 같다를 표현할 때와 할당을 표현할 때를 잘 구분하여 수식을 작성해야 합니다.

일반적으로 if ~ then 안에서는 조건을 비교하여 만족유무를 판단하므로 할당연산자(=)는 거의 사용되지 않습니다.

마) 구두점 및 괄호

기 호	이 름	사 용
:	콜론	선언의 시작
;	세미콜론	문장의 마침표
,	кома(쉼표)	함수 매개변수 구분 선언 변수 구분
()	소괄호	함수에서 매개변수 지정 연산자 간의 우선순위 지정 변수 선언 시 기초값 지정
{ }	중괄호	실행문 그룹지정
[]	대괄호	이전값 참조 배열의 크기지정



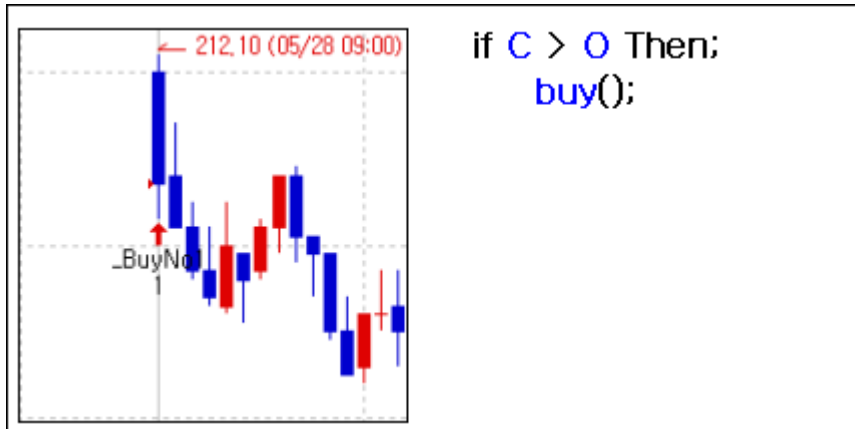
콜론은 변수의 선언을 시작을 알릴 때만 사용합니다.

Input : 외부변수 선언 시작, var : 내부변수 선언시작 array : 배열변수 선언시작을 의미합니다.

해당 부분 외에는 사용되는 내용이 없습니다.

세미콜론은 문장의 마침을 나타내며 전체적으로 보면 if~ then문의 then뒤 즉 제어문 뒤만 빼고 모두 찍는

다고 보면 됩니다. 세미콜론을 잘못 표기하면 문장의 단절이 일어나 작성된 수식이 의도와는 다른 결과를 반환 할 수 있습니다.



가령 위와 같이 then 뒤에 세미콜론을 찍게 되면 if~then문과 buy();과 단절이 되므로 buy()는 if문과 별개로 동작하여 원래 의도한 양봉에서 매수한다라는 내용과는 다르게 음봉에서도 신호가 발생하게 되므로 유의해야 합니다.

콤마는 값의 구분이 필요할 때 사용합니다. 선언해야 할 변수가 복수일 때 각 변수의 선언사이에 콤마를 찍어 구분을 해주고 함수에서 매개변수를 지정할 때 각 매개변수를 구분할 때 사용합니다.

바) 연산자 우선 순위

여러 개의 연산을 수행할 경우 연산자 우선 순위라고 하는 미리 정해진 순서에 따라 식의 각 부분을 해석하여 분해합니다. 소괄호를 사용하면 연산에 대한 우선 순위를 바꾸거나 다른 부분을 계산하기 전에 식의 한 부분을 먼저 계산할 수 있습니다.

괄호 안의 연산은 언제나 괄호 밖의 연산보다 먼저 수행되지만, 괄호 안에서는 일반적인 연산자 우선 순위를 따릅니다.

괄호 > 산술 연산자 > 비교연산자,관계 연산자 > 논리 연산자 > 할당 연산자

4) 데이터 및 데이터 정보

실시간으로 거래가 발생하게 되면 거래소로부터 체결가격(Tick)이 수신됩니다. 이와 같이 실시간 거래가 발생할 때마다 전송 받은 데이터를 이용해서 봉을 그리게 되고 그려진 봉 하나하나에는 일정한 시간간격 동안의 가격의 움직임인 시가, 고가, 저가, 종가, 거래량 등의 시세정보가 나타나게 됩니다.

데이터	약어	설명
Open	O	시가

Close	C	종가
High	H	고가
Low	L	저가
Money	M	거래대금
Volume	V	거래량
UpVol		상승형 거래량
DownVol		하락형 거래량
Upticks		상승형 체결건수
DownTicks		하락형 체결건수
Ticks		체결건수
OpenInterest	OI	미결제약정
Bids		매수잔량
Asks		매도잔량
bDate		영업일(Business date)
sDate		봉의 시작날짜
Date		봉의 끝날짜
sTime		봉의 시작시간
Time		봉의 끝시간
BasePrice	BP	기준가
Index, Barindex, Globalbarindex		봉 번호

※ EL모드에서만 사용되는 데이터 예약어

데이터	약어	설명
D		끝날짜
OpenInt, I		미결제약정

데이터 정보	설명				
BarInterval	적용된 차트의 시간간격을 리턴 (5분봉이면 → 5)				
BigPointValue	1계약의 1포인트를 금액으로 환산. (옵션 100,000 선물 500,000 현물은 1)				
CodeCategory	종목 구분값 <table> <tr> <td>1</td><td>KOSPI</td></tr> <tr> <td>2</td><td>KOSDAQ</td></tr> </table>	1	KOSPI	2	KOSDAQ
1	KOSPI				
2	KOSDAQ				

	3	제3시장
	4	선물
	5	지수
	6	옵션
	7	종목옵션
	8	해외종목
	9	참조데이터
CodeCategoryEx	상세 종목 구분값	
	11	KOSPI 종목
	12	KOSDAQ 종목
	13	ETF_KOSPI
	14	ETF_KOSDAQ
	15	ELW
	21	KP200 선물
	22	KOSDAQ 선물
	23	주식 선물
	24	미니 KP200 선물
	25	KP200 선물 스프레드
	26	주식선물 스프레드
	27	미니 KP200 선물 스프레드
	31	KP200 콜옵션
	32	KP200 풋옵션
	33	KOSDAQ 콜옵션
	34	KOSDAQ 풋옵션
	35	주식옵션
	36	미니 KP200 콜옵션
	37	미니 KP200 풋옵션
	38	WEEKLY 콜옵션
	39	WEEKLY 풋옵션
	41	변동성지수 선물
	42	섹터지수 선물
	43	변동성지수 선물 스프레드
	44	섹터지수 선물 스프레드

	45	유로스톡스50지수 선물
	46	유로스톡스50지수 선물 스프레드
	47	KQ150 선물
	48	KQ150 선물 스프레드
	49	KRX300지수 선물
	50	KRX300지수 선물 스프레드
	51	코스피지수
	52	코스닥지수
	53	KQ150 콜옵션
	54	KQ150 풋옵션
	61	해외 데이터
	71	거래 가능 참조데이터
	72	거래 불가 참조데이터
	73	사용자가 추가한 데이터
CurrentBar	수식계산에 필요한 필수 봉갯수 이후의 봉수를 반환	
DataCompress	차트에 적용된 데이터의 주기를 반환 (틱봉 :0, 초봉:1, 분봉:2, 일봉:3, 주봉:4, 월봉:5, 삼선전환도:6, P&F:7)	
DeliveryMonth	만기월(추후 제공을 위한 예약어)	
DeliveryYear	만기년(추후 제공을 위한 예약어)	
ExpirationDate	만료일(추후 제공을 위한 예약어)	
Interval	적용된 차트의 시간간격을 리턴, Barinterval과 같음 (5분봉이면 → 5)	
PointValue	PriceScale을 금액으로 환산 ex) 선물 : 25,000, 옵션 : 3미만 1,000, 3이상 5,000	
PriceScale	해당 종목의 호가 단위, 틱 ex) 선물 : 0.05, 옵션 : 3미만 0.01, 3이상 0.05	
SymbolCode	종목코드를 리턴, 반환값 문자열 var : Code(""); Code = SymbolCode; MessageLog("종목코드 %s 증가 %.2f ",Code,Close);	
SymbolName	종목명 리턴 var : Name(""); Name = SymbolName;	

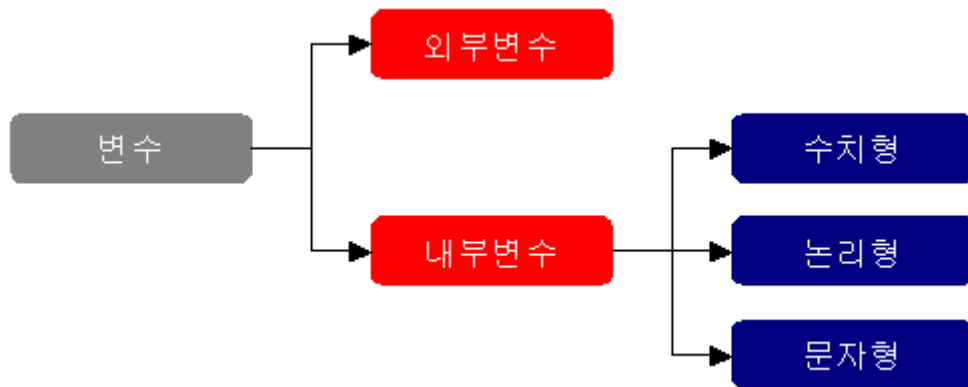
	MessageLog("종목명 %s 증가 %.2f ",Name,Close);
UserID	<p>프로그램 접속 아이디를 리턴합니다.</p> <p>수식에서 UserID로 조건을 주면 해당 아이디로 접속했을때만수식이 동작하게 제어할 수 있습니다.</p> <p>Input : Period(14), LPercent(30), SPercent(70);</p> <p>Var : value(0);</p> <p>value = RSI(Period);</p> <p>if UserID == "ABCDE" Then{</p> <p style="padding-left: 40px;">If CrossUP(value, LPercent) Then</p> <p style="padding-left: 80px;">{</p> <p style="padding-left: 120px;">Buy("매수");</p> <p style="padding-left: 80px;">}</p> <p style="padding-left: 40px;">If CrossDown(value, SPercent) Then</p> <p style="padding-left: 80px;">{</p> <p style="padding-left: 120px;">Sell("매도");</p> <p style="padding-left: 80px;">}</p> <p>}</p>

5) 변수

가) 변수란?

변수는 계산식이나 논리식의 결과값을 할당 받아 저장할 수 있는 공간으로 이전 값을 최근 값으로 교체하면서 하나의 데이터를 저장하는 공간입니다.

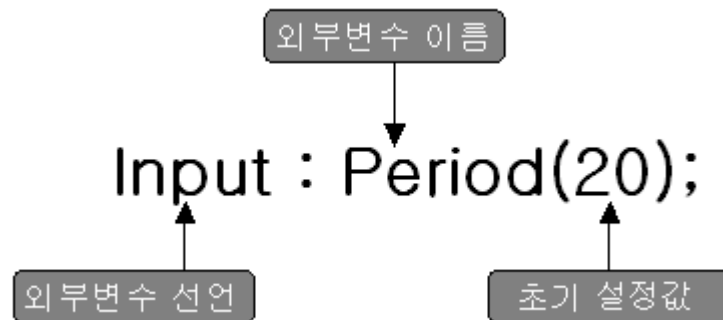
어떤 단어를 변수로 사용하기 위해서는 해당 단어가 변수로 사용될 것임을 예스랭귀지에 알려 주는 작업이 필요하고 이런 작업을 변수 선언이라고 합니다. 모든 변수는 선언 후에 값을 할당 받아 사용될 수 있습니다.



변수는 **외부변수**와 **내부변수**로 나뉘고 내부변수는 다시 수치형,논리형 그리고 문자형으로 나뉘지게 됩니다.

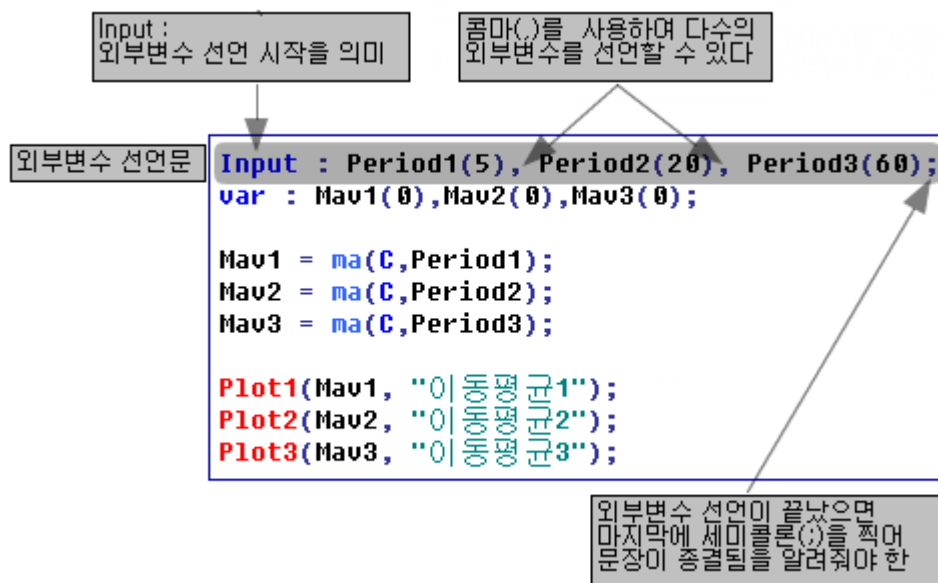
나) 외부변수

외부변수는 수식에서 사용되는 특정 값을 사용자 임의로 변경이 용이하게 하기 위한 변수 선언입니다.



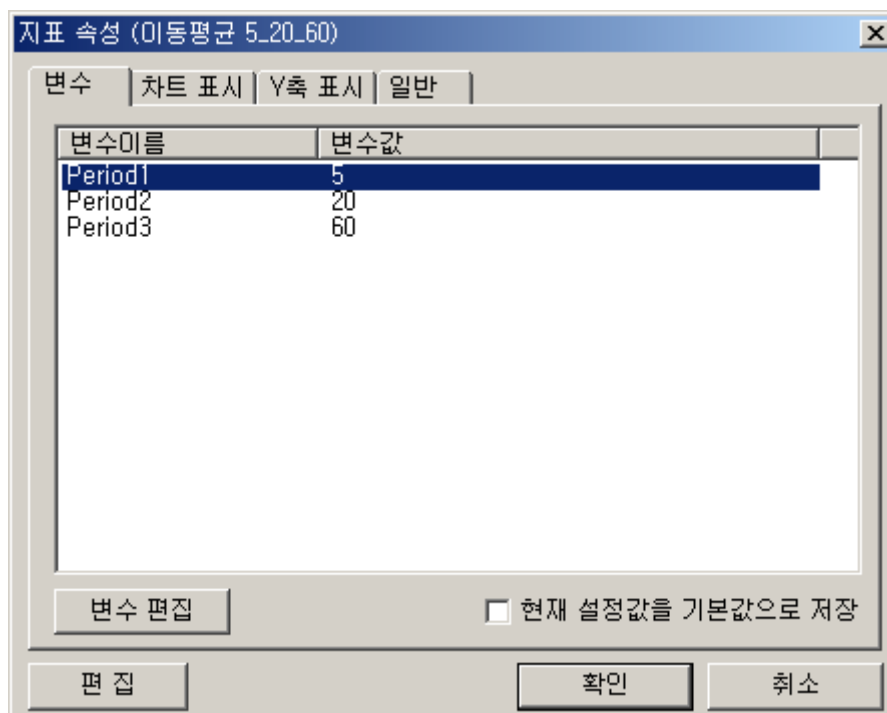
모든 HTS의 차트에는 이동평균선을 볼 수 있고 이동평균선의 기간은 사용자가 간단히 설정화면에서 임의로 변경이 가능합니다. 이것은 내부 로직에 기본으로 설정된 기간은 있지만 사용자가 임의로 기간을 변경하면 그 입력된 값을 받아서 재계산하여 보여주기 때문입니다. 만약 설정화면에서 기간을 변경하지 못하게 되어 있다면 기본으로 설정된 기간으로만 이동평균선을 사용하거나 해당 프로그램 언어를 분석하여 수정해야 하는 불편함이 있습니다.

이렇게 수식에서 사용되는 특정 값들 중에는 임의로 변경하며 사용하고자 하는 값이 있는데 이런 값들을 매번 엑셀러 Gui 편집기를 실행하여 해당 값을 변경하는 것은 상당히 번거로운 작업이므로 외부변수로 선언하여 설정화면(해당 수식의 속성창의 변수 TAB)에서 값의 수정이 가능 하게 만들어 주면 간단히 클릭 몇 번으로 변경된 값으로 재계산한 결과를 확인 할 수 있게 해줍니다. 또한 시스템 수식에서 특정 값을 외부변수로 선언하면 시뮬레이션 차트에서 변수 최적화(Optimization)가 가능하게 됩니다.



외부변수는 input이나 inputs로 선언을 하고 모두 마쳤으면 마지막에 세미콜론(;)을 찍어줍니다.

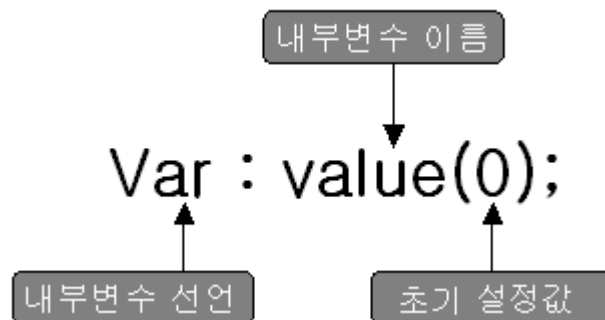
Input : 과 세미콜론(;) 사이에는 다수의 외부변수를 선언할 수 있고 선언과 선언 사이에 콤마(,)를 찍어주면 됩니다. 외부변수로 선언한 이름은 속성창의 변수 Tab에 그 이름과 기본값이 표시되고 변수편집이 가능하게 됩니다..



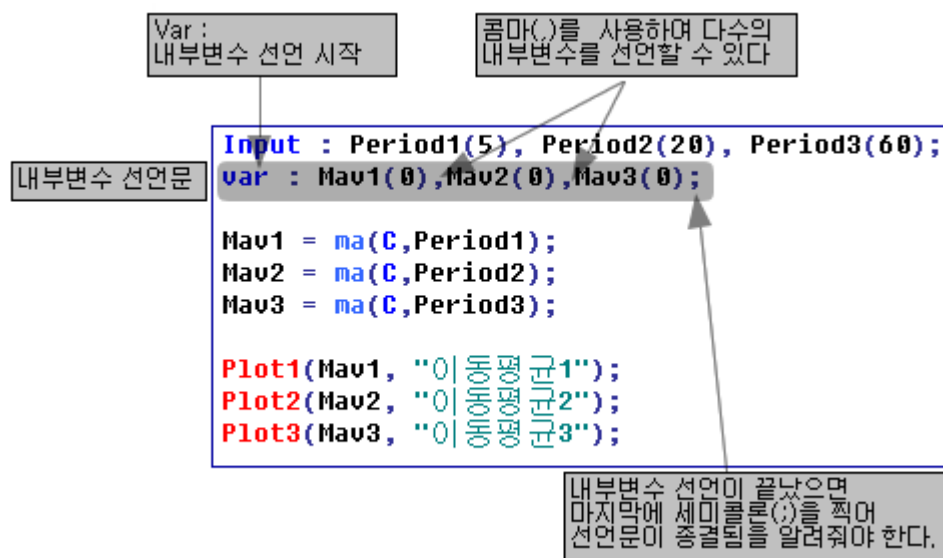
다) 내부변수

내부변수는 수식내에서 값을 할당받아 사용하는 변수입니다. 수식에서 수치를 얻기 위한 계산식이나 논리

판단의 결과값 혹은 텍스트를 저장 받는 변수입니다.



내부변수는 Var, Vars, Variable, Variables로 선언을 하며 콤마(,)를 이용해서 여러 개를 선언할 수 있으며 선언이 모두 마쳤으면 세미콜론(;)을 찍어줍니다.



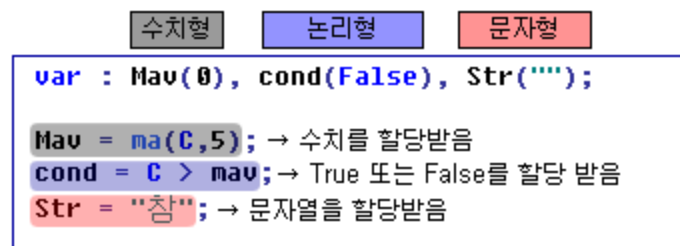
수식을 작성할 때 특정한 값을 얻기 위한 계산식이나 논리식 등을 반복적으로 사용하는 경우가 빈번하게 발생합니다. 이때 매번 이 계산식이나 논리식 등을 반복적으로 입력해 주는 것은 상당히 수고스러운 일이므로 특정한 공간(변수)에 결과값을 할당해 놓으면 이후에 이 결과값을 사용하고자 할 때는 동일한 계산식이나 논리식 등을 반복하여 작성할 필요 없이 이미 계산하여 값을 저장한 변수가 있으니 해당 변수를 가져와 사용하기만 하면 됩니다.

내부변수는 선언 유형에 따라 3가지로 나뉘게 된다.

변수구분	할당되는 결과값	초기 설정값
------	----------	--------

수치형 내부변수	수치(numeric)	(0)
논리형 내부변수	True(참) 또는 false(거짓)	(false)
문자형 내부변수	문자열	(" ")

수치 결과값 만을 받아들이는 수치형 내부변수와 논리판단의 결과 값만을 받아들이는 논리형 내부변수, 그리고 문자열만을 받아들이는 문자열 내부변수가 있습니다. 그러므로 어떤 값을 할당할 것이냐에 따라 해당 내부변수의 선언을 달리해야 합니다.



① 수치형 내부변수

수치형 내부변수는 수치를 할당 받는 변수입니다. 선언 시에 (0)으로 설정하면 해당변수는 수치형 변수이고 초기 설정값은 0이라는 의미이므로 논리식의 결과값이나 문자열은 할당이 불가능합니다. 반드시 0으로 선언할 필요는 없습니다. 기초값을 숫자로만 지정하면 됩니다.

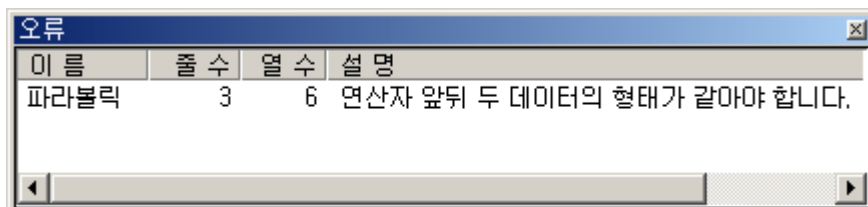
② 논리형 내부변수

논리형 내부변수는 논리판단의 결과값인 True 또는 False를 할당받는 변수입니다. 선언 시에 (False)로 선언하면 해당 변수는 논리형 변수이고 초기 설정값은 False 이고 수치나 문자열을 할당이 불가능합니다.

③ 문자형 내부변수

문자형 내부변수는 문자열을 할당 받는 변수입니다. 문자형 변수는 선언 시에 큰따옴표(Double Quotation Mark)를 기초 설정값으로 사용합니다. 할당할 때도 큰따옴표 안에 할당을 문자를 넣어주면 됩니다.

④ 연산자 앞뒤 두 데이터의 형태가 같아야 합니다.



수식을 작성 후에 검증을 하면 위와 같은 오류 메시지를 간혹 볼 수 있습니다.. 해당 메시지는 할당 연산자인 등호(=) 앞의 변수 선언유형과 뒤의 할당된 결과값 형태가 다르다는 의미로 해당 메시지가

나오면 변수의 선언유형을 확인하여 맞는 유형으로 변경해 주면 됩니다.

라) 내장변수

변수명	유형	초기 설정값
var1~var99	수치형	0
value1~value99	수치형	0
condition1~condition99	논리형	False

사용자의 편의를 위해 예스랭귀지에는 미리 선언되어 사용할 수 있는 내부변수가 있습니다.

수치형 내부변수 : var1~var99, value1~value99

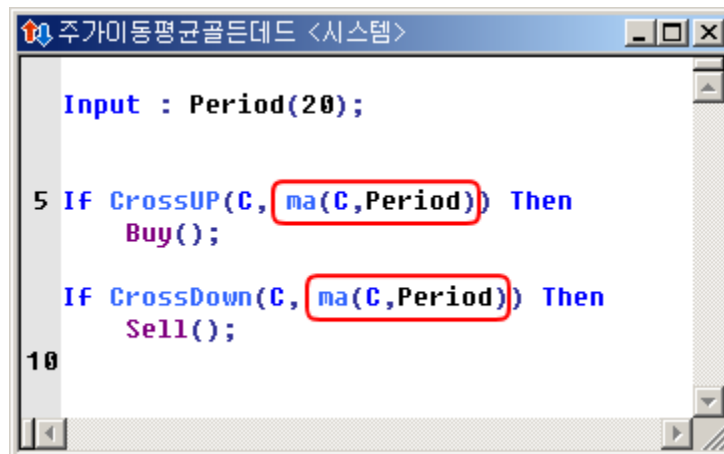
논리형 내부변수 : conditiond1~conditiond99

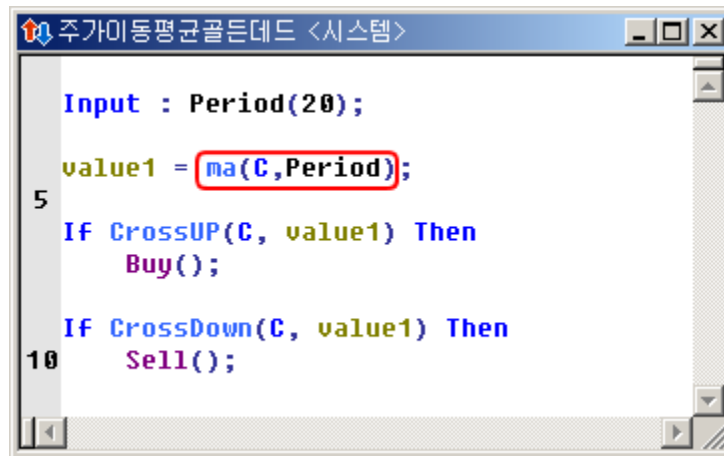
마) 변수 이용의 장점

변수를 사용하면 복잡한 내용을 반복적으로 작성함으로 인한 오류를 줄일 수 있고 식 내용이 시각적으로 더 간략해 지므로 수식을 독해할 때 더욱 수월하게 됩니다.

변수를 사용함으로써 얻을 수 있는 가장 중요한 장점은 계산 속도의 향상입니다.

동일한 계산의 반복 횟수를 줄여 주므로 계산 작업에 대한 부하가 감소함으로 해서 속도가 증가되고 더욱 효율적으로 프로그램이 운영되게 됩니다.





위 2개의 간단한 수식을 보면 2개의 식은 주가가 20 이동평균을 상향돌파하면 매수하고 하향 이탈하면 매도하는 수식으로 동일한 결과를 가져오는 시스템식입니다.

위에서부터 아래로 읽어 내려가면서 20이동평균의 계산을 몇 번을 하는지 각각 세어보면 위의 식은 총 2번의 계산을 하게 되고 아래 식은 1번 한다는 것을 알 수 있습니다. 물론 위와 같이 간단한 식에서는 1번 계산과 2번 계산은 차이는 크게 없다고 볼 수 있습니다. 하지만 이런 내용이 하나의 수식에서 여러 케이스가 있다면 절대 무시할 수 없는 속도상 차이가 발생할 수 있습니다. 그러므로 변수를 이용하여 보다 작은 메모리나 연산시간으로 해답을 얻을 수 있게 수식을 작성해 주어야 합니다.

6) 제어문

제어문이란 수식 작성을 할 때 특정 조건에 대해 만족했는지 여부를 확인하거나 특정 계산을 반복적으로 해야 할 경우에 사용되는 구문들을 말합니다. 예스랭귀지에서는 if문, for문, while문이 제공됩니다.

가) if 문

If문은 수식 작성 중에 가장 빈번하게 사용되고 기초가 되는 제어문으로 조건이 만족하면 지정된 실행문을 수행하고 만족하지 않으면 수행하지 않거나 다른 실행문을 수행합니다.

① if 조건 then 문

```
if crossup(C,ma(C,5)) then
    Hvalue = H;
# -> 종가가 5일 이동평균을 상향돌파하면 Hvalue에 고가 할당.

If High > Hvalue and Low > Low[1] then
    Buy();
# -> 고가는 Hvalue보다 크고 저가는 전봉 저가보다 크면 매수
```

if 조건 then 문은 if와 then 사이의 조건을 분석하여 해당 조건이 만족하느냐 안하느냐에 따라 실행문을

수행하는 제어문입니다.

If문은 반드시 then과 함께 사용 되어 하며 If 조건 then 문은 해석상 “만약 어떤 조건이 만족한다면”이라는 내용이므로 아직 문장이 종결되지 않았으므로 then뒤에는 문장의 마침을 뜻하는 세미콜론(;)을 찍지 않고 뒤에 오게 될 실행문에 세미콜론을 찍으면 됩니다.

② If 조건 then 그룹 문

기본적으로 if~ then문은 조건이 만족하면 하나의 실행문을 수행하게 됩니다. 그러나 수식을 작성하다 보면 if문이 만족하면 다수의 실행문을 수행해야 할 경우가 많이 발생합니다. 이 때는 if ~then 그룹 문을 이용해서 실행문을 하나의 그룹으로 묶고 그룹은 중괄호“{~}” 나 begin~end;라는 단어를 이용하여 실행문을 둘러싸서 하나의 그룹임을 나타내 주면 됩니다.

즉 if ~ then이 True이면 Begin부터 시작하여 end; 까지 그룹지어진 실행문이 모두 수행하게 됩니다.

종가가 5일 이동평균을 상향돌파하면
Hvalue에 고가 할당하고
매수

```
if crossup(C,ma(C,5)) then
{
    Hvalue = H;
    Buy()
}
```

고가는 전봉 고가보다 작고 저가는 전봉 저가보다 크면
매수하고
var1에 저가할당하고
cond에 True 할당

```
If High < High[1] and Low > Low[1] then
Begin
    Buy();
    Var1 = L;
    Cond = True;
End;
```

③ if 조건 then 중첩문

if 조건 then 중첩문은 if 조건 then문을 여러 번 겹쳐 사용한 경우로 실행문이 수행되기 위해서는 두개 이상의 if 조건 then문이 만족해야만 수행되는 구조입니다.

예를 들면, 다음과 같습니다.

시가가 전일증가보다 1포인트 이상 크고 양봉이면 매수를 한다. 시가가 전일증가보다 1포인트 이상 크고 음봉이면 매도를 한다.	
<pre> # 수식1 if 0 > C[1]+1 and C > 0 then buy(); if 0 > C[1]+1 and C < 0 then sell(); </pre>	<pre> #수식2 if 0 > C[1]+1 then { if C > 0 then buy(); if C < 0 then sell(); } </pre>

예시와 같이 매수식과 매도식은 수식1과 같이 각각 작성해도 되지만 수식2와 같이 if 조건 then 중첩문을 이용하여 중첩되는 내용을 그룹문으로 만들고 개별적인 내용을 그룹 안에 if문으로 포함하여 문장을 작성할 수 있습니다.

수식2에서 먼저 $0 > C[1]+1$ 조건이 먼저 만족하고 나서 buy는 $C > 0$ 이 만족하면 실행되고 sell은 $C < 0$ 가 만족해야 실행된다. 우선 조건인 $0 > C[1]+1$ 이 만족하지 못하면 바로 빠져나가게 됩니다.

복잡한 수식의 경우 if 조건 then 중첩문을 이용하면 동일한 계산을 반복할 필요가 없으므로 계산시간을 단축하는 효과가 있습니다.

④ if 조건 then 실행1 else 실행2

수행해야 할 실행문이 다수가 있는 상황에서 주어진 조건이 True(참)값일 때 수행해야 할 실행문과 False(거짓)일 때 수행해야 할 수행문으로 분리되어 있을 경우 이용할 수 있는 구문입니다.

If~then~else ~문은 두 가지 상반된 if문으로 표시하는 것보다는 간편하게 사용할 수 있습니다.

금일시가가 전일증가보다 크면 매수 금일시가가 전일증가보다 크지 않으면 (작거나 같으면)매도	
<pre> #수식1 if 0 > C[1] then buy(); if 0 <= C[1] then sell(); </pre>	<pre> #수식2 if 0 > C[1]then buy(); else sell(); </pre>
금일시가가 전일증가보다 크면 매수하고 고가를 저장 금일시가가 전일증가보다 크지 않으면 매도하고 저가를 저장	

<pre> #수식A if 0 > C[1] then { buy(); var1 = H; } if 0 <= C[1] then { sell(); var2 = L; } </pre>	<pre> #수식B if 0 > C[1] then { buy(); var1 = H; } else { sell(); var2 = L; } </pre>
---	---

⑤ if 조건 then 실행1 else if 실행2 else 실행3 문

if 조건 then 실행1 else 실행2 문에서 False(거짓)일 경우를 세분화한 구분이라고 보면 됩니다.

양봉일 경우 1값 할당 음봉일 경우 -1값 할당 십자봉일 경우 0값 할당	
<pre> #수식1 if C > 0 then var1 = 1; if C < 0 then var1 = -1; if C == 0 then var1 = 0; </pre>	<pre> #수식2 if C > 0 then var1 = 1; else if C < 0 then var1 = -1; else var1 = 0; </pre>
양봉일 경우 1값과 고가 할당 음봉일 경우 -1값과 저가 할당 십자봉일 경우 0값과 종가 할당	

<pre> #수식A if C > 0 then { var1 = 1; var2 = H; } if C < 0 then { var1 = -1; var2 = L; } if C == 0 then { var1 = 0; var2 = C; } </pre>	<pre> #수식B if C > 0 then Begin var1 = 1; var2 = H; End else if C < 0 then begin var1 = -1; var2 = L; end else begin var1 = 0; var2 = C; end; </pre>
---	---

※ else문에서는 Begin~end로 범위를 지정할 경우 최종 else 위의 end는 문장이 끝이 아니므로 세미콜론 (;)을 붙이지 않고 최종 else 이후의 end에 세미콜론 (;)을 붙여야 합니다.

나) 반복 처리문

반복 처리문에는 for문과 While문 두가지가 있습니다. For문은 반복적인 계산작업을 자동화하고 특정한 계산작업을 일정한 횟수만큼 반복하고 while문은 특정한 조건이 만족하는 한 일정한 계산작업을 반복합니다.

For문은 반복되는 횟수를 지정하기에 결과값이 어느 정도는 예측이 가능하지만, While문은 While 조건이 만족하는 한 계속적으로 실행되지 때문에 종료되는 시점이나 결과값이 예측이 어렵습니다. 따라서 반복횟수를 지정할 수 있는 경우에는 For문을 사용하고 얼마나 반복해야 할지 예측이 되지 않을 때는 While 문을 사용하면 됩니다.

① For문

For 변수 = 초기값 to(downto) 최종값 Step 증감값

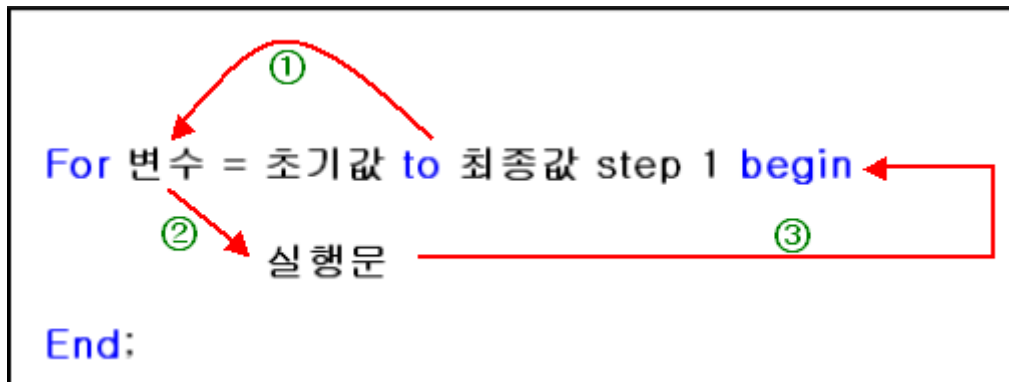
실행문

※ Step 1은 생략가능

For문은 변수에 최초 초기값을 할당하고 증감값만큼 증감시키면서 최종값 이하까지 할당하면서 실행문을 반복적으로 수행하는 제어문이다. Step이 1일 때는 생략할 수 있습니다.

변수에 초기값이 최종값보다 작다면 To를 사용하여 오름차순으로 할당할 수 있게 해주고 초기값이 최종값보다 크다면 downto를 사용해서 내림차순으로 할당되게 for문을 작성해 줍니다.

For문을 흐름을 그림으로 표현하면 아래와 같습니다.



① → ② → ③ 의 흐름을 최종값을 초과할 때 까지 반복합니다.

```

var : cnt(0),sum(0);

sum = 0;
For cnt = 0 to 9 {
5   sum = sum +c[cnt];
}

plot1(sum);
  
```

위 식은 최근 10개봉의 종가를 누적하는 식을 For문을 작성한 식입니다.

어떤 방식으로 10개봉의 종가를 누적하는지 살펴보면,

우선 변수 cnt에 최초로 0이 할당되면 cnt는 0값을 가지고 실행문으로 들어가게 됩니다. 그러면 C[cnt]는 C[0]이 되어 현재봉의 종가를 Sum변수에 할당합니다.

For문으로 돌아가 cnt에 1값이 할당되고 1값을 가지고 실행문으로 들어가 C[cnt]를 C[1]로 만들어 전봉의 종가를 sum에 누적합니다.

이렇게 1값씩 증가시키면서 9까지를 cnt에 할당하고 실행문이 수행되면 for문에서 빠져나가게 됩니다.

※ for문 위 줄의 `sum = 0;`은 for문이 시작하기 전에 값이 누적되는 변수를 초기화를 시켜 준 것입니다.

초기화를 해주지 않으면 이전 계산에서 누적된 값에 또 값을 누적하게 됩니다.

다시 다른 예로 흐름을 알아보면 아래의 식은 최근 10개봉 중에 고가가 저가대비 2% 이상 높은 봉이 몇 개나 있는지를 계산하는 식입니다.

```
For문 예제2 <지표>
Var : cnt(0),count(0);

count = 0;
For cnt = 0 to 9 Begin
5   if H[cnt] >= L[cnt]*1.02 Then
      count = count+1;
end;

plot1(count);
```

우선 초기값인 0이 변수 cnt에 할당되고 0을 가지고 실행문에 들어가면 $H[0] > L[0]$ 이라는 식이 되어 현재봉의 고가가 현재봉의 저가 대비 2%이상 큰지를 판단하게 됩니다. 크다면 count에 1값이 누적되고 작다면 다시 for문으로 올라가 cnt에 1값이 할당되고 실행문은 $H[1] > L[1]$ 이 되어 조건이 만족하는지 판단하게 됩니다. 이렇게 cnt에 9까지 할당되고 실행문을 수행한 다음 for문이 종료가 됩니다.

For문은 계산 문장이나 if~then문을 여러 번 나열하면서 작성 해도 될 문장을 간단히 몇 줄로 짧게 끝낼 수 있게 해줍니다. 아래와 같이 풀어서 작성되어 있는 내용을 for문을 이용해 짧게 작성했다고 보면 됩니다.

<pre>For문 예제1 풀머쓰기 <지표> Var : sum(0); sum = 0; sum = sum+C[0]; 5 sum = sum+C[1]; sum = sum+C[2]; sum = sum+C[3]; sum = sum+C[4]; 10sum = sum+C[5]; sum = sum+C[7]; sum = sum+C[8]; sum = sum+C[9]; 15plot1(sum);</pre>	<pre>For문 예제2 풀머쓰기 <지표> var : count(0); count = 0; if H[0] > L[0]*1.02 Then 5 count =count+1; if H[1] > L[1]*1.02 Then count =count+1; if H[2] > L[2]*1.02 Then count =count+1; 10if H[3] > L[3]*1.02 Then count =count+1; if H[4] > L[4]*1.02 Then count =count+1; if H[5] > L[5]*1.02 Then 15 count =count+1; if H[6] > L[6]*1.02 Then count =count+1; if H[7] > L[7]*1.02 Then count =count+1; 20if H[8] > L[8]*1.02 Then count =count+1; if H[9] > L[9]*1.02 Then count =count+1; 25plot1(count);</pre>
--	--

어떤 값이나 조건을 찾기 위해 for문을 이용해 식으로 작성한다면 실제 지정된 횟수만큼 루프를 돌리지 않아도 중간에 값을 찾으면 루프를 그만두고 빠져 나와야 할 경우가 있습니다. 랭귀지에서는 별도의 Break 함수가 없으므로 값이나 조건을 찾았다면 변수에 최종값 이상의 값을 할당하게 되면 루프를 빠져 나오게 됩니다.

```

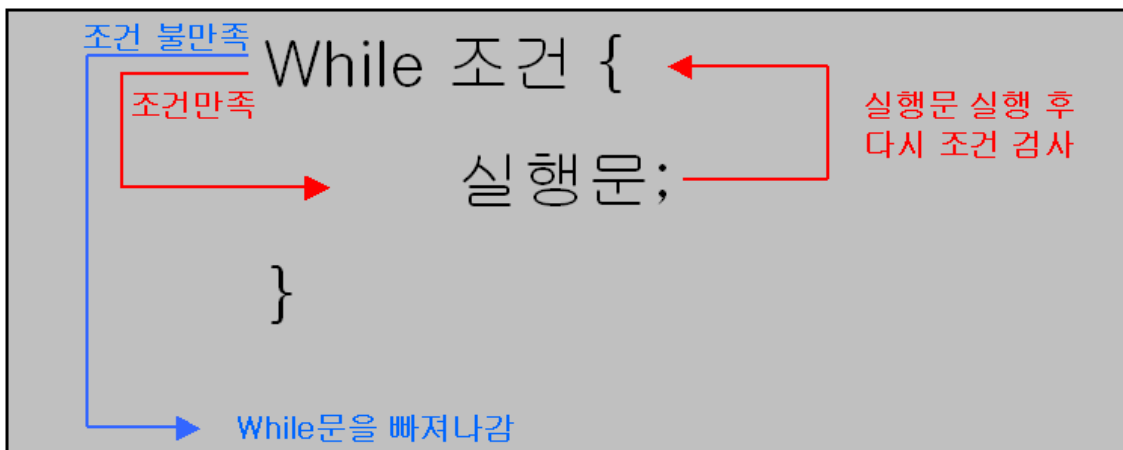
var : cnt(0), SHbar(0);

SHbar = -1;
for cnt = 0 to 100{
5   if H[cnt] == SwingHigh(1,H,3,3,100) Then{
        Shbar = cnt;
        cnt = 999;
    }
10 }
plot1(SHbar);

```

위의 식과 같이 최근 Swinghigh와 같은 값을 가지는 고가가 현재로부터 몇 봉 전에 발생했는지를 찾으면 Shbar변수에 cnt값을 저장하고 cnt에는 최종값보다 높은 999값을 주어 루프를 중지하게 하면 됩니다.

② While 문



While문은 조건이 만족하면 실행문을 수행하고 만족하지 않으면 Loop를 빠져나가는 제어문입니다. 즉 조건문이 만족하는 한 계속적으로 실행문을 수행하게 됩니다. 그러므로 계속적으로 while 조건이 참이 된다면 무한히 루프를 돌게 되어 컴퓨터가 다운이 되게 되므로 주의해서 사용해야 합니다.

예를 들어 현재봉의 증가보다 큰 증가가 현재로부터 몇 봉 전에 발생했었는지 알고자 할 때 For문을

쓰기에는 적당하지 않습니다. For문은 반복횟수를 지정하는 타입이라 이런 경우 반복 횟수를 지정하기에는 애매하기 때문이다. 물론 루프횟수를 적당히 크게 하여 for문으로 작성할 수는 있겠지만 그 횟수로도 부족할 경우가 있을 수 있습니다. 그러므로 루프 횟수를 지정하기 어려운 이러한 조건들은 While문을 이용해 값을 찾는 것이 적당합니다.

```

var : count(0);

count = 1;
While C > C[count] {
5   count = count+1;
}

plot1(count);

```

최초에 count는 1이므로 $C > C[1]$ 을 판단하여 종가가 전 봉의 증가보다 크면 실행문으로 내려가 count에 1을 더해 count는 2가 되고 다시 while 조건문으로 들어가 $C > C[2]$ 조건을 만들어 만족여부를 판단하여 실행문을 수행하던지 while문을 빠져나가게 됩니다. 이런 일련의 과정을 While조건문이 False가 될 때까지 수행하게 됩니다.

7) 함수 및 예약어

예스랭귀지에서는 많은 함수들이 제공이 됩니다. 각 함수에 대해서 간략히 설명하고 사용빈도가 높은 몇몇 함수에 대해서만 자세히 설명하겠습니다

가) 주문함수

주문함수	설명
Buy	신규매수 진입 (매도 포지션이 있을 경우 모두 청산 후 매수포지션으로 진입)
Sell	신규매도 진입 (매수 포지션이 있을 경우 모두 청산 후 매도포지션으로 진입)
ExitLong	매수포지션 청산
ExitShort	매도포지션 청산

예스랭귀지에서는 시스템과 관련한 다양한 함수들을 제공합니다.

기본적으로 신호를 발생하고 주문을 넣을 수 있는 주문함수가 있고 포지션의 손익관련 함수가 있고 진입과 청산의 시점과 관련된 함수가 있는 등 여러 함수가 존재합니다. 함수의 분류 중 포지션 함수와 전략성과함수로 분류된 함수 등이 이에 해당하며 모두 시스템식에서만 사용이 가능한 함수입니다.

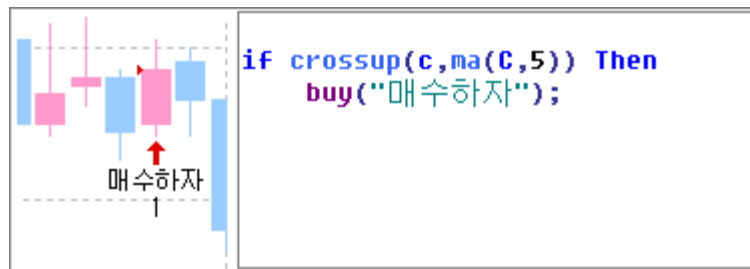
주문 함수는 매수 포지션에 대한 진입과 청산 함수와 매도 포지션에 대한 진입과 청산함수로 나뉘서 4가지로 제공됩니다.

신규로 매수를 진입하고자 하면 buy함수를 사용해야 하며 신규로 매도로 진입하고자 한다면 sell함수를 사용해야 합니다. 두 함수는 기존의 포지션이 없는 상태(무포지션)라면 신규로 포지션만 취하지만, 기존에 반대 포지션이 있다면 반대 포지션을 모두 청산하고 새로운 포지션을 취하게 된다. 만약 같은 포지션이 있는 상태라면 피라미딩(진입누적) 설정을 하지 않는 이상 추가로 진입하지는 않고 무시됩니다.

매수 포지션에 대한 청산만을 의도하면 ExitLong함수를 사용해야 하며 매도포지션에 대한 청산을 하고자 한다면 ExitShort함수를 이용해야 합니다. 그러므로 현물에서 사고팔고의 전략은 buy와 ExitLong 조합을 이용해야 합니다.

함수	구문
Buy	Buy("신호명", 신호타입, 가격조건, 수량)
Sell	Sell("신호명", 신호타입, 가격조건, 수량)
Exitlong	ExitLong("신호명", 신호타입, 가격조건, "진입명", 수량, 수량청산옵션)
Exitshort	ExitShort("신호명", 신호타입, 가격조건, "신호명", 수량, 수량청산옵션)

① 신호명



"신호명"은 신호의 이름을 지정할 수 매개변수입니다. 시스템식을 작성하다 보면 주문함수를 다수로 사용할 경우가 많은데 이름을 부여하면 신호 발생 시 차트에 어떤 신호로 발생했는지 이름과 같이 표시해서 구별을 용이하게 해주고 수식에서 진입별로 제어가 필요할 경우 해당 이름을 가져와 사용할 수 있게 됩니다. 신호명을 설정할 때는 큰따옴표(Double Quotation Mark)와 함께 이름을 설정하고 하나의 시스템 식에서 같은 이름을 복수의 주문함수에 설정할 수는 없습니다.

② 신호타입과 조건가격

신호타입은 주문타입이라고 불리기도 하지만 실제 주문가격을 지정하거나 하는 타입은 아닙니다.

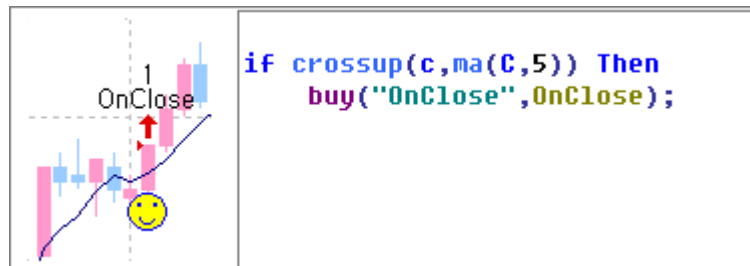
신호가 발생하고 실제 주문에 대한 가격은 시스템 트레이딩 설정창의 매매가격에서 지정한 가격으로 주문이

발생하므로 주문함수에서 주문가격을 설정할수는 없습니다.

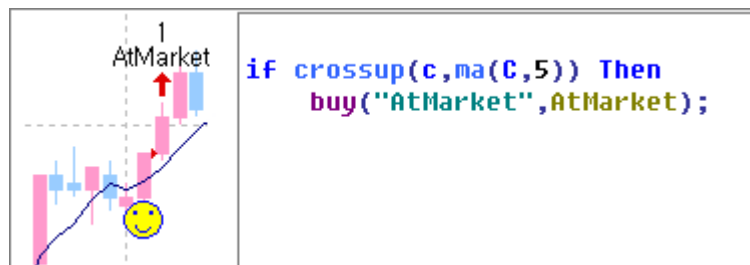
신호발생 시의 가격조건을 설정하는 부분입니다.

신호타입은 Onclose, Atmarket, Atstop, Atlimit으로 4가지가 제공되는데 예약어의 명칭이 증가, 시장가, 지정가 등으로 매매유형과 비슷하여 주문 가격을 설정하는 것으로 오해할 수 있습니다.

OnClose는 조건을 만족하는 봉의 종가를 신호의 가격으로 사용하는 타입입니다. 즉 if로 작성한 조건이 만족한 봉의 마지막 시세에 더 큰 의미를 부여한 타입입니다. 신호타입 뒤에 지정한 매개변수가 없다면 Onclose는 생략이 가능합니다.







AtMrket은 조건만족 봉의 다음 봉의 시가를 신호의 가격으로 사용하는 타입이다. If문은 봉 완성 시에 조건만족여부를 판단하게 되는데 봉이 완성되는 시점은 다음 봉의 시가가 수신될 때입니다. 그러므로 Atmarket은 if조건이 만족됐다고 판단하고 실제 주문이 발생하는 시점의 가격에 더 큰 의미를 부여한 신호타입이라고 보면 됩니다.



OnClose와 AtMarket은 따로 가격조건을 주는 타입이 아니므로 수량설정 시에는 가격조건을 Def로 작성해 주면 됩니다.

AtSop과 AtLimit은 최근 완성 봉 기준으로 가격을 설정한 후에 시세가 가격조건을 만족하는 즉시 신호가 발생하는 타입이다.

	AtStop	AtLimit
--	--------	---------

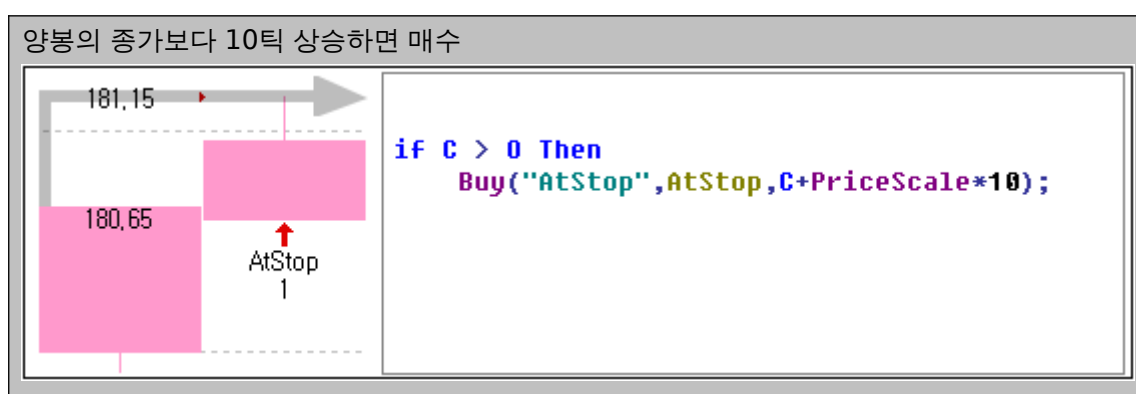
매수주문 Buy ExitShort		
매도주문 Sell ExitLong		

가격조건만 만족하면 봉이 아직 완성되지 않았다고 해도 즉시 신호가 발생하므로 일반적으로 조건만족 즉시 주문하는 타입으로 알려져 있습니다.

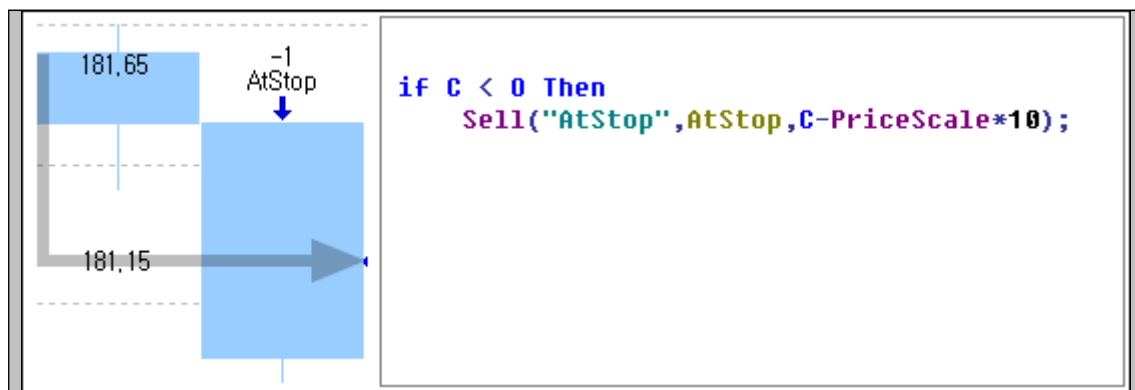
다만 혼동하지 말아야 할 점은 봉 미완성시에 if 조건 문이 만족하는 즉시가 아니고 설정된 가격조건이 만족하는 즉시라는 부분입니다. 또한 봉완성시 가격조건을 설정하고 다음 봉의 시세를 감시하므로 가장최근 완성된 봉을 기준으로 합니다. 또한 가격을 설정해야 하는 타입이므로 사용자가 조건이 되는 가격을 지정해 주어야 하며 def로 작성해서는 안 됩니다.

Atstop과 Atlimit의 가격조건 감시는 설정 후 다음 하나의 봉에서만 감시를 하게 되고 다음 봉이 가격조건에 맞는 시세가 발생하지 않으면 설정은 해제가 됩니다.

AtStop은 매수주문 함수인 Buy나 ExitShort의 경우 지정된 가격조건 이상의 시세가 발생하면 신호가 발생하게 되고 Sell과 ExitLong같은 매도주문 함수에서는 지정된 가격조건 이하의 시세가 발생하면 신호가 발생하게 됩니다.

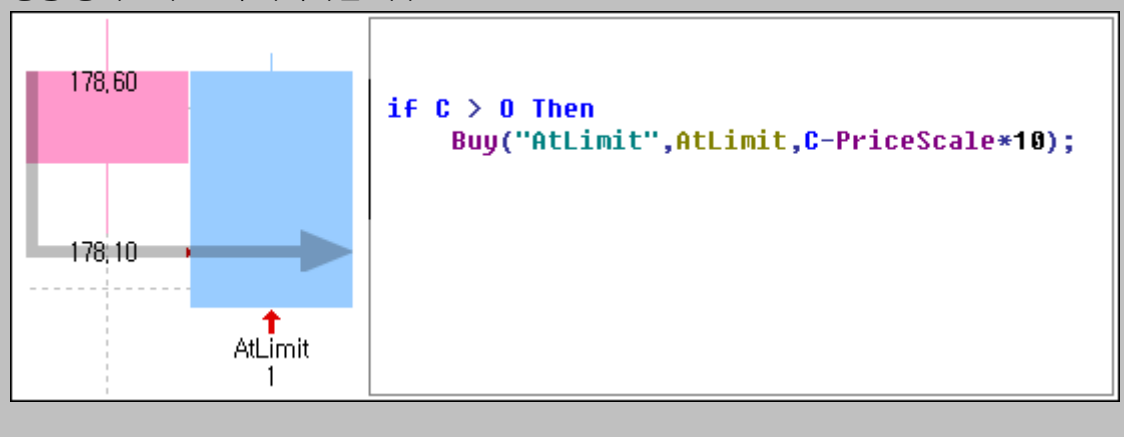


음봉의 종가보다 10틱 하락하면 매도

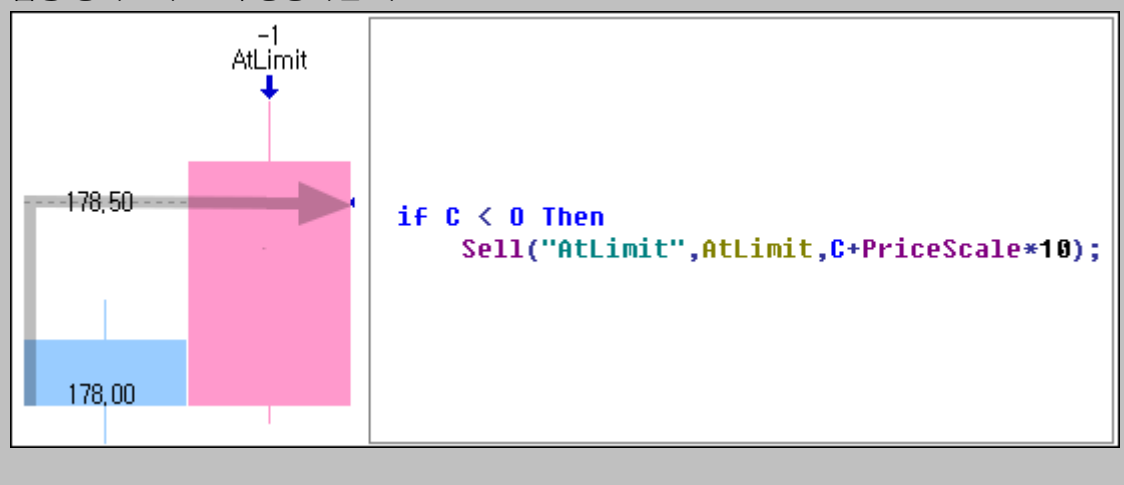


AtLimit은 AtStop과는 반대로 Buy나 ExitShort의 경우 지정된 가격조건 이하의 시세가 발생하면 신호가 발생하게 하고 Sell과 exitlong에서는 지정된 가격조건 이상의 시세가 발생하면 신호가 발생하게 됩니다.

양봉 증가보다 10틱 하락하면 매수



음봉 증가보다 10틱 상승하면 매도



③ 진입명

진입명은 청산함수에서만 설정하는 매개변수입니다. 여러 진입들 중에 특정 진입명으로 진입한 경우에만 청산함수가 발동하게 작성하고자 한다면 진입명을 설명하면 됩니다.

진입명은 큰따옴표(Double Quotation Mark)안에 설정하면 되고 설정하지 않으려면 큰따옴표(Double Quotation Mark)만 표시하면 됩니다.

→ 매수1이라는 진입명으로 매수진입 후 최고가에서 1포인트 하락하면
전체수량에서 1계약 청산
ExitLong("Tr",AtStop,highest(H,BarsSinceEntry)-1,"매수1",1,1);

→ 매수 진입 후 최고가에서 1포인트 하락하면 전체수량에서 1계약 청산
(진입명 관계없음)
ExitLong("Tr",AtStop,highest(H,BarsSinceEntry)-0.5, " ",1,1);

④ 수량

주문함수에서는 수량을 설정할 수 있습니다. 원하는 수량을 상수로 입력하거나 혹은 계산식이나 계산에 의해 할당된 값을 넣어도 됩니다. 진입 함수에 수량을 설정하지 않으면 시스템 트레이딩 설정창의 수량/비용 탭에서 설정한 수량으로 진입하게 되고 청산 함수에 수량을 설정하지 않으면 포지션의 잔량을 모두 청산하게 됩니다.

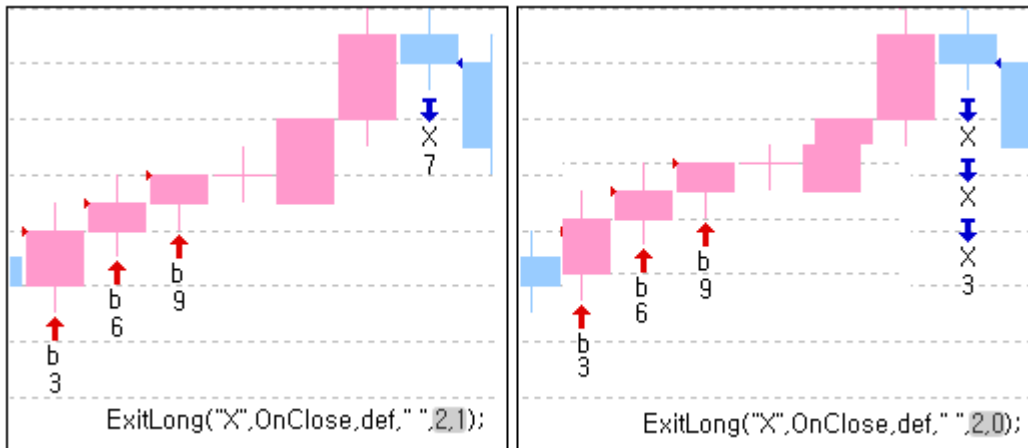
⑤ 수량옵션

수량옵션은 청산함수에서만 제공되며 세개의 옵션이 제공됩니다.

0 : 각각의 진입신호에서 일괄적으로 지정된 수량만큼을 청산

1 : 현재 보유한 전체 수량에서 지정된 수량만큼만 청산

2 : 현재 보유한 전체 수량에서 지정한 수량만큼만 청산하고 청산함수가 진입횟수에 관계없이 조건만족시 반복적으로 실행하게 함.



위의 그림과 같이 수량을 1이라고 설정한 상태에서 수량옵션을 1로 설정하면 전체 수량 9주(계약)에서 지정된 수량만큼만 청산이 되므로 최종 수량이 7주(계약) 남게 됩니다. 수량옵션을 0으로 설정하게 되면 각각의 진입신호에서 설정된 2주(계약)씩 청산하므로 청산수량은 6주(계약)가 되고 3주(계약) 남게 됩니다. 수량옵션 0은 진입이 피라미딩 됐을 때만 의미가 있으므로 진입을 피라미딩으로 하는 시스템이라면 유의해서 설정하도록 해야 합니다. 0 옵션은 청산수량을 가늠하기가 쉽지 않으므로 거의 사용되지 않고 1을 사용하여 전체수량에서 지정된 수량을 차감하는 방식으로 청산식을 작성하는 것이 일반적입니다. 수량옵션3은 1과 같습니다. 다른점은 청산함수가 진입횟수만큼만 동작하는데 진입횟수와 관계없이 반복적으로 발생하고자 할 때 사용하게 됩니다.

나) 주문예약어

OnClose	주문함수 신호타입 예약어 – 종가
AtMarket	주문함수 신호타입 예약어 – 다음봉 시가
AtLimit	주문함수 신호타입 예약어 – 지정가
AtStop	주문함수 신호타입 예약어 – 역지정가
PercentStop	강제청산 예약어 – Percent 설정
PointStop	강제청산 예약어 – Point 설정

다) 수확함수

Avg, Avglist	리스트에 있는 모든 항목의 평균값
Max, Maxlist	리스트의 항목 중 최대값
Maxlist2	리스트의 항목 중 두번째 큰 값
NthMaxList	리스트의 항목 중 N번째 큰 값
Min, MinList	리스트의 항목 중 최소값
MinList2	리스트의 항목 중 두번째 작은 값
NthMinList	리스트의 항목 중 N번째 작은 값

SumList	리스트에 있는 모든 항목의 합
Abs, AbsValue	입력된 값의 절대값
Mod	나눈 나머지
Round	지정된 자릿수 만큼 반올림한 값
Ceiling	소수 부분을 올림한 정수 값
Floor	소수 부분을 내림한 정수 값
FracPortion	소수아래의 값
Int, IntPortion	정수값
Neg	음의 값으로 반환
Pos	양의 값으로 반환
Pie	원주율
Pow	누승
Power	누승
SqRt	제곱근
Square	제곱
SquareRoot	제곱근
Random	0과 num 사이의 난수 값
Exp, ExpValue	승수로 거듭 제공하는 자연로그의 밑인 e를 반환하는 함수
Log	자연로그
Log10	상용로그
Cos, Cosine	코사인
CosH	하이퍼블릭코사인
Acos	아크코사인
Sin, Sine	사인
SinH	하이퍼블릭 사인
Asin	아크사인
Tan, Tangent	탄젠트
TanH	하이퍼블릭 탄젠트
Atan ,ArcTangent	아크탄젠트
Cotangent	코탄젠트

라) 분석함수

수식을 작성하다 보면 봉에 대한 정보를 이용하여 조건을 만드는 경우가 많이 발생합니다. 분석함수는 이런 비교 분석을 더욱 용이하게 해주는 함수입니다.

DayClose(N)	N일전 일간의 증가
-------------	------------

DayHigh(N)	N일전 일간의 고가
DayLow(N)	N일전 일간의 저가
DayOpen(N)	N일전 일간의 시가
DayOi(N)	N일전 일간의 미결제약정
DayVolume(N)	N일전 일간의 거래량
CrossUp(Value1,Value2)	Value1이 Value2를 상향돌파
CrossDown(Value1,Value2)	Value1이 Value2를 하향이탈
Highest(Value, Length)	Length 기간동안 Value 중 최고 값
NthHghest(Nth, Value, Length)	Length 기간동안 Value 중 Nth번째 큰 값
NthHighestBar(Nth, Value, Length)	Length 기간동안 Value 중 Nth번째 큰 봉의 위치
Lowest(Value, Length)	Length 기간동안 Value 중 최저 값
NthLowest(Nth, Value, Length)	Length 기간동안 Value 중 Nth번째 작은 값
NthLowestBar(Nth, Value, Length)	Length 기간동안 Value 중 Nth번째 작은 봉의위치
TimeClose(StartTime,EndTime)	지정된 시간 동안의 종가
TimeCloseEx(StartTime,EndTime)	지정된 시간 동안의 종가(변동)
TimeOpen(StartTime,EndTime)	지정된 시간 동안의 시가
TimeOpenEx(StartTime,EndTime)	지정된 시간 동안의 시가(변동)
TimeHigh(StartTime,EndTime)	지정된 시간 동안의 고가
TimeHighEx(StartTime,EndTime)	지정된 시간 동안의 고가(변동)
TimeLow(StartTime,EndTime)	지정된 시간 동안의 저가
TimeLowEx(StartTime,EndTime)	지정된 시간 동안의 저가(변동)
Ema(Value, Length)	지수이동평균
Ma(Value, Length)	단순이동평균
Iff(조건,참,거짓)	조건을 만족하면 참값을 아니면 거짓값을 리턴
NextBarOpen	다음 봉 시가
NextBarSdate	다음 봉 시작날짜
NextBarStime	다음봉 시작시간
Data1~Data99	타종목 참조함수
DayOfWeek(Date)	주어진 날짜의 요일 반환(월:1,화:2,수:3,목:4,금:5)
MinutesToTime(Mimutes)	주어진 분단위를 시간으로 표시(HHMMSS)

TimeToMinutes(Time)	주어진 시간을 밤12시 이후 경과된 분단위로 표시
DateToJulian(Date)	주어진 날짜를 Julian date 로 변경
JulianToDate(Juliandate)	Julian date를 날짜로 변경(YYYYMMDD)
CurrentDate	컴퓨터의 현재 날짜
CurrentTime	컴퓨터의 현재 시간
IsNan(Value)	Value 값이 N/A면 True를 리턴
Nan	N/A를 할당
LastBarOnChart	차트의 마지막 봉이면 1값 반환
LastCalcMMTime	차트의 마지막 봉의 TimeToMinutes을 반환
InStr("String1","String2")	String1문자열에서 String2문자의 위치값을 리턴
LeftStr("String", Num)	좌측에서 Num번째 문자열까지 리턴
RightStr("String", Num)	우측에서 Num번째 문자열까지 리턴
LowerStr("String")	문자열을 소문자로 변환
UpperStr("String")	문자열을 대문자로 변환
MidStr("String",Num1,Num2)	문자열을 중 Num1~Num2번째 사이의 문자 리턴
StrToNum("String")	문자열을 숫자로 변환
NumToStr(Figure,N)	소수점 N자리까지 숫자를 문자로 변환
EL_DateStr(DD,MM,YYYY)	YYYYMMDD형식의 문자열로 변환
StrLen	지정한 문자열의 길이를 리턴
MaxBarsBack	수식 계산에 필요한 필수 데이터의 수량
Spaces(Num)	Num으로 지정한 수만큼 빈공간을 만든다.
NewLine	새로운 행을 삽입한다.
NoValue	화면 표시 제거
Extremes(Price, Length, HiLo, oExtremeVal, oExtremePosRaw)	극단값과 위치
HarmonicMean(Price, Length)	조화평균
Mode(Price, Length, hiLo)	최빈값
NthExtremes(Price, Length, N, HiLo, oExtremeVal, oExtremePosRaw)	Nth번째 극단값과 위치
SummationRec(Price, Length)	역수합
SummationSqr(Price, Length)	제곱합
TrimMean(Price, Length, TrimPct)	극단치 제외 평균

Variance(Price, Length, DataType)	모집단의 분산 혹은 표본집단의 분산
Covar(IndepArray,DepArray,Length)	공분산
AvgDeviation(Price,Length)	편차평균
DevSqrd(Price,Length)	편차 제곱합
StdError(Price, Length)	표준오차
StdError2(Indep, Dep, ArraySz)	두 값의 표준오차
CoefficientR(Indep,Dep,Length)	피어슨 상관계수R
RSquare(Indep, Dep, Length)	피어슨 상관계수R의 제곱
Skew(Price, Length)	왜도
LinReg(Price, Length, TgtPos, oLRSlope, oLRAngle,oLRIntercept, oLRValueRaw)	선형회귀선의 기울기, 각도, 절편, 예측값
LinReg2(Indep,Dep, Length, TgtPos, oLRSlope, oLRAngle, oLRIntercept, oLRValueRaw)	두 값의 선형회귀선의 기울기, 각도, 절편, 예측값
LinRegForecast(Price, Length, TgtPos)	선형회귀를 기반으로 주어진 X값에 대한 예측값
LinRegIntercept(Price, Length)	선형회귀선의 절편
LinRegSlope(Price, Length)	선형회귀선의 기울기
NormalCumDensity(Price, Length)	정규누적분포
NormalDensity(Price, Length)	정규분포
Standardize(Price, Length, NumDevs)	표준편차 및 평균에 따라 표준화된 값을 반환
InsertArray(Array,location, 삽입할값)	배열에 값 삽입.
AverageArray(PriceArray,Size)	배열의 평균
ExtremesArray(PriceArray,Size,HiLo, oExtremeVal,oExtremePosRaw)	배열의 극단값과 위치
HarmonicMeanArray(PriceArray, Size)	배열의 조화평균
HighestArray(PriceArray, Size)	배열의 최고가
LowestArray(PriceArray, Size)	배열의 최저가
ModeArray(PriceArray, Size, hiLo)	배열의 최빈값
NthExtremesArray(PriceArray, Size, N, HiLo, oExtremeVal, oExtremePosRaw)	배열의 Nth번째 극단값과 위치
NthHighestArray(PriceArray, Size, N,oValue,oIndex)	배열의 Nth번째 최고값과 위치
NthLowestArray(PriceArray, Size, N,oValue,oIndex)	배열의 Nth번째 최저값과 위치
SummationArray(PriceArray, Size)	배열의 합계

SummationRecArray(PriceArray, Size)	배열의 역수값의 합계
SummationSqrArray(PriceArray, Size)	배열의 제곱합
TrimMeanArray(PriceArray, Size, TrimPct)	배열의 극단치 제외 평균
VarianceArray(PriceArray, Size, DataType)	배열의 모집단 분산 혹은 표본집단 분산
CovarArray(IndepArray, DepArray, Size)	두 배열의 공분산
AvgDeviationArray(PriceArray, Size)	배열의 편차 평균
DevSqrArray(PriceArray, Size)	배열의 편차 제곱합
StdErrorArray(PriceArray, Size)	배열의 표준오차
StdErrorArray2(IndepArray, DepArray, ArraySz)	두 배열의 표준오차
StandardDevArray(PriceArray, Size, DataType)	배열의 모집단 표준편차 혹은 표본집단 표준편차 값을 반환
CoefficientRArray(IndepArray, DepArray, Size)	두 배열의 피어슨 상관계수R
RSquareArray(IndepArray, DepArray, Size)	두 배열의 피어슨 상관계수R의 제곱
CorrelationArray(IndepArray, DepArray, Size)	두 배열의 상관계수
SortArray(PriceArray, Size, HiLo)	1차원배열 정렬
MedianArray(PriceArray, Size)	배열을 중간값 반환
QuartileArray(QRank, PriceArray, Size)	배열의 사분위수 반환
NumericRankArray(PriceToRank, PriceArray, Size, SortOrder)	배열을 정렬 후 지정한값의 순위를 반환
PercentRankArray(PriceToRank, PriceArray, Size)	배열에서 지정한 값의 백분율
PercentileArray(PctRank, PriceArray, Size)	배열의 지정한 순위율에 해당하는 값
KurtosisArray(PriceArray, Size)	배열의 첨도
SkewArray(PriceArray, Size)	배열의 왜도
LinRegArray(PriceArray, Size, TgtPos, oLRSlope, oLRAngle, oLRIntercept, oLRValueRaw)	배열의 선형회귀선의 기울기, 각도, 절편, 예측값
LinRegArray2(IndepArray, DepArray Size, TgtPos, oLRSlope, oLRAngle, oLRIntercept, oLRValueRaw)	두 배열의 선형회귀선의 기울기, 각도, 절편, 예측값
LinRegForecastArray(PriceArray, Size, TgtPos)	배열의 선형회귀를 기반으로 주어진 X값에 대한 예측값
LinRegInterceptArray(PriceArray, Size)	배열의 선형회귀선의 절편
LinRegSlopeArray(PriceArray, Size)	배열의 선형회귀선의 기울기
NormalCumDensityArray(PriceArray, Size)	배열의 정규누적분포
NormalDensityArray(PriceArray, Size)	배열의 정규분포

StandardizeArray(PriceArray, Size, NumDevs)	배열의 표준편차 및 평균에 따라 표준화된 값을 반환
Sort2DArray(Array, RowSize, ColSize, HiLo)	2차원배열 정렬, 지정한 범위를 0행을 기준으로 내림차순/오름차순으로 정렬.
SortArray2D(Array, RowSize, ColSize, HiLo, RowNo, Col No)	2차원배열 정렬, 시작행열을 기준으로 범위를 지정하고 시작행을 기준으로 내림차순/오름차순으로 정렬

마) 출력함수

Plot1~Plot99	지표 출력
FixPlotShift	Plot 좌우이동
PlotBaseLine1 ~ PlotBaseLine99	기준선 출력
PlotPaintBar	강조 출력
Print	지정된 데이터를 외부로 추출
MessageLog	지정된 데이터를 디버깅창에 표시
ClearDebug	디버깅창의 내용을 모두 삭제
PlaySound	소리 출력
Alert	지정한 내용을 경보창으로 표시
AlertEnabled	경보설정의 상태
SetAlert	경보 설정
NoPlot	결과 표시 제거
RGB	RGB값을 조합하여 색상 지정

바) 추세선출력함수

TL_New	새로운 추세선을 기본차트 영역에 생성
TL_New_Self	새로운 추세선을 지표가 적용된 영역에 생성
TL_Delete	지정한 추세선 개체를 삭제
TL_SetBegin	지정한 추세선 개체의 시작점을 새로 설정
TL_SetEnd	지정한 추세선 개체의 끝점을 새로 설정
TL_SetColor	지정한 추세선 개체의 색상을 설정
TL_SetSize	지정한 추세선 개체의 굵기를 설정
TL_SetStyle	지정한 추세선 개체의 선의 형태를 설정
TL_SetExtLeft	지정한 추세선 개체의 왼쪽 확장을 설정
TL_SetExtRight	지정한 추세선 개체의 오른쪽 확장을 설정

TL_SetDrawMode	지정한 추세선 개체의 반전색 표시를 설정
TL_GetBeginDate	지정한 추세선 개체의 시작점의 날짜를 리턴
TL_GetBeginTime	지정한 추세선 개체의 시작점의 시간을 리턴
TL_GetBeginVal	지정한 추세선 개체의 시작점(Y축)의 값을 리턴
TL_GetEndDate	지정한 추세선 개체의 끝점의 날짜를 리턴
TL_GetEndTime	지정한 추세선 개체의 끝의 시간을 리턴
TL_GetEndVal	지정한 추세선 개체의 끝점의 값(Y축)을 리턴
TL_GetValue	지정한 추세선 개체의 특정일 특정시간의 값을 리턴
TL_GetExtLeft	지정한 추세선 개체의 왼쪽 확장 여부를 리턴
TL_GetExtRight	지정한 추세선 개체의 오른쪽 확장 여부를 리턴
TL_GetColor	지정한 추세선 개체에 설정 된 색상값을 리턴
TL_GetSize	지정한 추세선 개체에 설정된 굵기를 리턴
TL_GetStyle	지정한 추세선 개체에 설정된 선의 형태를 리턴
TL_GetDrawMode	지정한 추세선 개체에 반전색 설정을 리턴

사) 텍스트출력함수

Text_New	기본차트 영역에 새로운 텍스트 개체를 생성
Text_New_Self	지표 영역에 새로운 텍스트 개체를 생성
Text_Delete	지정한 텍스트 개체를 삭제
Text_SetColor	지정한 텍스트 개체의 색상을 설정
Text_SetLocation	지정한 텍스트 개체의 위치를 변경
Text_SetString	지정한 텍스트 개체의 출력문자열을 새로운 문자열로 설정
Text_SetStyle	지정한 텍스트 개체의 수평 및 수직 정렬 위치를 설정
Text_SetBold	지정한 텍스트 개체를 굵게 설정.
Text_SetFont	지정한 텍스트 개체의 폰트를 설정.
Text_SetSize	지정한 텍스트 개체의 크기를 설정.
Text_GetColor	지정한 텍스트 개체의 색상값을 리턴
Text_GetDate	지정한 텍스트 개체의 출력 봉의 날짜를 리턴
Text_GetTime	지정한 텍스트 개체의 출력 봉의 시간을 리턴
Text_GetValue	지정한 텍스트 개체의 출력위치값(Y축)을 리턴
Text_GetHStyle	지정한 텍스트 개체의 수평 정렬 위치를 리턴
Text_GetVStyle	지정한 텍스트 개체의 수직 정렬 위치를 리턴
Text_GetString	지정한 텍스트 개체의 출력문자열을 리턴

아) 박스출력함수

Box_New	기본차트 영역에 새로운 박스 개체를 생성
Box_New_Self	지표 영역에 새로운 박스 개체를 생성
Box_Delete	지정한 박스 개체를 삭제
Box_SetColor	지정한 박스 개체의 색상을 설정
Box_SetStyle	지정한 박스 개체의 선형태 설정
Box_SetSize	지정한 박스 개체의 선굵기 설정
Box_SetFill	지정한 박스 개체의 내부채움 설정
Box_SetExtFill	지정한 박스 개체의 내부채움을 차트전체영역으로 확장을 설정
Box_SetBegin	지정한 박스 개체의 시작점을 설정
Box_SetEnd	지정한 박스 개체의 끝점을 설정
Box_GetColor	지정한 박스 개체의 색상을 리턴
Box_GetStyle	지정한 박스 개체의 선형태를 리턴
Box_GetSize	지정한 박스 개체의 선굵기를 리턴
Box_GetFill	지정한 박스 개체의 내부채움을 리턴
Box_GetExtFill	지정한 박스 개체의 내부채움의 확장여부를 리턴
Box_GetBeginDate	지정한 박스 개체의 시작점 날짜를 리턴
Box_GetBeginTime	지정한 박스 개체의 시작점 시간을 리턴
Box_GetBeginVal	지정한 박스 개체의 시작점 가격을 리턴
Box_GetEndDate	지정한 박스 개체의 끝점 날짜를 리턴
Box_GetEndTime	지정한 박스 개체의 끝점 시간을 리턴
Box_GetEndVal	지정한 박스 개체의 끝점 가격을 리턴

자) 타원출력함수

Circle_New	기본차트 영역에 새로운 타원 개체를 생성
Circle_New_Self	지표 영역에 새로운 타원 개체를 생성
Circle_Delete	지정한 타원 개체를 삭제
Circle_SetColor	지정한 타원 개체의 색상을 설정
Circle_SetStyle	지정한 타원 개체의 선형태 설정
Circle_SetSize	지정한 타원 개체의 선굵기 설정
Circle_SetFill	지정한 타원 개체의 내부채움 설정
Circle_SetBegin	지정한 타원 개체의 시작점을 설정
Circle_SetEnd	지정한 타원 개체의 끝점을 설정
Circle_GetColor	지정한 타원 개체의 색상을 리턴
Circle_GetStyle	지정한 타원 개체의 선형태를 리턴

Circle_GetSize	지정한 타원 개체의 선굵기를 리턴
Circle_GetFill	지정한 타원 개체의 내부채움을 리턴
Circle_GetBeginDate	지정한 타원 개체의 시작점 날짜를 리턴
Circle_GetBeginTime	지정한 타원 개체의 시작점 시간을 리턴
Circle_GetBeginVal	지정한 타원 개체의 시작점 가격을 리턴
Circle_GetEndDate	지정한 타원 개체의 끝점 날짜를 리턴
Circle_GetEndTime	지정한 타원 개체의 끝점 시간을 리턴
Circle_GetEndVal	지정한 타원 개체의 끝점 가격을 리턴

차) 그리드출력함수

Grid_New	새로운 그리드(표) 생성
Grid_SetBackColor	그리드(표) 배경색 설정
Grid_SetBorderColor	그리드(표) 셀경계선 색상 설정
Grid_SetBorderWidth	그리드(표) 셀경계선 두께 설정
Grid_SetFrameColor	그리드(표) 외곽선 색상 설정
Grid_SetFrameWidth	그리드(표) 외곽선 두께 설정
Grid_SetPosition	그리드(표) 위치 설정
Grid_SetTextBold	그리드(표) 텍스트 굵게 설정
Grid_SetTextSize	그리드(표) 텍스트 크기 설정
Grid_SetTransparency	그리드(표) 투명도 설정
Grid_Cell	지정한 셀의 옵션들 일괄 설정
Grid_CellSetBackColor	지정한 셀의 배경색 설정
Grid_CellSetHeight	지정한 셀의 높이 설정
Grid_CellSetWidth	지정한 셀의 넓이 설정
Grid_CellSetText	지정한 셀의 텍스트 설정
Grid_CellSetTextBold	지정한 셀의 텍스트 굵게 설정
Grid_CellSetTextColor	지정한 셀의 텍스트 색상 설정
Grid_CellSetTextSize	지정한 셀의 텍스트 크기 설정
Grid_CellSetTextHAlign	지정한 셀의 텍스트 수평정렬 설정
Grid_CellSetTextVAlign	지정한 셀의 텍스트 수직정렬 설정
Grid_Clear	지정한 셀영역 삭제
Grid_MergeCells	지정한 셀영역 병합
Grid_Delete	지정한 그리드(표) 삭제

카) 잔고함수

GetAccount	계좌목록의 계좌 중 지정한 순번의 계좌번호를 리턴
GetAccountStatus	가원장 구축상태를 리턴(구축 :1, 미구축 : 0)
GetAccountType	계좌종류,(1:위탁, 2:저축, 3:선/옵선)
GetNumAccounts	보유계좌의 총갯수
GetNumPositions	지정한 계좌의 보유 총 종목수
GetOpenOrderInitialMargin	지정한 계좌의 위탁증거금
GetPositionSymbol	계좌의 종목들 중에서 지정한 순번의 종목코드(단축코드)를 리턴
GetPositionAveragePrice	지정한 계좌의 지정한 종목의 평균가
GetPositionMarketValue	지정한 계좌의 지정한 종목의 현재가
GetPositionOpenPL	지정한 계좌의 지정한 종목의 평가손익
GetPositionQuantity	지정한 계좌의 지정한 종목의 보유수량
GetUnclearedDeposits	지정한 계좌의 예수금 (선/옵인 경우 예탁총액, 위탁/저축인 경우 예수금)

타) 포지션 함수

AvgEntryPrice	평균 진입가격
BarsSinceEntry	포지션 진입 이후 경과된 봉수
BarsSinceExit	포지션 청산 이후 경과된 봉수
CommissionMethod	수수료 설정 상태
CurrentContracts	포지션의 청산되지 않은 진입 계약수
CurrentEntries	포지션의 청산되지 않은 진입 횟수
EntryCommission	진입 수수료
EntryDate	진입 날짜
EntryName	진입명을 문자열로 반환
EntryPrice	진입 가격
EntrySlippage	진입 슬리피지
EntryTime	진입 시간
ExitCommission	청산 수수료
ExitDate	청산 날짜
ExitName	청산명을 문자열로 반환
ExitPrice	청산 가격
ExitSlippage	청산 슬리피지
ExitTime	청산 시간

IsEntryName	지정된 문자열이 포지션의 진입명과 같은지 판단
IsExitName	지정된 문자열이 포지션의 청산명과 같은지 판단
MarketPosition	포지션 상태
MaxContracts	포지션의 최대보유 계약수
MaxEntries	포지션의 최대 진입수
MaxPositionLoss	포지션의 최대 손실액
MaxPositionProfit	포지션의 최대 수익액
OpenPositionProfit	미청산 포지션의 손익액
PositionProfit	포지션의 손익액
SlippageMethod	슬리피지 설정상태
LatestEntryName	가장 최근 진입명
LatestEntryPrice	가장 최근 진입의 가격
LatestExitName	가장 최근 청산명
LatestExitPrice	가장 최근 청산의 가격

파) 전략 성과 함수

AvgBarsLosTrade	종결된 거래 중 손실거래의 평균 봉 개수
AvgBarsWinTrade	종결된 거래 중 수익거래의 평균 봉 개수
GrossLoss	종결된 거래 중 손실거래의 손실 총합
GrossProfit	종결된 거래 중 수익거래의 수익 총합
LargestLosTrade	종결된 손실거래 중 최대 손실금액
LargestWinTrade	종결된 손실거래 중 최대 수익금액
MaxConsecLosers	종결된 거래 중 최대 연속 손실거래 횟수
MaxConsecWinners	종결된 거래 중 최대 연속 수익거래 횟수
MaxContractsHeld	포지션 중에서 최대 보유한 계약수
MaxIDDDrawDown	최대 자본 인하분
NetProfit	종결된 모든 거래의 손익금액
NumLosTrades	종결된 거래 중 손실거래의 횟수
NumWinTrades	종결된 거래 중 수익거래의 횟수
PercentProfit	승률
TotalBarsLosTrades	종결된 거래 중 손실거래의 총 봉수
TotalBarsWinTrades	종결된 거래 중 수익거래의 총 봉수
TotalTrades	종결된 총 거래 횟수

하) 종목검색함수

종목 검색 함수는 지정된 조건에 만족하는 종목을 찾으라는 출력명령 함수이므로 종목검색 편집창에서만 사용할 수 있습니다.

Find	종목 검색
------	-------

Find함수는 하나의 매개변수를 가지게 됩니다.

Find 함수 안에 특정 데이터나 계산식 등을 입력하면 if조건에 충족되는 종목을 검색하여 종목검색화면에 리스트해 주고 검색결과 항목에 find함수에서 지정한 내용을 표시해 줍니다.

```

Var: Formula0(0);

if ma(M[1],4)>2000000000 and V<V[1]*4.00 and money>=1000000000 and C<20000 and ma(C[1],5)>=ma(C[2],5) and ma(C,20)>=ma(C[1],20) and C<ma(C,20)*1.20 and C>=ma(C,5) and C>=500 Then
Find(StochasticsK(10,5)); 검색결과 출력값지정

```

검색	S	종목명 (209)	검색결과값	현재가	전일대비(%)
		(K) 중국원양자원	87.42	9,030 ▲	130 (1
		(K) 메리츠화재	80.53	8,290 ▲	40 (0
		(K) 한화손해보험	95.85	10,300 ▲	200 (1
		(K) 홀국화재	95.59	6,600 ▲	70 (1
		(K) 한진해운홀딩스	92.06	19,500 ▲	600 (3
		(K) 경남기업	81.06	8,750 ▲	120 (1
		(K) 유니온	78.79	5,800 ▲	120 (2
		(K) 동부하이텍	67.28	12,650 ▲	100 (0
		(K) 중외제약	86.89	17,600 ▲	200 (1
		(K) KG케미칼	70.95	10,850 ▲	50 (0
		(K) 대한전선	79.94	10,300 ▼	250 (2
		(K) 동양메이저	80.13	2,610 ▼	20 (0
		(K) 종근당	87.47	18,200 ▲	100 (0
		(K) 대상	65.35	9,490 ▲	170 (1
		(Q) 무림SP	72.65	2,300 ▲	35 (1
		(Q) 미화공영	58.79	4,990 ▲	55 (1
		(K) 삼익약기	79.05	1,075 ▲	10 (0

특별히 검색결과로 보고자 하는 값이 없을 경우 find(1);로 작성하면 됩니다.

9) 사용자함수(외장함수)로 제공되는 기본함수

기본으로 제공되는 함수들 중에 사용자함수(외장함수)로 제공되는 함수들입니다.

내장함수와는 다르게 수식내용을 볼 수 있습니다.

분석함수 종류	
Accum(Value)	Value값을 누적
AccumN(Value, Length)	Value값을 Length 기간동안 누적
Correlation(A,B,Period)	상관관계, Period기간 동안 A와 B의 상관관계
CloseD(N)	N일전 일간 증가
HighD(N)	N일전 일간 고가
LowD(N)	N일전 일간 저가
OpenD(N)	N일전 일간 시가
DayIndex	분봉에서 당일 봉의 번호
CountIF(condition,Period)	최근 Period기간동안 Condition이 몇 번 만족했는지 반환
MedianPrice	중간가격((고가+저가)/2)
MRO(조건,Period,nth)	Period 기간 동안 발생한 조건들 중에 현재봉으로부터 nth번째 가까운 조건이 몇 봉 전에 발생했는지를 반환
Range	캔들 길이(고가-저가)
STD	표준편차
SwingHigh(Nth,기준값,좌측,우측,기간)	Period 기간 동안 발생한 swinghigh들 중 현재로부터 Nth번째로 가까운 Swinghigh 값
SwingHighBar(Nth,기준값,좌측,우측, 기간)	Period 기간 동안 발생한 swinghigh들 중 현재로부터 Nth번째로 가까운 Swinghigh 값이 몇 봉 전에 발생했는지를 반환
SwingLow(Nth,기준값,좌측,우측,기간)	Period 기간 동안 발생한 SwingLow 들 중 현재로부터 Nth번째로 가까운 SwingLow 값
SwingLowBar(Nth,기준값,좌측,우측,기간)	Period 기간 동안 발생한 SwingLow 들 중 현재로부터 Nth번째로 가까운 SwingLow 값이 몇 봉 전에 발생했는지를 반환
TrueHigh	진(眞)고가
TrueLow	진(眞)저가
TrueRange	잔(眞)가격범위
WC	가중증가
지표함수 종류	

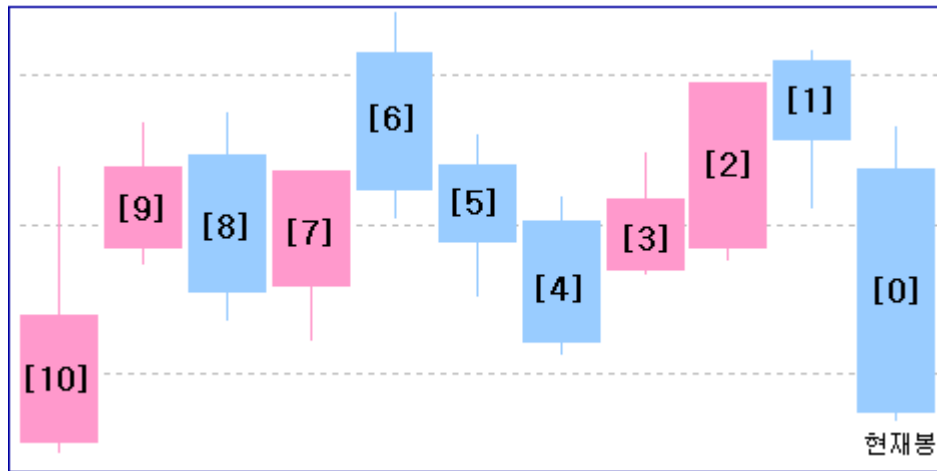
AccDist	Accumulation_Distribution
ADX	Average Directional Movement Index
ATR	Average True Range
Average	단순이동평균
AverageFC	단순이동평균
BollBandDown	볼린저밴드 하단
BollBandUp	볼린저밴드 상단
BW	Binary Wave
CCI	CCI(Commodity Channel Index)
CMO	CMO(Chandre Momentum Oscillator)
CO	CO(Chaikin's Oscillator)
CSar	파라볼릭(증가)
Sar	파라볼릭
CV	CV(Chaikin's Volatility)
Detrend	Detrend
Disparity	이격도(Disparity)
DiMinus	DI-(Directional Moving Average Index Minus)
Diplus	DI+(Directional Moving Average Index Plus)
DMI	Directional Movement Index
EnvelopeDown	Envelope Band 하한
EnvelopeUp	Envelope Band 상한
EOM	Ease Of Momentum
KeltnerChannel	Keltner Channel
LRL	Linear Regression Line(선형회귀선)
LRS	Linear Regression Slope(선형회귀곡선)
MACD	MACD(Moving Average Convergence & Divergence)
MACD_OSC	Moving Average Convergence & Divergence Oscillator
MassIndex	Mass Index
MFI	Money Flow Index
Momentum	모멘텀, N일전 증가대비 현재가 비율
NVI	Negative Volume Index
OBV	On Balance Volume
OSCP	Price Oscillator
OSCV	Volume Oscillator

PercentR	%R
PROC	Price Moving Average Oscillator
PVI	Positive Volume Index
PVT	Volume Trend
RSI	Relative Strength Index
Simrido	심리도
FastK	Stochastics Fast K(기본적 K)
FastD	Stochastics Fast D
SlowK	Stochastics Slow K (= Fast D)
StochasticsK	Stochastics %K (= Slow K, = Fast D_
StochasticsD	Stochastics Slow D
SMI	Stochastics Momentum Index
SONAR	Sonar Momentum
TRIX	TRIX 지표함수
TSI	True Strength Index
TXAverage	3중 지수이동평균
UltimateOsc	Ultimate Oscillator
VHF	Vertical Horizontal Filter
VR	Volume Ratio
VROC	Volume Ratio of Change
WILLA	Williams' Accumulation/Distribution
WILLR	Williams' %R
WMA	가중이동평균
봉패턴 종류	
BullishEngulfing	상승 장악형 캔들 패턴일 경우 True 아니면 False
BearishEngulfing	하락 장악형 캔들 패턴일 경우 True값 아니면 False
C_Doji	십자 캔들 패턴일 경우 1값 아니면 0
DarkCloud	먹구름형 캔들 패턴일 경우. True 아니면 False
EveningStar	석별형 캔들 패턴일 경우 True 아니면 False
Hammer	망치형 캔들 패턴일 경우 True 아니면 False
HangingMan	교수형 캔들 패턴일 경우 True 아니면 False
MorningStar	셋별형 캔들 패턴일 경우 True 아니면 False

9) 이전 값 참조

데이터나 함수의 이전 값 참조는 봉 기준이다. 이전 값을 참조하고자 할 때 간단히 데이터나 함수, 변수의

뒤에 대괄호[]를 붙이고 현재봉(기준봉)으로부터 몇 번째 과거 봉인지를 세어서 대괄호[] 안에 넣어주면 됩니다. 현재봉은 [0]이며 생략이 가능합니다.



→ [0]은 생략이 가능하므로 C 와 C[0]은 같은 값
 → 볼린저밴드 상단의 2봉 전 값 : bollbandup(20,2)[2]
 → value1의 3봉 전 값 : value1[3]
 → 주가 상승반전 : $C > C[1]$ and $C[1] < C[2]$
 → 주가 하락반전 : $C < C[1]$ and $C[1] > C[2]$
 → if $C[1] \leq ma(c,5)[1]$ and $C > ma(C,20)$ then buy();
 전봉에는 주가가 5일 이동평균보다 작았고 현재봉에서는 주가가 5일 이동평균 보다 크면 매수

※ 주의

DayOpen, Dayhigh, Daylow, dayclose

위 함수는 분봉에서 일간의 시가,고가,저가,종가를 가져오는 함수이다.

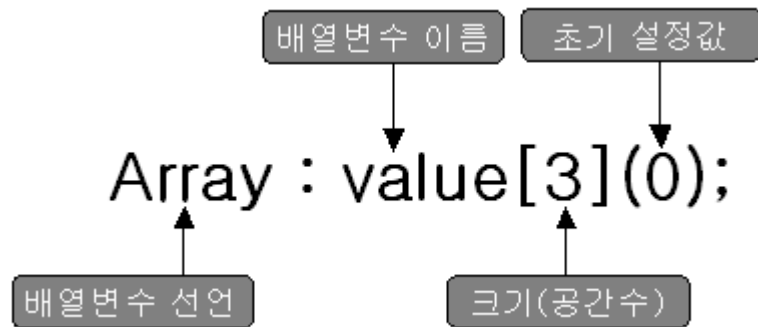
전일 값을 가져오기 위해서는 DayOpen[1], Dayhigh[1], Daylow[1], dayclose[1]와 같이 작성하면 DayOpen(0)[1], Dayhigh(0)[1], Daylow(0)[1], dayclose(0)[1]와 같은 내용으로 당일 시가,고가,저가,종가를 전봉 기준으로 불러와 사용한다는 의미이다.

DayOpen(1), Dayhigh(1), Daylow(1), dayclose(1)로 작성해야 전일 값을 가져와 사용할 수 있다.

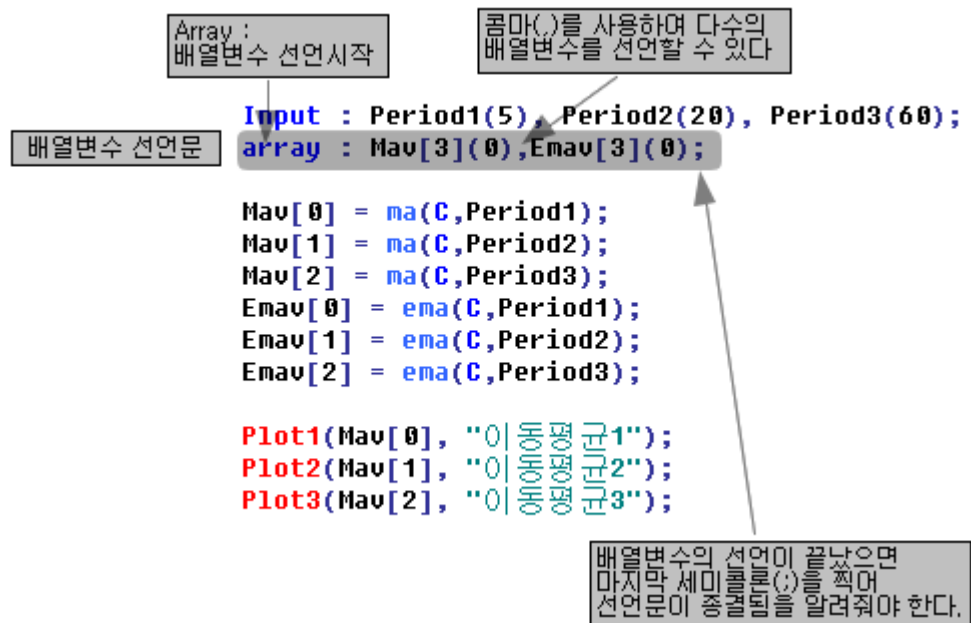
4 에스랭귀지 활용

1) 배열

배열이란 동일한 성격의 데이터를 관리하기 쉽도록 하나로 묶어 사용하는 형태를 말합니다.

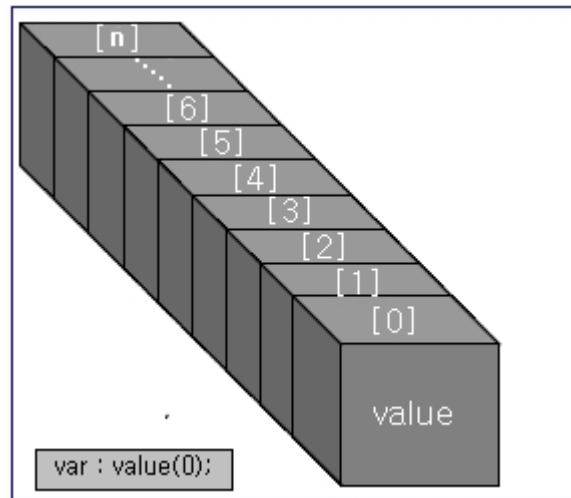


배열변수는 일반변수를 선언하는 것과 유사합니다. 다만 선언을 Array나 Arrays로 선언해야 하며 초기값을 설정하기 전에 공간을 몇 개로 분리할 것인지 지정해 주어야 합니다. 분리된 공간은 번호가 부여가 되고 이 번호는 0부터 시작하므로 크기를 3으로 지정하면 0,1,2로 총 3개의 공간이 생기게 됩니다.

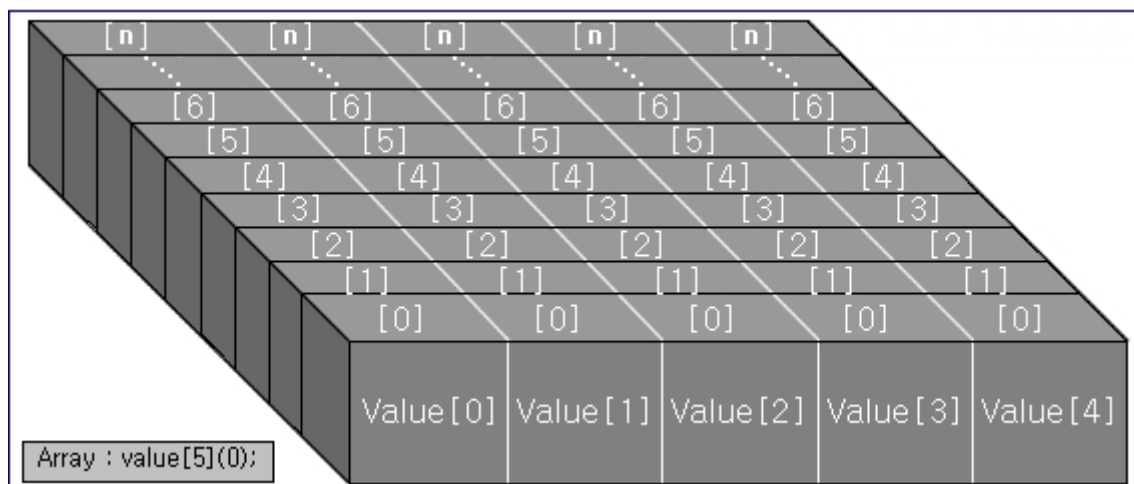


일반 변수는 하나의 공간에 하나의 값을 받아들이고 이런 공간이 각 봉마다 있는 구조라면 배열변수는 공간을 분할하여 여러 개의 공간을 만들고 이런 구조의 공간이 각 봉마다 존재하는 변수입니다.

간단히 그림으로 살펴보면 아래와 같습니다.



일반변수는 선언하게 되면 하나의 공간이 생기고 그 공간에 한 개의 값을 저장하게 되고 이 저장된 값을 현재봉(기준봉)으로부터 [0],[1]..., [n]으로 이전 값을 참조하게 됩니다.



배열변수는 선언하게 되면 하나의 공간을 지정된 개수만큼 분할하게 되고 간단하게 X개의 방이 있는 집을 만든다고 보면 되며 각 방은 각각 번호가 부여되고 각 방별로 이전 값 참조가 가능합니다.

배열변수는 순차적인 여러 변수가 필요할 때 사용하게 되는데 변수가 100개가 필요한 상황이라면 일반변수라면 아래와 같이 작성이 되므로 선언하는 것 자체도 상당히 시간이 드는 작업이 됩니다.

`var : A0(0), A1(0), A2(0), A3(0), A4(0), A5(0), A6(0), A7(0), A8(0), A9(0), ..., A99(0);`

이때 배열변수를 사용하면 아래와 같이 간단히 작성을 해 A[0]~A[99]까지 총 100개의 공간(방)이 있는 변수를 만들어 사용할 수 있습니다.

`Array : A[100](0);`

간단한 예로 1분봉 차트에 적용하여 60분봉의 고가를 구현하는 지표식입니다.

```

var : Frame(0),Hval(0);

Frame = TimeToMinutes(stime)%60;

if dayindex == 0 or Frame < Frame[1] Then
    Hval = H;

if H > Hval Then
    Hval = H;

plot1(Hval);

```

현재 12시 30분이라면 Hval은 12시에서 12시 30분까지의 최고가를 리턴하게 됩니다. 하지만 위의 식은 과거 60분 동안의 최고가 값을 가져올 수 없습니다. 과거 60분 동안의 값을 총 10개가 필요하다면 필요한 수만큼 변수를 추가로 선언하고 해당 변수에 할당을 해야 하므로 상당히 번거로운 식이 되고 수식 또한 길어지게 됩니다.

```

var : Frame(0),Hval(0),Hval1(0),Hval2(0),Hval3(0),Hval4(0),
    Hval5(0),Hval6(0),Hval7(0),Hval8(0),Hval9(0);

Frame = TimeToMinutes(stime)%60;

if dayindex == 0 or Frame < Frame[1] Then{
    Hval = H;
    Hval1 = Hval[1];
    Hval2 = Hval1[1];
    Hval3 = Hval2[1];
    Hval4 = Hval3[1];
    Hval5 = Hval4[1];
    Hval6 = Hval5[1];
    Hval7 = Hval6[1];
    Hval8 = Hval7[1];
    Hval9 = Hval8[1];
}

if H > Hval Then
    Hval = H;

plot1(Hval);

```

하지만 아래와 같이 배열변수를 사용하게 되면 수식도 짧아지면서 더 많은 변수가 필요하다면 배열변수의 크기만을 조절하면 되므로 수식의 추가 없이 더 많은 값을 저장하여 사용할 수 있습니다.

```

var : Frame(0),count(0);
Array : Hval[10](0);

Frame = TimeToMinutes(stime)%60;

if dayindex == 0 or Frame < Frame[1] Then{
    Hval[0] = H;
    for count = 1 to 9{
        Hval[count] = Hval[count-1][1];
    }
}

if H > Hval[0] Then
    Hval[0] = H;

plot1(Hval[0]);

```

※ TimeToMinutes(stime)%분단위

만약 1분봉에서 더 긴 주기인 60분봉의 값을 구현하기 위해서는 우선 1분봉에서 60분 단위로 봉을 구분하여야 한다.

TimeToMinutes 함수는 시간을 밤 12시 이후 경과된 분으로 변경해 주는 값이다.

9시면 540이고 10시면 600이다. 이 값을 60으로 나누어 나머지를 구하면 정각에는 0값을 할당하고 이후 초과되는 분으로 값을 표시한다.

0 일 경우, 값이 전봉보다 작아지는 봉이 분 단위의 경계이다.

예를 들면 1분봉에서는 9시(0)→ 9시1분(1)→ 9시 2분(2)→ → 9시 59분(59)
→ 10시(0)

※ For count = 1 to 9

Hval[count] = Hval[count-1][1];

Hval[0]에는 항상 최근 60분의 최고가를 저장해야 한다. 그러므로 새로운 60분이 시작되면 기존의 Hval[0]의 값은 전 60분의 최고값이므로 Hval[1]로 이동시키고 기존 Hval[1]의 값은 Hval[2]로 옮기는 등 순차적으로 옮겨줘야 현재시점에서 전60분 전전 60분으로 값을 가져와 사용할 수 있다.

수식의 for문은 새로운 60분이 시작되면 최고값을 재계산해 넣어야 하므로 기존의 값을 이동시키는 작업이다.

기존의 [0]방의 값은 [1]번방으로 옮기고 [1]번은 [2]번으로 순차적으로 옮기게 된다.

Count에 1이 할당되면 Hval[1]에 Hval[1-1][1]값을 저장한다.

즉 Hval[1] = Hval[0][1];이 된다.



2) 디버깅

디버깅(debugging)이란 원래 프로그램의 오류를 찾아내어 수정하는 것을 말합니다. 랭귀지 편집기에서 디버깅이란 수식의 계산값들을 출력하여 오류를 바로 잡는 기능이라고 보면 됩니다.

수식을 작성하고 적용하게 되면 의도와는 다르게 결과가 발생하는 경우가 많이 발생합니다.

지표식이나 강조식, 검색식과 같은 경우에는 사용되는 변수나 함수, 계산식 등을 하나씩 지표로 출력하면서 잘못된 점을 찾을 수도 있지만 시스템식의 경우에는 지표출력함수가 같이 사용될 수 없고 반대로 지표식에는 포지션함수 등이 사용될 수 없으므로 지표식으로 수식을 검증하기에는 한계가 있습니다.

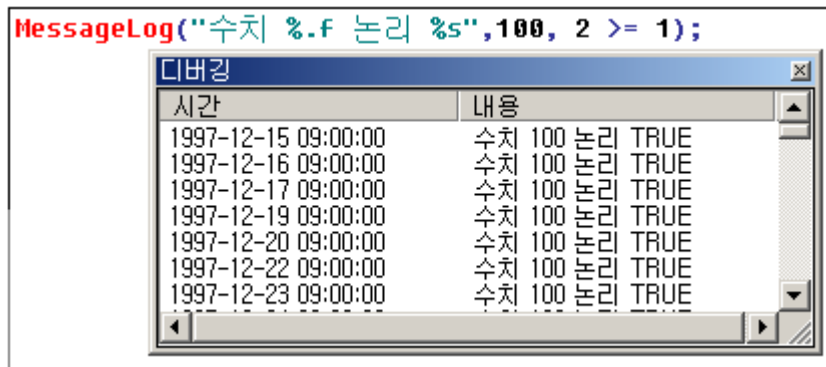
이렇게 수식이 의도와는 다른 결과를 보일 때 사용될 수 있는 것이 디버깅이고 디버깅을 하기 위해서는 MessageLog함수를 이용해야 해야 합니다.

함수	MessageLog("Expression", 출력대상)
매개변수	<p>※ Expression</p> <p>출력대상의 이름과 표현방식</p> <p>출력대상이 수치라면 표현방식은 (이름) %.f</p> <p>출력대상이 논리값이나 문자열이라면 표현방식은 (이름) %s</p> <p>이름은 생략가능.</p> <p>※ 출력대상 : 수치, 논리값, 문자열</p>

출력대상의 이름과 표현방식은 큰따옴표(double Quotation mark)안에 설정을 해 줍니다. 이름은 생략해도 되지만 표현방식은 생략하면 안 됩니다.

%.f는 출력대상이 숫자(figure)라는 의미이다. 수치를 출력할 때는 수치의 소수점 자릿수도 설정할 수 있다. f앞에 소수점 자릿수를 써주면 됩니다. %.2f라고 설정하면 소수점 2자리까지 출력되게 되고 %s는 출력대상이 문자열(String)이라는 의미입니다. 논리식이나 논리형 변수를 출력할 때는 %s로 표현방법을 설정해야 하며 표현방식의 f나 s는 소문자로만 작성해야 합니다.

출력대상은 수치값이나 논리값, 문자열 모두 가능하며 한번에 여러 개를 출력한다면 출력대상의 이름과 표현방식도 동일한 수 만큼 설정을 해 줘야 합니다. 출력대상은 10개인데 표현방식은 6개라면 출력대상으로 지정한 것 중 좌측에서 6개만 출력되게 됩니다. 또한 표현방식도 유의해서 지정해야 합니다. 출력대상은 논리식인데 %.f로 설정하면 디버깅 창에 0으로만 값이 찍히게 되며 출력대상은 수치인데 %s로 설정하면 (null)이라 출력됩니다.



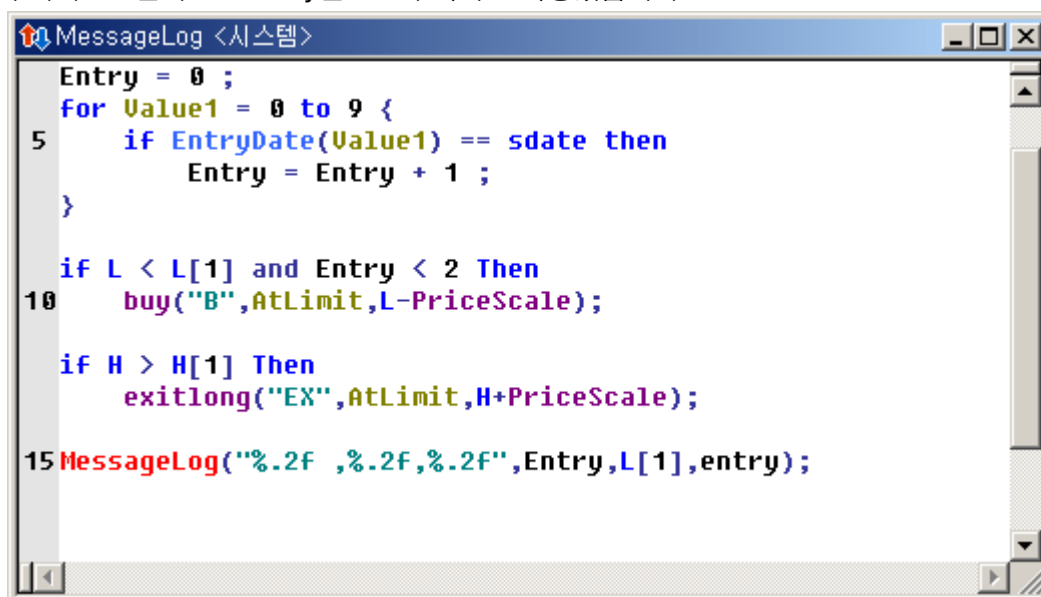
간단한 식을 작성하여 수식에 사용된 값을 MessageLog로 찍는 연습을 해보겠습니다.

기본전략

매수 : 저가가 연속적으로 하락하면 매수

청산 : 고가가 연속적으로 상승하면 청산

위와 같은 전략에 추가로 당일 1번까지만 진입이 들어가게 하기 위해 진입 횟수를 카운트하는 계산식을 작성하여 추가하고 1번이므로 entry는 2보다 작다고 작성했습니다.



당일 진입횟수는 for문을 이용해서 최근의 10개의 진입날짜와 오늘 날짜를 비교하여 같은 날짜가 몇 개나

되는지 세어 entry에 할당하도록 작성했습니다.



시스템을 30분봉 차트에 적용하면 당일 한번의 진입을 의도 했는데 2번까지 진입이 발생하는 것을 볼 수 있다. 어떤 이유로 한번이 아닌 2번이 발생했는지 디버깅 창에서 메시지로 로그 출력한 entry값을 확인해 보도록 합니다.

시간	내용
2009-02-03 09:00:00	진입횟수 0,00
2009-02-03 09:30:00	진입횟수 1,00
2009-02-03 10:00:00	진입횟수 1,00
2009-02-03 10:30:00	진입횟수 1,00
2009-02-03 11:00:00	진입횟수 1,00
2009-02-03 11:30:00	진입횟수 1,00
2009-02-03 12:00:00	진입횟수 1,00
2009-02-03 12:30:00	진입횟수 1,00
2009-02-03 13:00:00	진입횟수 1,00
2009-02-03 13:30:00	진입횟수 2,00
2009-02-03 14:00:00	진입횟수 2,00
2009-02-03 14:30:00	진입횟수 2,00
2009-02-03 15:00:00	진입횟수 2,00
2009-02-04 09:00:00	진입횟수 0,00

디버깅 화면을 보면 9시 30분에 당일 첫번째 진입이 들어가고 entry가 1값이 할당되어 유지되다가 13시 30분에 두번째 진입이 발생하자 2값으로 변경되는 것을 볼 수 있다. Entry에 값의 할당되는 것만 보면 계산식의 문제는 아닌 것을 알 수 있습니다.

다시 디버깅 창에 entry값이 할당되는 것을 보면 당일 첫 봉에 진입이 없으므로 0값이 할당된 것을 볼 수 있습니다. 즉 당일 첫 진입은 entry가 0일 때 발생하고 두번째 진입은 entry가 1값일 때 발생하게 됩니다. 그러므로 진입 조건문에 문제가 있는 것을 알 수 있습니다. 당일 첫 진입은 entry가 0일 때 발생하는 진입이므로 진입조건문의 entry < 2를 entry < 1로 변경해 주면 당일 한번만 진입하는 수식이 됩니다.

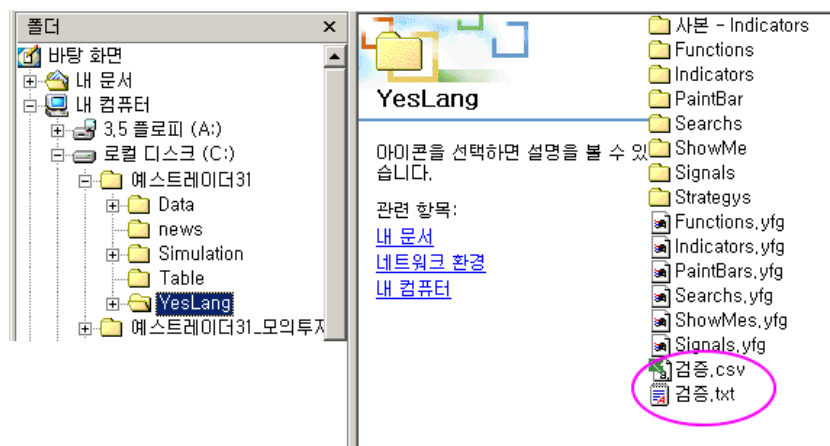
```
if L < L[1] and Entry < 1 Then
    buy("B",AtLimit,L-PriceScale);
```

3) 데이터 추출

간혹 수식의 계산된 결과값이나 차트의 데이터 등을 추출하여 텍스트문서나 엑셀 등 외부 프로그램에서 보고자 할 때가 있습니다. 예스랭귀지에서는 텍스트문서 형식이나 CSV(Comma Separate Value)형식으로 외부로 추출하여 내보낼 수 있습니다. 데이터를 추출할 때는 Print 함수를 사용하면 되며 시뮬레이션 차트에서는 사용할 수 없습니다. Print함수는 MessageLog함수와 사용법이 비슷 하지만 매개변수로 파일명과 경로를 지정해 주어야 합니다.

함수	Print("파일명","Expression", 출력대상)
매개변수	<p>※ 파일명 : 저장 경로 와 파일명을 지정</p> <p>※ Expression :</p> <p>출력대상의 이름과 표현방식</p> <p>출력대상이 수치라면 표현방식은 (이름) %.f</p> <p>출력대상이 논리값이나 문자열이라면 표현방식은 (이름) %s</p> <p>이름은 생략가능.</p> <p>※ 출력대상 : 수치, 논리값, 문자열</p>

외부로 파일을 만들어 내보내야 하므로 이름과 확장자를 큰따옴표(double Quotation mark) 안에 지정해 줘야 합니다. 텍스트 문서로 추출하려면 "이름.txt"로 지정하면 되고 csv 형식이면 "이름.csv"로 지정해 주면 됩니다.



기본적으로 추출된 파일은 예스트레이더 설치폴더의 YesLang 폴더 아래에 추출됩니다. 다른 경로로 추출하고 싶으면 경로를 같이 지정해 주면 됩니다.

```
print("test.txt","종가 %.2F",Close);
```

기본 추출경로에 test.txt 파일에 close를 종가라는 이름으로 소수점 2자리까지 출력

```
print("C:\test.txt","종가 %.2F",Close);
```

로컬디스크(C)에 test.txt파일에 Close를 증가라는 이름으로 소수점 2자리까지 출력

출력대상의 이름과 표현방식은 큰따옴표(double Quotation mark)안에 설정을 해 줍니다. 이름은 생략해도 되지만 표현방식은 생략하면 안 됩니다.

%f는 출력대상이 숫자(figure)라는 의미이고 수치를 출력할 때는 수치의 소수점 자릿수도 설정할 수 있습니다. f앞에 소수점 자릿수를 써주면 되며 %.2f라고 설정하면 소수점 2자리까지 출력되게 됩니다. %s는 출력대상이 문자열(String)이라는 의미입니다. 논리식이나 논리형 변수를 출력할 때는 %s로 표현방법을 설정해야 하며 표현방식의 f나 s는 소문자로만 작성해야 합니다.

출력대상은 수치값이나 논리값, 문자열 모두 가능하며 한번에 여러 개를 출력한다면 출력대상의 이름과 표현방식도 동일한 수 만큼 설정을 해 줘야 합니다. 출력대상은 10개인데 표현방식은 6개라면 출력대상으로 지정한 것 중 좌측에서 6개만 출력되게 됩니다. 또한 표현방식도 유의해서 지정해야 합니다. 출력대상은 논리식인데 %f로 설정하면 디버깅 창에 0으로만 값이 찍히게 되며 출력대상은 수치인데 %s로 설정하면 (null)이라 출력됩니다.

텍스트 문서로 추출하여 엑셀에서 보기 위해서는 엑셀에서 텍스트 문서를 불러와 텍스트 마법사를 이용해야 하는 번거로운 작업이 있으므로 csv형식으로 추출하고 해당 파일을 클릭만하면 바로 엑셀에서 볼 수 있습니다.

CSV(Comma Separate Value) 형식은 데이터베이스나 표 계산 소프트웨어 데이터를 보존하는 형식의 하나. 각 항목이나 판매 내용마다 콤마(,)로 구분해서 적는다. 이 형식의 파일은 텍스트 파일로 보존하기 때문에, 워드 프로세서나 편집기에서 열람, 편집할 수 있다. 수많은 애플리케이션에서 취급하는 범용 형식이기 때문에 개인 휴대 정보 단말기(PDA)와 개인용 컴퓨터(PC) 간에 주소록이나 표 데이터를 주고받을 때에도 데이터 파일을 이 형식으로 변환하여 송수신하는 경우가 많다.

중요한 점은 csv파일로 추출하기 위해서는 Print함수 작성시 한가지 작업을 더해 줘야 한다는 것입니다. 출력대상이 여러 개 일 때 각 개별 값을 열(column)로 구분되어 출력되기를 원한다면 콤마(,) 이용해 일련된 값을 구분해 줘야 한다는 것입니다.

예를 들어 아래와 같이 작성하여 파일을 추출하여 엑셀에 불러보면

```
print("test.csv","시 %.2f 종 %.2f",Open,Close);
```

	A	B	C
1	2007-11-26 11:30:00 시 232.95 종 232.95		
2	2007-11-26 12:00:00 시 232.95 종 233.95		
3	2007-11-26 12:30:00 시 233.90 종 234.25		
4	2007-11-26 13:00:00 시 234.25 종 235.80		

값들이 열을 나눠 표시되는 것이 아니라 모두 A열에 표시가 되어 편집이 용이하지 않습니다.

```
print("C:\\test.txt","시,%.2f,종,%.2f",Open,Close);
```

	A	B	C	D	E
1	2007-11-26 11:30	시	232.95	종	232.95
2	2007-11-26 12:00	시	232.95	종	233.95
3	2007-11-26 12:30	시	233.9	종	234.25
4	2007-11-26 13:00	시	234.25	종	235.8

이름과 표현방식 앞에 콤마(,)를 찍어 값을 구분한 후 출력하면 열(column)로 구분되어 추출되게 됩니다.

4) 타종목 참조(Multi Data)

다수의 데이터를 이용하여 수식을 작성 할 수도 있습니다. 다양한 데이터를 차트에 적용하여 한 개의 주종목(Data1)을 기준으로 최대 98개의 데이터(Data2~Data99)를 참고로 하는 수식을 작성할 수 있습니다.

타종목 참조는 주종목(기본종목)과 다른 종목을 참조하는 것을 말하며 타주기 참조는 주종목(기본종목)과 다른 주기의 데이터를 참조하는 것을 말합니다. 당연히 주종목(기본종목)과 종목도 다르고 주기도 다르게 참조할 수도 있고 주종목(기본종목)과 종목을 동일하지만 주기만 다르게 하여 참조도 가능합니다.

예를 들어 옵션을 거래하는데 선물지수를 참조한다거나 5분봉 선물에서 30분봉이나 일봉의 선물지수를 참조하는 것을 말합니다. 다만 주종목(Data1)에 대해서만 포지션을 취할 수 있다.

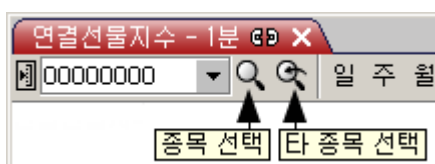
타종목을 참조하는 수식을 만든다면 두가지 정도를 염두에 두어야 합니다.

우선 차트를 타종목을 참조할 수 있게 차트에 데이터를 적용하여 구성해야 하고 수식도 참조데이터를 이용할 수 있게 작성해야 한다는 것입니다.

선물지수에서 5-20 이동평균이 골든크로스가 나면 옵션 종목을 매수하고 5-20 데드크로스가 발생하면 옵션 종목을 청산하는 간단한 전략을 시스템식으로 작성하면서 자세히 알아 보겠습니다. 주기는 두 종목 모두 동일한 주기인 1분봉으로 하겠습니다.

가) 차트 구성

차트에는 종목을 선택할 수 있는 종목선택창이 두개가 있습니다. 하나는 주종목(기본종목)을 선택할 수 있는 종목 선택창이고 다른 하나는 참조종목을 선택할 수 있는 종목 선택창입니다.



옵션을 거래하는데 선물지수를 참조하여야 하므로 옵션은 종목선택버튼을 클릭하면 나오는 종목선택창에서 선택하여 적용하고 선물지수는 타종목선택 버튼을 클릭하면 나오는 종목선택창에서 선택하여 적용해야 합니다.



참조종목으로 적용한 연결선물지수에 data2라는 표시되어 있는 것을 확인할 수 있습니다. 이것의 의미는 참조종목으로 적용한 연결선물지수는 타종목 참조 함수 중 data2로 그 값을 가져와 사용할 수 있다는 의미입니다.

참조종목은 선택하여 적용하는 순서에 따라 data2~data99까지 순차적으로 데이터 참조 함수 명칭이 붙게 됩니다. 주종목은 data1이고 차트에는 주종목 포함 99개의 데이터를 적용할 수 있고 필요한 데이터를 모두 적용하면 차트의 구성은 완료됩니다.

나) 타종목 참조 수식 작성

close는 종가를 가져오는 데이터 예약어입니다. 수식에서 close라고 작성하면 주종목(기본종목)의 종가입니다. 참조종목의 종가는 data2(Close)로 작성해야만 차트에 적용된 data2의 close를 가져오게 됩니다. 마찬가지로 ma(C,5)는 주종목의 이동평균선이고 참조종목의 이동평균선은 data2(ma(c,5))입니다. 이렇게 타종목 참조 함수 안에 데이터 예약어나, 함수, 계산식 등을 넣으면 해당 참조종목의 값으로 계산하게 됩니다.

우선 주종목을 기준으로 하는 5-20 이동평균 골든크로스 매수, 데드크로스 청산하는 시스템식을 작성해보면 Crossup, Crossdown, Ma 세가지 함수를 이용하여 간단히 작성됩니다.

```

input : SP(5),LP(20);

if crossup(ma(c,SP),ma(C,LP)) Then
    buy("b");

if CrossDown(ma(c,SP),ma(C,LP)) Then
    ExitLong("x");

```

이 식은 모두 주종목의 데이터를 분석하여 신호를 발생하므로 선물 1분봉 차트에 적용하면 아래와 같이 신호가 발생합니다.



이젠 기준식이 만들어 졌으므로 선물을 참조하여 옵션을 매매하는 식으로 변경해야 합니다.

```

input : SP(5),LP(20);

if crossup(data2(ma(c,SP)),data2(ma(C,LP))) Then
    buy("b");

if CrossDown(data2(ma(c,SP)),data2(ma(C,LP))) Then
    ExitLong("x");

```

차트에 선물지수 데이터를 참조 종목으로 적용하여 data2라는 타종목 참조함수를 사용하라고 지시를 받았으므로 타종목 참조 함수 data2를 이용하여 기본수식에서 사용된 데이터 예약어나 함수, 계산식 등을 처리해주면 됩니다.

그리고 수식을 적용하여 동일한 위치에서 신호가 발생하는지 확인하면 타종목 참조 수식이 완성됩니다.



기본적으로는 타종목 참조는 주거나 종목은 차트에서 설정하므로 수식에서는 해당 타종목을 주종목으로 놓고 작성한 수식에 타종목 참조함수를 추가하면 됩니다.

다) 타종목/타주기 유의사항

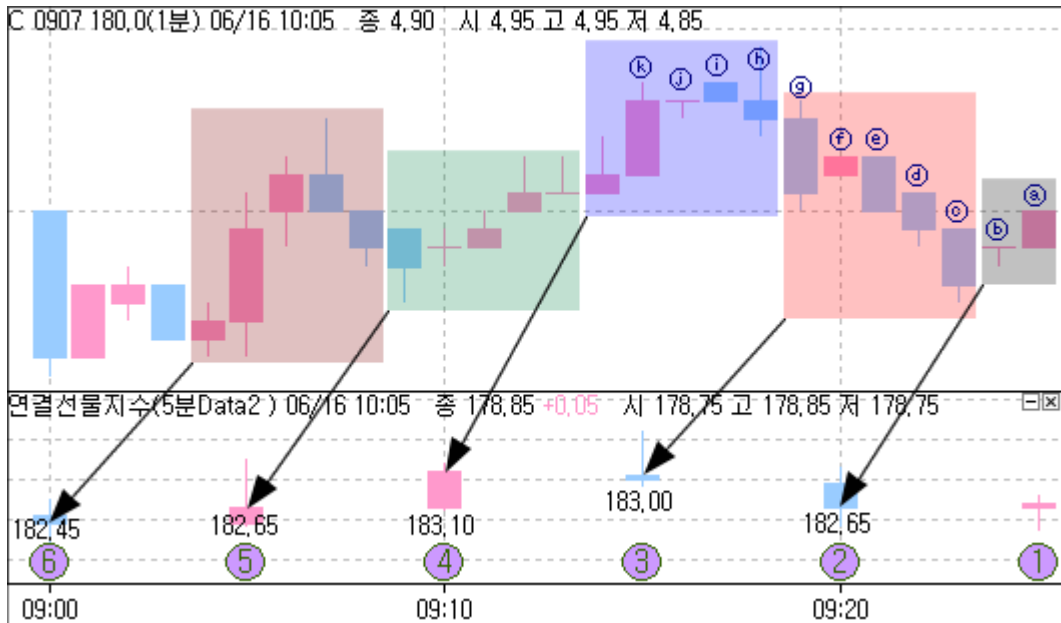
① 랭귀지는 완성봉의 값만 사용가능하다.

타주기 참조종목의 data2(c)는 어떤 봉의 값인가?

수식은 완성된 봉의 값만 사용할 수 있습니다. 그러므로 타주기를 이용할 때 주종목 봉이 어떤 봉의 값을 참조하는 지를 알아야 합니다.

주종목 기준으로 가장 최근에 완성된 봉이 어느것인지 구별해 보겠습니다.

주종목은 1분봉이고 참조종목은 5분봉입니다.



시간	내용
2009-06-12 09:02:00	data2(Stime) 144500 data2(c) 182.75
2009-06-12 09:03:00	data2(Stime) 144500 data2(c) 182.75
2009-06-12 09:04:00	data2(Stime) 90000 data2(c) 182.45
2009-06-12 09:05:00	data2(Stime) 90000 data2(c) 182.45
2009-06-12 09:06:00	data2(Stime) 90000 data2(c) 182.45
2009-06-12 09:07:00	data2(Stime) 90000 data2(c) 182.45
2009-06-12 09:08:00	data2(Stime) 90000 data2(c) 182.45
2009-06-12 09:09:00	data2(Stime) 90500 data2(c) 182.65
2009-06-12 09:10:00	data2(Stime) 90500 data2(c) 182.65
2009-06-12 09:11:00	data2(Stime) 90500 data2(c) 182.65
2009-06-12 09:12:00	data2(Stime) 90500 data2(c) 182.65
2009-06-12 09:13:00	data2(Stime) 90500 data2(c) 182.65
2009-06-12 09:14:00	data2(Stime) 91000 data2(c) 183.10
2009-06-12 09:15:00	data2(Stime) 91000 data2(c) 183.10
2009-06-12 09:16:00	data2(Stime) 91000 data2(c) 183.10
2009-06-12 09:17:00	data2(Stime) 91000 data2(c) 183.10
2009-06-12 09:18:00	data2(Stime) 91000 data2(c) 183.10
2009-06-12 09:19:00	data2(Stime) 91500 data2(c) 183.00
2009-06-12 09:20:00	data2(Stime) 91500 data2(c) 183.00
2009-06-12 09:21:00	data2(Stime) 91500 data2(c) 183.00
2009-06-12 09:22:00	data2(Stime) 91500 data2(c) 183.00
2009-06-12 09:23:00	data2(Stime) 91500 data2(c) 183.00
2009-06-12 09:24:00	data2(Stime) 92000 data2(c) 182.65
2009-06-12 09:25:00	data2(Stime) 92000 data2(c) 182.65

MessageLog("data2(Stime) %.f data2(c) %.2f ",data2(Stime),data2(c));

㉔봉은 마치 ㉑번 봉을 참조할 것 같지만 ㉑번 봉은 현재 완성된 봉이 아니므로 ㉔봉에서는 참조를 할 수 없습니다. ㉔봉에서 가장 최근 완성된 봉은 ㉒번 봉이므로 과거데이터 중 ㉒번 봉 까지만 봉의 값을 가져와 사용할 수 있습니다. 디버깅 화면을 보면 ㉔봉인 9시 25분 봉에 참조하는 봉은 9시 20분 봉이고 가격은 186.65임을 확인 할 수 있습니다.

9시 24분봉(㉔봉)은 9시 20분 봉과 동일한 시간에 완성이 되므로 ㉔봉에서는 ㉒번 봉인 값을 사용할 수 있습니다. 이와 마찬가지로 ㉑~㉔ 봉은 ㉓번 봉이 가장 최근 완성된 봉이고 ㉓번 봉 까지만 데이터를 가져와

사용할 수 있고 ④봉은 3번 봉과 같이 시간에 완성되므로 ③번 봉이 가장 최근 완성된 봉입니다.

또 다른 유의 사항은 타종목의 값을 할당받는 변수의 선언문제입니다.

일반적으로 변수는 수치를 할당 받으면 (0)으로 선언하고 논리결과를 할당 받으면 (false), 문자를 할당 받으면 (" ")로 선언합니다. 하지만 타종목의 값을 할당 받는 변수는 선언문 작성 시 추가해야 할 내용이 있다.

아래와 같이 수식을 작성하여 차트에 적용해 보면

```
var : DC1(0),DC2(0,data2);  
  
DC1 = data2(C);  
DC2 = data2(C);  
  
plot1(DC1[1]);  
plot2(DC2[1]);  
  
MessageLog("DC1[1] %.2f DC2[1] %.2f ",DC1[1],DC2[1]);
```

DC1 변수와 DC2 변수에 동일하게 data2(c)를 할당하였고 DC1은 선언문에 (0)으로 선언을 하고 DC2는 (0,data2)로 선언을 하였습니다. 지표로 각 변수의 이전 봉 값을 지표로 출력하도록 식을 작성하면 아래와 같이 적용된 지표를 보면 똑 같이 그려져야 할 것 같은 2개값이 다른 값으로 그려지는 것을 볼 수 있습니다.

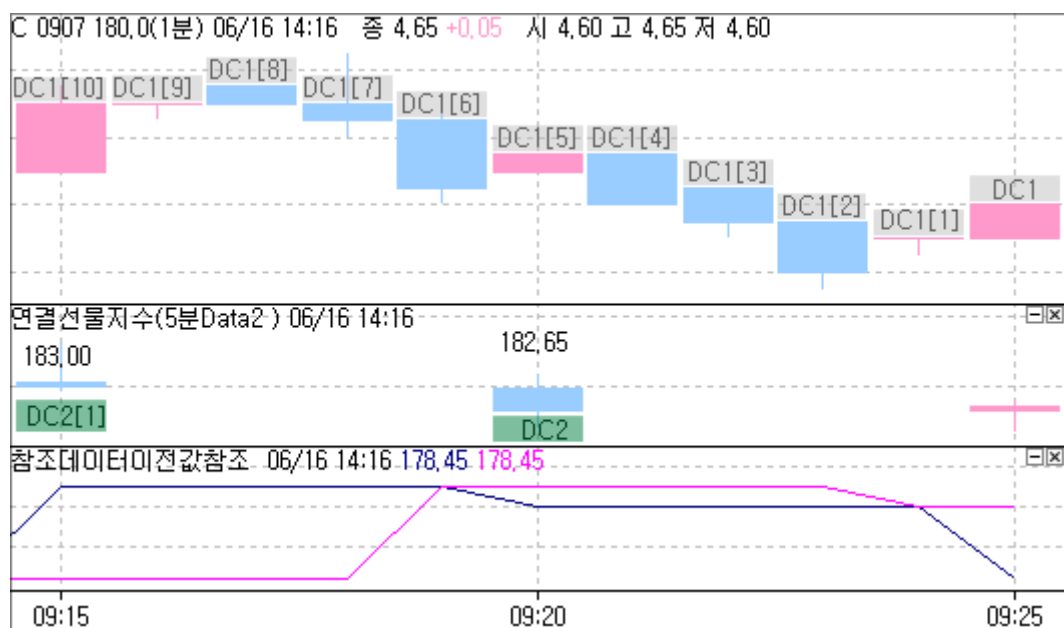


다시 메시지로그를 통해 값의 할당이 어떻게 되고 있는지 디버깅 창에서 확인해 보면

시간	내용
2009-06-12 09:00:00	DC1[1] 182,75 DC2[1] 182,80
2009-06-12 09:01:00	DC1[1] 182,75 DC2[1] 182,80
2009-06-12 09:02:00	DC1[1] 182,75 DC2[1] 182,80
2009-06-12 09:03:00	DC1[1] 182,75 DC2[1] 182,80
2009-06-12 09:04:00	DC1[1] 182,75 DC2[1] 182,75
2009-06-12 09:05:00	DC1[1] 182,45 DC2[1] 182,75
2009-06-12 09:06:00	DC1[1] 182,45 DC2[1] 182,75
2009-06-12 09:07:00	DC1[1] 182,45 DC2[1] 182,75
2009-06-12 09:08:00	DC1[1] 182,45 DC2[1] 182,75
2009-06-12 09:09:00	DC1[1] 182,45 DC2[1] 182,45
2009-06-12 09:10:00	DC1[1] 182,65 DC2[1] 182,45
2009-06-12 09:11:00	DC1[1] 182,65 DC2[1] 182,45
2009-06-12 09:12:00	DC1[1] 182,65 DC2[1] 182,45
2009-06-12 09:13:00	DC1[1] 182,65 DC2[1] 182,45
2009-06-12 09:14:00	DC1[1] 182,65 DC2[1] 182,65
2009-06-12 09:15:00	DC1[1] 183,10 DC2[1] 182,65
2009-06-12 09:16:00	DC1[1] 183,10 DC2[1] 182,65
2009-06-12 09:17:00	DC1[1] 183,10 DC2[1] 182,65
2009-06-12 09:18:00	DC1[1] 183,10 DC2[1] 182,65
2009-06-12 09:19:00	DC1[1] 183,10 DC2[1] 183,10
2009-06-12 09:20:00	DC1[1] 183,00 DC2[1] 183,10
2009-06-12 09:21:00	DC1[1] 183,00 DC2[1] 183,10
2009-06-12 09:22:00	DC1[1] 183,00 DC2[1] 183,10
2009-06-12 09:23:00	DC1[1] 183,00 DC2[1] 183,10
2009-06-12 09:24:00	DC1[1] 183,00 DC2[1] 183,00
2009-06-12 09:25:00	DC1[1] 182,65 DC2[1] 183,00

DC1[1]과 DC2[1]의 값이 각 봉에서 할당되는 값이 서로 상이한 경우가 있음을 확인할 수 있습니다.

DC1은 (0)으로 선언을 했기 때문에 이전 값 참조가 주종목의 주기로 이전 값을 참고하게 되고 그러므로 DC1[1]은 참조종목의 전봉이 아니라 주종목으로 전봉에서 봤을 때의 가장최근 완성된 봉의 값을 가져오게 됩니다.



반면에 (0,data2)로 선언한 DC2는 선언시 data2의 주기로 이전 값을 참조하라고 선언문에 내용을 추가해

웠으므로 이전 값 참조가 정상적으로 참조 종목을 대상으로 하게 됩니다.

타종목/타주기를 이용하여 수식을 작성 할 경우에 변수선언에 이전 값 참조시의 대상도 고려해서 선언해 주도록 해야 합니다.

② 참조차트 이용시 소급 발생된 매매신호의 자동주문 미처리 위험

참조차트를 이용한 시스템 트레이딩 자동매매에서 기본차트 또는 참조차트가 틱차트로 설정되어 있으면 시스템 매매신호에 따른 자동주문이 실행되지 않는 경우가 발생할 수 있으므로 사용에 주의를 해야합니다.

기본차트 또는 참조차트가 틱차트로 설정되어 있으면, 봉이 완성되는 주기가 서로 일치하지 않기 때문에 기본차트가 완성되는 시점에 과거 봉의 데이터를 재연산하게 되는데 이때 기본차트의 과거 봉에서 참조한 참조차트가 시세의 지연 도달 등으로 변화가 있는 경우 과거 봉 진행 당시에는 발생하지 않았던 매매신호가 재연산으로 인해 새롭게 만들어 질 수 있습니다. 이렇게 만들어진 신호는 과거 봉에서 소급해서 만들어진 것이므로 이 신호에 따른 자동주문은 실행되지 않습니다. 이처럼 소급하여 발생된 매매신호에 대해서는 자동주문 처리를 하지 않기 때문에 매매신호와 계좌 잔고의 불일치가 발생할 수 있으므로 참조차트를 이용한 자동매매에서는 매매신호 발생여부와 자동주문의 실행여부, 주문의 체결 여부를 사용자가 반드시 확인하여야 하며, 불일치 발생시 사용자가 직접 시스템 설정 변경이나 수동주문 등으로 불일치를 해소해 주어야 합니다.

③ 참조차트 이용시 실시간 매매신호와 재조회 매매신호의 불일치 위험

참조차트를 이용한 시스템 트레이딩 자동매매에서 실시간으로 발생된 매매신호와 재조회시 발생된 매매신호가 서로 달라질 수 있으므로 사용에 주의해야 합니다.

참조차트를 이용한 시스템 트레이딩 자동매매에서 기본차트와 참조차트를 동일한 시간주기나 동일한 틱주기로 사용하더라도 참조차트의 시세가 지연 도달하는 등의 이유로 실시간에서 기본차트가 참조하는 참조데이터와 재조회시 기본차트가 참조하는 참조데이터가 변경될 수 있습니다. 이 경우 실시간에서 발생한 매매신호와 재조회시에 표시되는 매매신호가 변경될 수 있습니다.

따라서 참조차트를 이용한 시스템 트레이딩 자동매매에서는 프로그램의 재접속, 차트의 재조회, 시스템 전략의 재적용시에 매매신호의 변경 여부를 사용자가 반드시 확인하여야 하며, 불일치 발생시 사용자가 직접 시스템 설정 변경이나 수동주문 등으로 불일치를 해소해 주어야 합니다.

④ 기타 유의사항

타종목 참조 후 참조한 종목의 가격이나 지표를 이용하여 매매신호를 발생시킬 경우 시뮬레이션과 실전매매의 차이가 발생할 수 있습니다.

타종목 참조를 이용할 경우 시뮬레이션과 차이가 발생할 수 있는 이유는 주종목과 참조종목의 데이터 도달 시간이 서로 상이하여 주종목의 봉이 완성된 이후에 참조 종목의 봉이 완성되는 경우가 생기기 때문입니다.

예를 들면 주종목의 봉이 완성되면서 그 시점까지의 타종목참조의 데이터를 이용하여 확정된 매매신호를 발생시켰는데, 그 이후에 타종목에 데이터가 추가로 들어오게 될 경우 실제매매에서 발생한 매매신호는 사라지지 않지만, 차트를 재조회 하게 될 경우 참조종목에 추가로 들어온 데이터 까지 포함하여 새롭게

매매신호를 발생시키기 때문에 발생되었던 매매신호가 사라질수도, 없던 매매신호가 새롭게 발생할수도 있게 됩니다.(동일한 1분봉 시간주기를 이용하여 매매하는데 주종목은 9시30분봉이 진행되고 있는데, 참조종목에서는 9시29분봉의 데이터가 들어오는 경우)

주종목과 참조종목의 데이터가 서로 다른 시간에 도달되는 이유는 두가지 경우인데, 첫번째는 시장이 다른 데이터(주식종목과 선물)에 대해서 증권회사에서 서로 다른 서버를 사용함으로써 데이터의 처리 시간이 달라지게 되는 경우입니다. 특히 선물과 옵션은 코스콤에서 서버의 시간을 내려주지만, 주식종목에 대해서는 이와 같은 정보를 보내주지 않고 증권회사 서버의 시간을 이용하기 때문에 증권회사마다 동일한 봉의 데이터가 서로 다른 결과를 나타내기도 합니다. 두번째는 선물과 옵션처럼 동일한 시장의 데이터이기 때문에 동일한 서버를 사용한다고 하더라도 데이터 도달시간에 차이가 발생할 수 있는데, 증권회사가 코스콤으로부터 데이터를 전송받을 때 여러개의 라인을 이용하여 데이터를 받기 때문에 코스콤 서버의 시간을 같이 전송받더라도 회선의 속도 차이 때문에 데이터의 도달시간이 달라지는 경우가 생기게 됩니다. 동일한 시장의 종목일 경우 서로 다른 시장간의 종목보다는 시뮬레이션과 실제매매의 차이가 발생하는 빈도수가 적어지겠지만, 시장 구조상 지속적으로 발생할 수밖에 없는 현상입니다.

타종목참조를 이용할 경우 시장 구조적인 문제 때문에 시뮬레이션과 실제매매를 완전히 일치시킬 수 있는 방법은 없지만, 매매의 시간주기를 길게 사용함으로써 발생빈도를 줄일 수 있습니다.

5) 사용자함수 만들기

사용자함수는 시스템식과 지표식등 일반 수식을 만드는 방법과 크게 2가지 차이가 있습니다.

첫번째 사용자함수 이름과 관련이 있고, 두번째는 외부변수의 선언과 관련된 내용입니다.

```
Input : N(1);
var : cnt(0);
array : WeekC[99](0);

5 if DayOfWeek(Date) < DayOfWeek(Date)[1] Then {
    For cnt = 1 To 98 {
        WeekC[cnt] = WeekC[cnt-1][1];
    }
}
10 WeekC[0] = C;

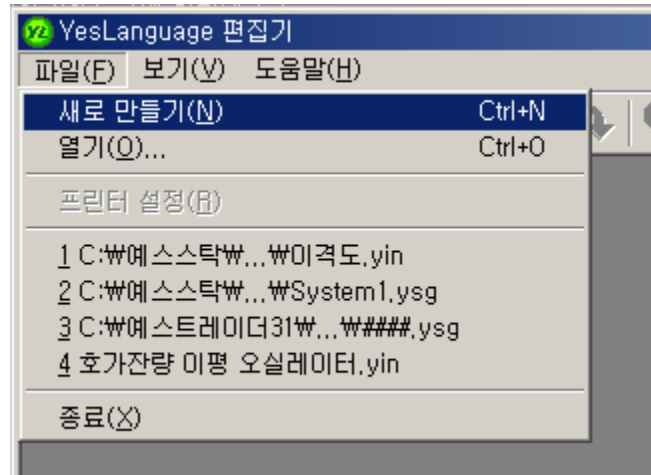
plot1(WeekC[N]);
```

위의 수식은 분봉이나 일봉에서 주봉의 증가값을 계산하기 위해 작성된 식입니다.

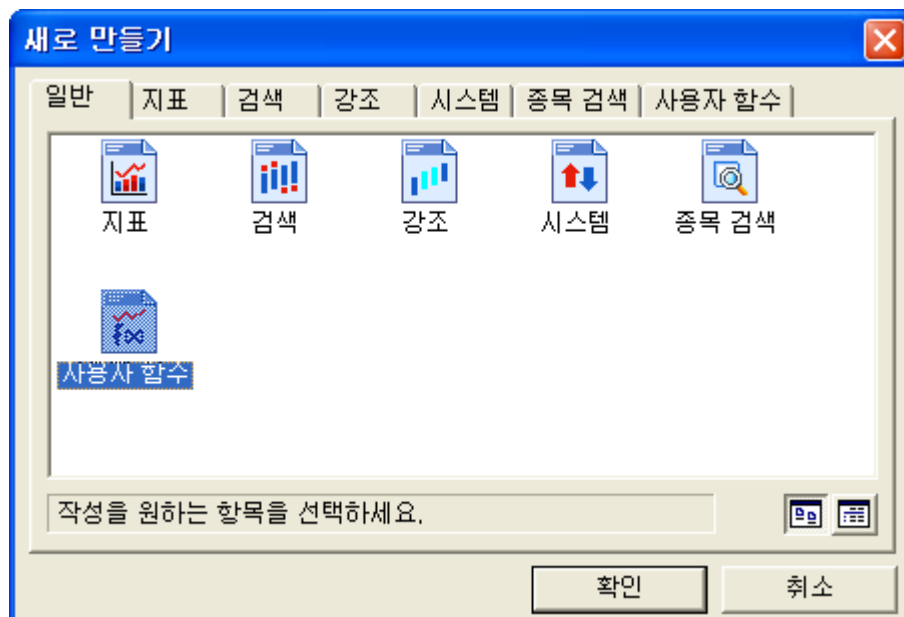
WeekC라는 배열변수를 만들어 방의 갯수는 넉넉히 99개로 만든 후에 이번주의 증가는 WeekC[0]에 저장하고 주가 변경될 때마다 지금까지 저장된 값을 다음 방으로 순차적으로 옮겨 이전 주의 값도 사용할 수 있게 만든 식입니다.

사용자 함수 편집창은 아래와 같은 과정으로 열 수 있다.

예스랭귀지 상단 메뉴 중 [파일]을 선택하고 목록에서 [새로 만들기]를 선택합니다.

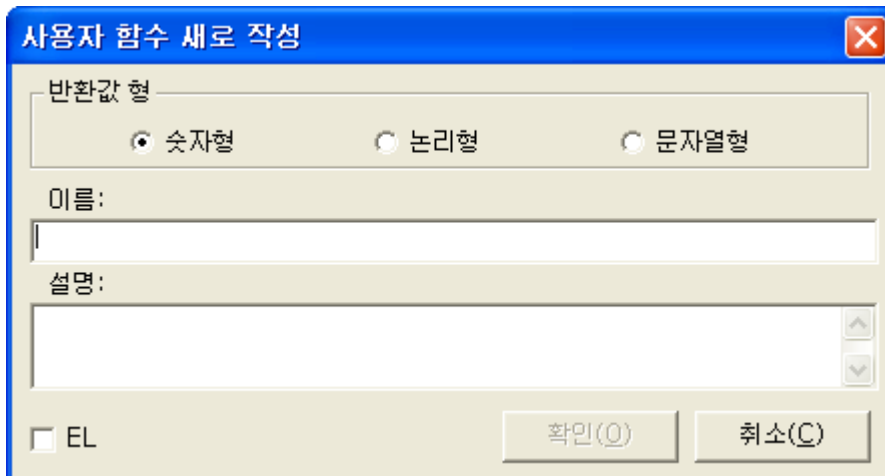


[새로 만들기] 화면은 편집창의 종류를 선택하는 화면입니다



새로 만들기 창에서 사용자함수 아이콘을 선택하고 확인버튼을 클릭합니다.

확인버튼을 클릭하면 사용자 함수 새로 작성이라는 화면이 생성됩니다.



사용자 함수 새로 작성

반환값 형

☒ 숫자형 ☐ 논리형 ☐ 문자열형

이름:

설명:

☐ EL

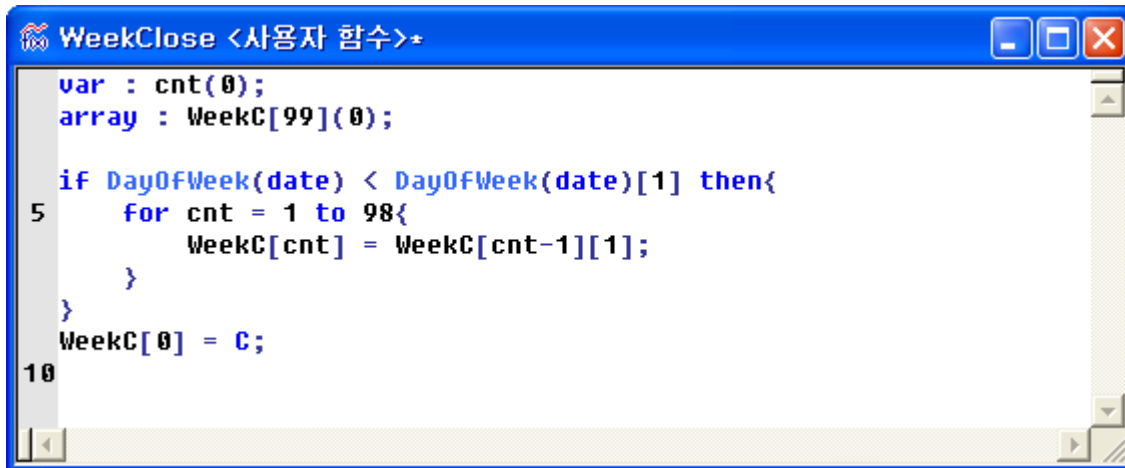
확인(O) 취소(C)

시스템이나 지표식과 같은 일반수식은 이름과 설명만을 입력하게 생성되지만 사용자 함수 새로 작성화면은 반환값 형을 선택해 주어야 합니다.

반환값은 사용자함수에서 최종 계산한 값의 형태로 수치로 최종 반환되는 함수를 만든다면 숫자형을, 최종 계산값이 True나 False로 반환되는 함수를 만든다면 논리형, 수치나 true/false가 아닌 문자열로 반환되는 함수라면 문자열형을 선택하면 됩니다.

만들고자 하는 수식이 수치값을 리턴하는 내용이므로 숫자형을 선택하고 이름을 입력하면 됩니다.

이때 이름을 함수가 계산하여 반환해 주는 내용이 무엇인지를 고려하여 이름을 지어주면 되며 주단위의 종가를 리턴하는 내용이므로 WeekClose라는 명칭으로 입력하고 확인버튼을 누르면 WeekClose라는 이름으로 사용자함수 편집창이 나타납니다.



```

var : cnt(0);
array : WeekC[99](0);

if DayOfWeek(date) < DayOfWeek(date)[1] then{
5   for cnt = 1 to 98{
      WeekC[cnt] = WeekC[cnt-1][1];
    }
}
WeekC[0] = C;
10

```

편집창에 수식을 입력하고 문법 검증(f4)을 하게 되면 아래와 같은 오류메세지가 출력되면서 수식작성이 완성되지 않습니다.

오류			
이름	줄 수	열 수	설명
WeekClose	0	0	함수가 값을 리턴하고 있지 않습니다. 함수는 반드시 값을 리턴해야 합니다.

해당 내용은 수식에서 계산된 값이 사용자함수의 이름에 입력이 되지 않았다는 내용입니다
 사용자함수에서 계산되어 최종 도출된 값은 해당 사용자함수의 이름에 반환이 되도록 식을 작성해 주어야 합니다.

아래와 같이 수식에 최종 도출된 값을 할당받는 WeekC[0]라는 변수를 다시 사용자함수명에 저장시켜 주어야만 사용자함수가 완성이 됩니다.

```

WeekClose <사용자 함수>
var : cnt(0);
array : WeekC[99](0);

if DayOfWeek(date) < DayOfWeek(date)[1] then{
5   for cnt = 1 to 98{
      WeekC[cnt] = WeekC[cnt-1][1];
    }
}
WeekC[0] = C;
10 WeekClose = WeekC[0];
  
```

시스템식에서 따로 계산식을 적지 않고 WeekClose라는 함수만 사용하면 주간 종가를 사용할 수 있습니다.
 하지만 위와 같이 사용자함수를 작성하게 되면 이번 주의 값만을 리턴하게 되고 이전 주의 값이 필요하다면 리턴할 방법이 없습니다. 수식에서는 이미 이전주값이 사용될 수 있게 작성이 되어 있으므로 외부변수로 이전 주를 선택할 수 있게 설정해 주어야 합니다.

```

input : N(1);
var : cnt(0);
array : WeekC[99](0);
5
if DayOfWeek(date) < DayOfWeek(date)[1] then{
    for cnt = 1 to 98{
        WeekC[cnt] = WeekC[cnt-1][1];
    }
10}
WeekC[0] = C;
WeekClose = WeekC[N];

```

배열변수의 방번호를 지정하여 Weekclose에 할당할 수 있게 외부변수를 만들어 처리해야 합니다.
 외부변수로 N이라는 이름의 변수를 만들고 WeekC 배열변수의 방번호를 선택하여 WeekClose에 할당하도록 식을 구성하였습니다. 이때 시스템식이나 지표식등 일반수식과 같이 외부의 변수의 초기값으로 특정숫자로 지정을 하게 되면 오류메세지가 나타납니다.

이름	줄 수	열 수	설명
WeekClose	2	11	함수의 입력변수는 초기값을 갖지 않습니다. 데이터 유형(예:Numeric)을 지정해야 합니다.
WeekClose	2	11	입력변수 데이터 유형이 올바릅니다. 예: Numeric, NumericSeries...

위 내용은 사용자함수의 외부변수는 특정숫자등으로 지정하지 말고 유형만을 지정하라는 의미이고 해당 변수에 숫자를 주입한다면 numeric등으로 True나 false를 주입한다면 logical, 문자열을 주입한다면 string으로 선언해 주어야 함을 의미합니다.

위 식을 아래와 같이 변경하면 함수가 완성됩니다.

```

WeekClose <사용자 함수>
input : N(numeric);
var : cnt(0);
array : WeekC[99](0);

5 if DayOfWeek(date) < DayOfWeek(date)[1] then{
    for cnt = 1 to 98{
        WeekC[cnt] = WeekC[cnt-1][1];
    }
}
10 WeekC[0] = C;
WeekClose = WeekC[N];

```

매개변수를 하나를 가지는 함수가 완성이 됩니다.

수식에서는 WeekClose(0), WeekClose(1)등으로 사용하면 됩니다.

이전에 지표식으로 만든식과 사용자 함수가 리턴되는 값이 동일한지 확인하여 같다면 완성이 됩니다.

```

Input : N(1);
var : cnt(0);
array : WeekC[99](0);

5 if DayOfWeek(Date) < DayOfWeek(Date)[1] Then {
    For cnt = 1 To 98 {
        WeekC[cnt] = WeekC[cnt-1][1];
    }
}
10 WeekC[0] = C;
plot1(WeekC[N]);

```

↓↓↓↓↓↓↓↓↓↓

```

plot1(WeekClose(1));

```

6) 신호체계

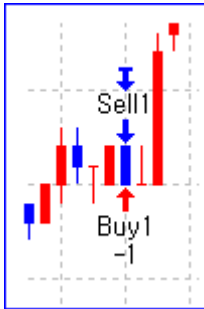
가) 수식에 의해 발생한 신호는 유효한 신호로 합니다.

사용자가 시스템을 작성한 결과에 의해 동일 시점에 여러 번의 진입청산이 이루어 질 경우 횡수에 상관없이 모두 유효한 신호로 합니다. 그러므로 시스템 작성시 동일한 시점에 진입,청산이 동시에 이루어지지 않도록 유의하여 작성하여야 합니다.

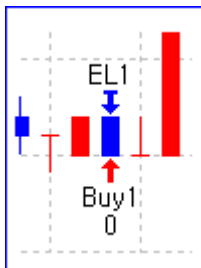
예)

다음과 같이 매수진입(Buy1)과 같은 시점에 매도진입(Sell1)이 동시에 발생되면 매수진입이 들어가고

동시에 매도진입에 의해 매수청산이 되고 매도진입이 이루어 집니다..



또한 다음과 같이 진입(Buy1)과 청산(EL1)이 동시에 발생하면 진입이 완성됨과 동시에 청산신호도 완성되게 됩니다.

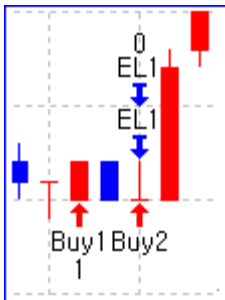


나) 신호의 청산은 동일한 시점에 발생한 진입 신호까지 청산합니다.

즉 하나의 청산 신호로 여러 개의 진입신호가 청산 될 수 있습니다.

예)

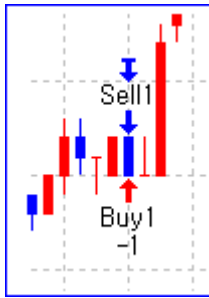
다음은 진입(Buy1)이 완성되고 다음으로 청산(EL1)이 완성되는 시점에 또 다른 진입(Buy2)이 있을 경우 청산(EL1)으로 두 진입이 청산됩니다.



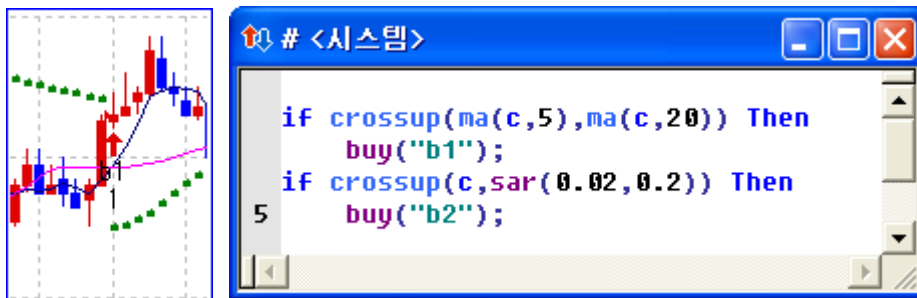
다) 동일한 시점에 매수진입과 매도진입이 발생할 경우 매수 진입이 우선합니다.

예)

다음과 같이 매수진입(Buy1)과 같은 시점에 매도진입(Sell1)이 동시에 발생되면 매수진입 우선 원칙에 따라 매수진입이 먼저 들어가고 동시에 매도진입에 의해 매수청산이 이루어 지고 매도진입이 이루어 집니다.



라) 동일한 시점에 동일한 진입이 동시에 이루어질 경우 수식에 기술된 순서가 먼저인 진입이 우선하게 됩니다.



마) 동일한 봉에서 신호의 우선 순위는 확정된 시점이 빠른 신호가 우선하게 됩니다.

장 중에 자동 업데이트 되고있는 최종 바의 신호는 자동 업데이트 되는 가격을 순서대로 받아들여 분석하여 신호를 완성한다. 그러므로 실제 장 움직임을 그대로 반영하게 됩니다. 그러나 이전 과거 데이터에 적용하였을 때는 가격의 움직임을 알 수 없습니다. 실제로 알 수 있는 가격은 해당 봉의 시가와 고가 저가 종가 뿐입니다. 그러므로 움직임에 대한 가정을 하여 적용하여야 합니다.

그러므로 실제 가격의 움직임과 이후 가정한 가격의 움직임이 서로 다를 경우 신호가 변경될 수 있습니다. 사용자는 이점을 유의하여 트레이딩 하여야 합니다.

7) 과거 데이터 움직임 가설

봉의 주가움직임은 시가로 시작하여 저가와 고가를 움직이며 종가로 마무리 하게 됩니다 .

이때 시가와 종가는 시간상으로 확정된 값을 가지므로 항상 일치된 정보를 얻을 수 있습니다 .

그러나 고가와 저가의 움직임은 즉 시가와 종가 사이 시간의 데이터에 대한 움직임을 알 수 있는 데이터를 가지고 있지 않으므로 알 수 없습니다.

실제 해당 봉의 모든 틱 데이터를 가지고 움직임을 알 수는 있으나 현실적으로 시간이 매우 오래 걸리고 데이터의 범위도 매우 많으므로 어려움이 있어 반영하기는 무리가 있습니다.

그러므로 어떤 형태로든지 시가와 종가 사이의 데이터 움직임을 정의가 필요하며 봉 중간에 이루어지는 신호의 움직임을 보장할 수 없게 됩니다.

일반적으로 과거 봉 데이터의 움직임을 보면 시가로 진입하여 고가나 저가로 움직일 경우 움직임이 짧은 곳의 움직임을 먼저 일어나는 경우가 반대의 경우보다 많으므로 시가에 보다 가까운 쪽의 움직임을 먼저 이루어진 것으로 간주합니다 .

또한 시가와 고가,저가의 가격 움직임을 같을 경우는 저가가 먼저 이루어진 것으로 간주합니다. 위와 같은 봉 내부의 움직임 가설로 봉 중간에 이루어 지는 신호를 결정하게 됩니다 .

그러나 위와 같은 가설은 실제의 봉 움직임과 다를 수 있으므로 실제 장 운영 시 적용한 신호와 이후 다시 적용(과거 데이터로)한 신호의 결과가 다르게 나타날 수 있으므로 이점을 유의 하여 시스템을 작성하고 또한 트레이딩 시 유의하여 적용하여야 합니다.

위의 가설로 봉 데이터의 주가움직임을 정리하면 다음과 같습니다..

가. 시가와 고가 시가와 저가의 가격폭이 같을 경우 봉 데이터의 움직임

시가 → 저가 → 고가 → 종가의 데이터 흐름으로 한다.

예) 진입신호가 다음과 같을 경우

```
Buy("Buy1", AtLimit, C-0.01);
```

```
Sell("Sell1", AtLimit, C+0.01);
```

위와 같은 두개의 진입 신호가 동시에 만족하고 현재의 봉의 시가와 고가, 시가와 저가의 가격폭이 같을 경우 다음과 같은 신호 결과를 나타낸다.



시가에서 저가로 먼저 움직임으로 인해 Buy1이 먼저 완성되고 이후 저가에서 고가로 움직이므로 다시 Sell1이 완성되면 이때 매수진입은 청산되게 되어 해당 봉에 최종적으로 매도 1계약(수량)이 진입으로 남게된다.

나. 시가와 고가 폭보다 시가와 저가의 가격폭이 작을 경우 봉 데이터의 움직임

시가 → 저가 → 고가 → 종가의 데이터 흐름으로 한다.

예) 진입신호가 다음과 같을 경우

```
Buy("Buy1", AtLimit, C-0.01);
```

```
Sell("Sell1", AtLimit, C+0.01);
```

위와 같은 두개의 진입 신호가 동시에 만족하고 현재의 봉의 시가와 고가 폭보다 시가와 저가의 가격폭이 작을 경우 다음과 같은 신호 결과를 나타낸다.



시가에서 저가로 먼저 움직임으로 인해 Buy1이 먼저 완성되고 이후 저가에서 고가로 움직이므로 다시 Sell1이 완성되면 이때 매수진입은 청산되게 되어 해당 봉에 최종적으로 매도 1계약(수량)이 진입으로 남게된다.

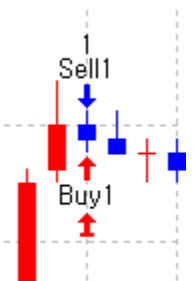
다. 시가와 고가 폭보다 시가와 저가의 가격폭이 클 경우 봉 데이터의 움직임
시가 → 고가 → 저가 → 종가의 데이터 흐름으로 한다.

예) 진입신호가 다음과 같을 경우

```
Buy("Buy1", AtLimit, C-0.01);
```

```
Sell("Sell1", AtLimit, C+0.01);
```

위와 같은 두개의 진입 신호가 동시에 만족하고 현재의 봉의 시가와 고가 폭보다 시가와 저가의 가격폭이 클 경우 다음과 같은 신호 결과를 나타낸다.



시가에서 고가로 먼저 움직임으로 인해 Sell1이 먼저 완성되고 이후 고가에서 저가로 움직이므로 다시 Buy1이 완성되면 이때 매도진입은 청산되게 되어 해당 봉에 최종적으로 매수 1계약(수량)이 진입으로 남게 된다

8) 예비신호

시스템 트레이딩시 차트창에 나타나는 예비신호는 매매함수에서 진입과 청산함수에 대해서만 차트창의 마지막봉에서 봉이 완성되기 전에 신호가 일시적으로 만족하면 속이 비어있는 신호로 표시하고 차트상단에 회색으로 예비신호의 종류를 보여줍니다.

이 예비신호는 수식에의해 발생할 수 있는 신호를 표시하는 것으로 이 신호가 확정된 신호가 아니므로 예비신호 사용시 유의하여 사용하시기 바랍니다.

9) 실전매매와 시뮬레이션의 차이

가. 강제청산 중 최대수익대비하락(SetStopTrailing) 청산 사용

최대수익대비하락(SetStopTrailing) 청산은 실전매매와 시뮬레이션의 결과가 달라질 수 있습니다. 설정하는 값이 작으면 작을 수록 결과에 실전매매와 시뮬레이션 간의 차이가 크게 발생하는데 이는 실제 매매에서는 모든 체결틱의 데이터를 이용하여 신호를 발생시키는데 반해 과거 봉에서는 모든 체결틱이 봉안에 없으므로 봉의 시가, 고가, 저가, 종가의 데이터만을 이용하여 주가 움직임 가설에 의해 움직인 것으로 판단하여 테스트되기 때문입니다.

과거의 봉데이터에 모든 체결틱 데이터가 제공되지 않는한 SetStopTrailing(최대수익대비하락) 청산에서 실제매매와 시뮬레이션의 불일치는 피할 수 없습니다.

이 두가지의 갭을 줄이기 위해서는 설정에 사용되는 값을 평균적인 봉의 길이보다 크게 사용하는 방법과 청산방법을 '수익금액대비'로 설정하지 말고 '최고가격대비'로 설정하는 방법도 도움이 될 수 있습니다.

수익금액대비로 설정할 경우 매수진입금액이 200.00이고 최소수익금액이 1%, 수익금액대비를 1%로 셋팅할 경우 202.00으로 가격이 상승한 이후부터는 1틱만 하락해도 청산주문이 발생하게 됩니다. 바로 '수익금액대비'로 설정했기 때문으로 '수익금액대비'를 '최고가격대비'와 자주 혼동하게 되는데 최고가격대비는 202.00이후의 최고가격*0.01 이지만 '수익금액대비'는 (202.00이후의 (최고가격-진입가격)*0.01 이므로 너무 작은 값을 사용할 경우 최소수익 이후에 한두틱만 하락해도 바로 청산하게 됩니다.

최대수익대비하락에서 이와 같은 현상이 발생하는 이유는 자기봉의 고가와 저가(매수인 경우는 고가, 매도인 경우는 저가)를 이용하기 때문이므로 갭을 줄이기 위한 또하나의 방법은 아래 예제와 같이 자기봉의 고가와 저가를 포함하지 않는 trailingStop 로직을 사용하면 해결할 수 있는 문제입니다.

```

#최고가격대비 Point Trailing Stop
//매수진입이후 전봉까지의 최고가격대비 0.7pt하락하면 매수청산
//매도진입이후 전봉까지의 최저가격대비 0.7pt상승하면 매도청산
input : TsValue(0.7);
5 var : Hvalue(0),Lvalue(0);

If MarketPosition() == 1 Then {
    Hvalue = Highest(H,BarsSinceEntry+1);
    ExitLong("trailStop_EL", AtStop, Hvalue - TsValue);
10}
If MarketPosition() == -1 Then {
    Lvalue = Lowest(L,BarsSinceEntry+1);
    ExitShort("trailStop_ES", AtStop, Lvalue + TsValue);
}
15

```

```

#최고가격대비 Percent Trailing Stop
//매수진입이후 전봉까지의 최고가격대비 0.7%하락하면 매수청산
//매도진입이후 전봉까지의 최저가격대비 0.7%상승하면 매도청산
input : TsValue(0.7);
5 var : Hvalue(0),Lvalue(0);

If MarketPosition() == 1 Then {
    Hvalue = Highest(H,BarsSinceEntry+1);
    ExitLong("trailStop_EL%", AtStop, Hvalue - Hvalue*TsValue/100);
10}
If MarketPosition() == -1 Then {
    Lvalue = Lowest(L,BarsSinceEntry+1);
    ExitShort("trailStop_ES%", AtStop, Lvalue + Lvalue*TsValue/100);
15}

```

나. SetStopEndOfDay

당일청산을 setstopendofday함수 또는 시스템트레이딩 설정창의 당일청산 시간으로 설정할 경우 시뮬레이션과 차이가 발생하게 됩니다.

예를들어 5분봉 시스템을 이용하면서 당일청산을 랭귀지 내에서 setstopendofday(1504); 로 작성해 놓고 시스템트레이딩 설정창에서 강제청산 시점을 조건만족시 즉시로 선택해 놓을 경우 실제 매매는 15시04분00초 이후에 체결데이터가 들어오는 즉시 당일청산 주문을 실행됩니다. 하지만 과거봉의 데이터는 봉의 시가, 고가, 저가, 종가만 있으므로 15시04분00초의 가격을 알지못하기 때문에 15시봉의 종가에 청산한 것으로 시스템리포트에 나타나게 됩니다.

당일청산 시간에 따라 시뮬레이션과 실제매매의 차이가 생기는 현상은 봉데이터 저장의 특성상 부득이한 것이고 시스템 성과의 본질적인 변화를 주는 내용은 아니기 때문에 중요도는 떨어지지만 역시 차이가 생기는 부분입니다

if 문으로 시간조건을 주어 작성한 청산은 시뮬레이션과 차이가 발생하지 않습니다.

```
if stime == 145500 then
{
    exitlong();
5    exitshort();
}
```

5분봉일 경우 14시55분봉(14:55:00초~14:59:59초)의 종가에 청산한 것으로 시뮬레이션 됩니다.
(실제 주문은 15:00:00초 이후의 첫번째 시세데이터가 들어올때 실행 됨)

다. 타종목참조

타종목참조후 참조한 종목의 가격이나 지표를 이용하여 매매신호를 발생시킬 경우 시뮬레이션과
실전매매의 차이가 발생할 수 있습니다.

타종목 참조를 이용할 경우 시뮬레이션과 차이가 발생할 수 있는 이유는 주종목과 참조종목의
데이터 도달 시간이 서로 상이하여 주종목의 봉이 완성된 이후에 타종목참조 종목의 봉이 완성되는
경우가 생기기 때문입니다.

예를들면 주종목의 봉이 완성되면서 그 시점까지의 타종목참조의 데이터를 이용하여 확정된
매매신호를 발생시켰는데, 그 이후에 타종목에 데이터가 추가로 들어오게 될 경우 실제매매에서
발생된 매매신호는 사라지지 않지만, 차트를 재조회 하게 될 경우 참조종목에 추가로 들어온 데이터
까지 포함하여 새롭게 매매신호를 발생시키기 때문에 발생되었던 매매신호가 사라질수도 있고 없던
매매신호가 새롭게 발생할수도 있게 됩니다.

주종목과 참조종목의 데이터가 서로 다른 시간에 도달되는 이유는 두가지 경우인데 첫번째는
시장이 다른 데이터(주식종목과 선물)에 대해서 증권회사에서 서로 다른 서버를 사용함으로써
데이터의 처리 시간이 달라지게 되는 경우입니다. 특히 선물과 옵션은 코스콤에서 서버의 시간을
내려주지만 주식종목에 대해서는 이와 같은 정보를 보내주지 않고 증권회사 서버의 시간을
이용하기 때문에 증권회사마다 동일한 봉의 데이터가 서로 다른 결과를 나타내기도 합니다.
두번째는 선물과 옵션처럼 동일한 시장의 데이터이기 때문에 동일한 서버를 사용한다고 하더라도
데이터 도달시간에 차이가 발생할 수 있습니다. 증권회사가 코스콤으로부터 데이터를 전송받을 때
여러개의 라인을 이용하여 데이터를 받기 때문에 코스콤 서버의 시간을 같이 전송받더라도 회선의
속도 차이 때문에 데이터의 도달시간이 달라지는 경우가 생기게 됩니다. 동일한 시장의 종목일 경우
서로 다른 시장간의 종목보다는 시뮬레이션과 실제매매의 차이가 발생하는 빈도수가 적어지겠지만
시장 구조상 지속적으로 발생할 수밖에 없는 현상입니다.

타종목참조를 이용할 경우 시장 구조적인 문제 때문에 시뮬레이션과 실제매매를 완전히 일치시킬 수 있는 방법은 없지만, 매매의 시간주기를 길게 사용함으로써 발생빈도를 줄일 수 있습니다.

라. 일간갭보정 차트 이용

분,틱봉의 일간갭 보정 기능 사용은 시뮬레이션과 실제매매의 차이가 발생하게 되는 주요한 원인이 됩니다.

기본차트 속성창에서 분,틱봉의 일간갭보정 기능을 체크하게 되면 오늘의 데이터는 그대로 두고 오늘 이전의 모든 데이터에 대해서 발생된 갭만큼의 비율로 과거의 차트를 보정하게 됩니다. 만일 갭보정 차트에 시스템을 적용하게 된다면 실제매매에서 발생되었던 매매신호가 다음날 갭보정에 의해서 사라지거나 새로 생겨날 수 있고 신호는 그대로 유지되었다고 오늘의 가격이 바뀌게 되므로 매매의 손익도 실제의 매매와 매일 다르게 변경되게 됩니다.

차트속성창에서 분, 틱봉의 일간갭보정 기능을 이용할 경우 시뮬레이션 결과와 실제매매의 결과를 전혀 다르게 왜곡시키므로 일간갭을 보정시킨 상태에서 시스템매매를 하거나 시스템테스트를 하는 것은 적절한 방법은 아닙니다. 이 기능은 시각적인 분석을 할때만 제한적으로 사용해야 하는 기능입니다.

.

라. 틱차트 이용시 실시간 매매신호와 재조회 매매신호의 불일치 위험

시스템 트레이딩 자동매매에서 기본차트 또는 참조차트를 틱차트로 사용할 경우 실시간으로 발생한 매매신호와 재조회시 발생한 매매신호가 서로 달라질 수 있으므로 사용에 주의해야 합니다.

틱차트는 네트워크 상에서 데이터의 유실, 데이터 처리 속도 향상을 위한 틱데이터 묶음 처리 등으로 실시간 업데이트 데이터와 재조회 데이터의 차이가 발생할 수 있습니다. 틱차트는 봉을 만드는 구조적인 특성상 중간에 하나의 틱데이터만 유실되어도 이후의 모든 차트 데이터가 변경되기 때문에 틱차트를 이용한 시스템 자동매매에서는 실시간 매매신호와 재조회시 매매신호가 달라지는 경우가 빈번히 발생할 수 있습니다. 따라서 틱차트를 이용한 시스템 트레이딩 자동매매에서는 프로그램의 재접속, 차트의 재조회, 시스템 전략의 재적용시에 매매신호의 변경 여부를 사용자가 반드시 확인하여야 하며, 불일치 발생시 사용자가 직접 시스템 설정 변경이나 수동주문 등으로 불일치를 해소해야 합니다.

마. 기타

1. Atstop / Atlimit

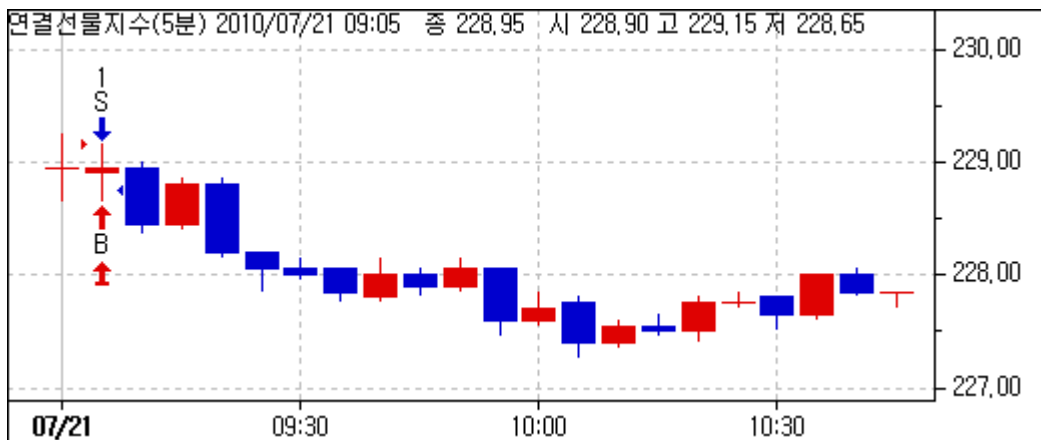
가령 두번째 봉에서 시초가대비 +0.2의 가격에 도달하면 매수하고, -0.2의 가격에 도달하면

매도하는 식을 작성한다면 아래와 같이 작성할 수 있습니다.

```
if dayindex == 0 then
{
    buy("B", atstop, day0Open+0.2);
5    sell("S", atstop, day0Open-0.2);
}
```

두번째 봉에서 시초가 ± 0.2 를 넘나드는 긴 봉이 만들어 졌을 때 과거데이터는 주가움직임 가설에 의해서 시가대비 고가와 저가의 폭을 계산하여 고가의 폭이 더 크거나 같으면 저가가 형성된 후 고가가 형성되었다고 보게되고 저가의 폭이 더 크다면 고가가 형성된 후 저가가 형성되었다고 보고 됩니다.

따라서 실제움직임과는 상관없이 고가의 폭이 크냐 또는 저가의 폭이 크냐에 따라서 매수신호가 먼저 발생되기도 하고 매도신호가 먼저 발생되기도 합니다.



위 그림에서 주가 움직임 가설에 의해서 시가 \rightarrow 저가 \rightarrow 고가 \rightarrow 종가 순서로 발생되었다고 가정하기 때문에 매도신호가 먼저 발생하고 나중에 매도청산과 매수신호가 발생하게 됩니다. 하지만 실제 매매에서는 고가가 먼저 형성된 후에 저가가 형성될 수도 있는데, 실제매매에서 고가가 먼저 형성되었다면 매수신호가 먼저 발생하고 이후에 매수청산신호와 매도신호가 발생하게 될 것입니다.

아래와 같은 수식도 위와 동일한 문제를 가지게 됩니다.

```

if dayindex == 0 then
{
  buy("B1", atstop, dayOpen+0.2);
5  buy("B2", atlimit, dayOpen-0.2);
}

```

주가 움직임의 가설을 사용하는 한 근본적으로 해결할 수 있는 방법은 없으므로 가급적 이와 같은 상황을 줄이는 방법은 atstop과 atlimit 주문을 이용할때 한봉에서 두가지 조건을 모두 만족하지 않을 정도의 큰 값을 사용하는 것이 문제 해결을 위한 방법이 될 수 있습니다.

2. 지표의 계산 방식에 따른 차이

보통 시뮬레이션은 장기간의 데이터로 테스트하고 실제매매는 짧은 기간의 데이터로 매매를 하게 되는데, 데이터의 시작 시점(또는 데이터의 조회기간)에 따라서 신호가 달라지는 경우가 발생할 수 있습니다.

예를 든다면 지수이동평균선을 이용하는 지표(지수이동평균선, 스토크스틱, macd..등)는 지수이동평균선의 특성 때문에 데이터의 조회 기간이 달라지면 지표의 값이 변경될 수 있고 이에따라 매매신호도 변경될 수 있습니다.

(지수이동평균선의 기간 값을 20으로 사용하는 경우 20개봉 만을 이용하여 지수이동평균선을 그리는 것이 아니라 그 이전의 데이터들도 이용하여 지표를 그린다.)

또 다른 경우는 accum 함수를 이용하거나 차트 전체에서 값을 누적하는 OBV 지표 처럼 데이터의 시작시점에 따라서 값이 달라지는 지표를 이용할 경우에도 매매신호가 조회기간에 따라서 달라지게 됩니다.

이외에 사용된 지표의 계산에 필요한 충분한 데이터가 조회되지 않았을 경우에도 매매신호가 영향을 받을 수 있습니다.(500일 이동평균선을 이용하는데 데이터를 300개만 조회한 경우)

지표의 계산방식에 따른 시뮬레이션과 실제매매의 차이의 오류를 줄이려면 가급적 긴 기간의 데이터를 불러놓고 시스템매매를 적용해야 합니다.

5. 함수설명

1) 주문함수

buy

설명 : 신규 매수진입 (기존에 반대포지션이 있으면 반대포지션을 청산(매도청산)하고 진입)

함수 : Buy("명칭",신호타입,신호가격,수량);

"명칭"

신호명칭, 하나의 전략에서 동일한 명칭으로 중복사용 불가

신호타입

생략 가능, 기본값은 OnClose

- OnClose : 기준 Bar 종가에 주문

- Atmarket : 다음 Bar 시가에 주문

- AtLimit : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이하의 시세 발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이상의 시세 발생 시 주문이 발생합니다.

- AtStop : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이상의 시세 발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이하의 시세 발생 시 주문이 발생합니다

신호가격

AtLimit, AtStop일 때 신호가격 지정.

OnClose, AtMarket의 경우는 필요하지 않으며 수량을 적용할 경우 DEF로 입력하면됩니다.

수량

주문수량 지정

여기에 주문 수량을 설정하면 트레이딩 설정창의 주문수량은 적용되지 않습니다.

Sell

설명 : 신규 매도진입 (기존에 반대포지션이 있으면 반대포지션을 청산(매수청산)하고 진입)

함수 : Sell("명칭",신호타입,신호가격,수량);

"명칭"

신호명칭, 하나의 전략에서 동일한 명칭으로 중복사용 불가

신호타입

생략 가능, 기본값은 OnClose

- OnClose : 기준 Bar 종가에 주문

- Atmarket : 다음 Bar 시가에 주문

- AtLimit : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이하의

시세발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이상의 시세발생 시 주문이 발생합니다.

- AtStop : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이상의 시세발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이하의 시세발생 시 주문이 발생 한다.

신호가격

AtLimit, AtStop일 때 신호가격 지정.

OnClose, AtMarket의 경우는 필요하지 않으며 수량을 적용할 경우 DEF로 입력하면 된다.

수량

주문수량 지정

여기에 주문 수량을 설정하면 트레이딩 설정창의 주문수량은 적용되지 않습니다.

ExitLong

설명 : 매수청산 (매수포지션에 대한 청산)

함수 : ExitLong("명칭",신호타입,신호가격,"진입명",수량,수량옵션);

"명칭"

신호명칭, 하나의 전략에서 동일한 명칭으로 중복사용 불가

신호타입

생략 가능, 기본값은 OnClose

- OnClose : 기준 Bar 종가에 주문

- Atmarket : 다음 Bar 시가에 주문

- AtLimit : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이하의 시세발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이상의 시세발생 시 주문이 발생.

- AtStop : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이상의 시세발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이하의 시세발생 시 주문이 발생.

신호가격

AtLimit, AtStop일 때 신호가격 지정.

OnClose, AtMarket의 경우는 필요하지 않으며 수량을 적용할 경우 DEF로 입력하면 된다.

"진입명"

진입명을 지정하면 해당진입이 있을 경우에만 발동한다.

진입명을 지정하지 않을 경우 " "로 표시하면 된다.

수량

주문수량 지정

여기에 주문 수량을 설정하면 트레이딩 설정창의 주문수량은 적용되지 않는다.

수량옵션

청산 시 청산 수량의 옵션

해당옵션을 생략할 경우 옵션의 적용은 0으로 한다.

0 : 현재 진입되어 있는 모든 진입 신호에서 일괄적으로 청산 수량 만큼 청산하게 된다

1 : 현재 진입되어 있는 진입 수량의 총합에서 정의된 청산수량만 청산하게 된다.

2 : 청산함수를 1개의 진입신호 당 여러 번 수행하도록 한다.

(청산함수는 기본적으로 1개의 진입신호에 대해 한번씩만 발생한다. 그러므로 1개의 진입에 대해 여러 번 청산신호를 발생하게 해서 분할로 청산할 경우 분할되는 수 만큼 청산함수가 필요하게 된다. 옵션을 2로 하게 되면 하나의 청산으로 동일조건으로 여러 번 청산을 수행하도록 식을 구현할 수가 있다)

ExitShort

설명 : 매도청산 (매도포지션에 대한 청산)

함수 : ExitLong("명칭",신호타입,신호가격,"진입명",수량,수량옵션);

"명칭"

신호명칭, 하나의 전략에서 동일한 명칭으로 중복사용 불가

신호타입

생략 가능, 기본값은 OnClose

- OnClose : 기준 Bar 종가에 주문

- Atmarket : 다음 Bar 시가에 주문

- AtLimit : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이하의 시세발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이상의 시세발생 시 주문이 발생.

- AtStop : 매수 포지션 주문(Buy, ExitShort)인 경우에는 지정한 신호가격 이상의 시세발생 시 매도 포지션 주문(Sell, ExitLong)인 경우에는 지정한 신호가격 이하의 시세발생 시 주문이 발생.

신호가격

AtLimit, AtStop일 때 신호가격 지정.

OnClose, AtMarket의 경우는 필요하지 않으며 수량을 적용할 경우 DEF로 입력하면 된다.

"진입명"

진입명을 지정하면 해당진입이 있을 경우에만 발동한다.

진입명을 지정하지 않을 경우 " "로 표시하면 된다.

수량

주문수량 지정

여기에 주문 수량을 설정하면 트레이딩 설정창의 주문수량은 적용되지 않는다.

수량옵션

청산 시 청산 수량의 옵션

해당옵션을 생략할 경우 옵션의 적용은 0으로 한다.

0 : 현재 진입되어 있는 모든 진입 신호에서 일괄적으로 청산 수량 만큼 청산하게 된다

1 : 현재 진입되어 있는 진입 수량의 총합에서 정의된 청산수량만 청산하게 된다.

2 : 청산합수를 1개의 진입신호 당 여러 번 수행하도록 한다.

(청산합수는 기본적으로 1개의 진입신호에 대해 한번씩만 발생한다. 그러므로 1개의 진입에 대해 여러 번 청산신호를 발생하게 해서 분할로 청산할 경우 분할되는 수 만큼 청산합수가 필요하게 된다. 옵션을 2로 하게 되면 하나의 청산으로 동일조건으로 여러 번 청산을 수행하도록 식을 구현할 수가 있다)

SetStopEndofday

설명 : 당일청산

함수 : SetStopEndofday(시간지정)

시간지정 : 당일청산 시간을 지정.

시간은 24시간 기준이며 6자리 HHMMSS 혹은 4자리 HHMM으로 설정한다.

예제 : SetStopEndofday(144500); →14시 45분 청산

SetStopEndofday(1445); →14시 45분 청산

※ SetStopEndofday(0);

시간을 0으로 입력하면 당일청산이 해제된다.

※ SetStopEndofday();로 작성하면 당일 마지막봉 청산으로 설정된다. 이 설정은 시뮬레이션 용도로만 사용하고 실제 매매에서는 시간을 지정하고 사용해야 한다.

※ 당일청산은 지정한 시간에 포지션이 있으면 청산을 하고 당일 진입을 하지 않는다는 설정입니다. 지정한 시간 이후부터 0시까지의 진입을 할수 없으므로 이용에 유의하시

기 바랍니다.

SetStopProfitTarget

설명 : 목표이익 청산

함수 : SetStopProfittarget(목표수익값,Method);

목표수익값

목표수익값, 이 값이 0이면 목표수익 설정이 해지 된다.

Method

PercentStop → % 설정

PointStop → 포인트(원) 설정

생략하면 % 설정

예제 : 10% 수익시 청산

SetStopProfittarget(10);

SetStopProfittarget(10,PercentStop);

2 포인트 수익 청산

SetStopProfittarget(2,PointStop);

10000원 수익시 청산

SetStopProfittarget(10000,PointStop);

목표수익 청산 해제

SetStopProfittarget(0);

SetStopLoss

설명 : 손절매

함수 : SetStopLoss(손절매값,Method);

손절매값

손절매값, 이 값이 0이면 목표수익 설정이 해지 된다.

Method

PercentStop → % 설정

PointStop → 포인트(원) 설정

생략하면 % 설정

예제 : 10% 손실시 청산

SetStopLoss(10);

SetStopLoss(10,PercentStop);

2 포인트 손실 청산

SetStopLoss(2,PointStop);

10000원 손실 시 청산

SetStopLoss(10000,PointStop);

손절매 청산 해제

SetStopLoss(0);

SetStopTrailing

설명 : 최대수익대비하락

함수 : SetStopTrailing(수익감소값,최소수익값,Method,tr_Method);

수익감소값 : 수익 감소 허용값, 이 값이 0이면 목표수익 설정이 해지 된다.

최소수익값 : 최소 달성해야 하는 수익값.

Method : PercentStop → % 설정, PointStop → 포인트(원) 설정, 생략하면 % 설정

tr_Method | 생략 가능, 기본값은 0(수익금액대비)

0 : 수익 금액 대비, 진입 후 진입 가격과 최고가격과의 차에 의한 대비

1 : 최고 가격 대비, 진입 후 최고가격대비 하락 시에 청산

예제 : 최소 10% 수익 이후에 수익의 50%가 감소하면 청산

SetStopTrailing(50,10);

SetStopTrailing(50,10,PercentStop);

SetStopTrailing(50,10,PercentStop,0);

최소 20% 수익 이후에 최고 가격대비 5% 하락하면 청산

SetStopTrailing(5,20,PercentStop,1);

최소 3 포인트 수익 이후에 1포인트 수익 감소하면 청산

SetStopTrailing(1,3,PointStop);

SetStopTrailing(1,3,PointStop,0);

※ PointStop의 포인트가 기준이 되므로 경우에는 수익금액대비/최고가격대비 모두 같은 동작을 한다.

※ SetStopTrailing은 실시간에서는 진행중인 봉의 모든 체결틱의 데이터를 이용하여 강제청산이 적용되지만 시뮬레이션에서는 모든 체결틱이 아닌 봉의 시가, 고가, 저가, 종가의 데이터만을 이용하여 만들어진 가설에 따라 움직인것으로 판단하여 테스트 되기 때문에 너무 작은 값을 입력할 경우 실제매매와 시뮬레이션결과에 차이가 발생할 수 있다. 차이를 최소화 하기 위해서는 일반적인 봉 움직임의 크기보다 더 큰 값을 입력해야 한다.

SetStopInactivity

설명 : 최소가격변화

함수 : SetStopInactivity(최소가격,기간,Method);

최소가격값 :

최소한 움직임 폭, 이 값이 0이면 목표수익 설정이 해지 된다.

기간 :

가격 감시 기간(봉수로 지정)

Method

PercentStop → % 설정

PointStop → 포인트(원) 설정

생략하면 % 설정

예제 : 시스템 트레이딩의 특성상 보합장에는 취약한 면이 있으므로 시장이 추세를 가지지 못하고 횡보하여 이로 인한 손실의 우려가 있을 때 설정하는 기능

SetStopInactivity (2, 5, PercentStop)

매수 포지션 진입 후 5봉 안에 2% 이상의 가격상승이 없을시 청산

매수 포지션 진입 후 5봉 안에 2% 이상의 가격하락이 없을시 청산

최소가격변화 청산 해제

SetStopInactivity(0);

SetStopPosition

설명 : 포지션별 강제청산

함수 : SetStopPosition

예제 : SetStopPosition;

SetStopProfittarget(1,PointStop);

가령 현재 진입이 120.00에 2 계약 진입이 되어있고 목표수익이 1 Pt로 설정되어 있을

경우, 2계약의 수익합산이 목표가 1Pt이상이 되어야 하므로 현재가가 120.50에 도달하면

목표수익 조건에 따라 해당 진입이 청산된다.

※ 기본설정은 SetStopContract이다.

SetStopContract

설명 : 개별 거래 단위당 청산

함수 : SetStopContract

예제 : SetStopContract;

SetStopProfittarget(1,PointStop);

가령 현재 진입이 120.00에 2 계약 진입이 되어있고 목표수익이 1 Pt로 설정되어 있을 경우, 현재가가 121.00에 도달하면 목표수익 조건에 따라 해당 진입이 청산된다.

※ 기본설정은 SetStopContract이다.

2) 주문예약어

OnClose

설명 : 신호타입(기본설정)

봉완성시에 신호가 발생하고 완성봉 종가를 신호의 가격으로 사용하는 타입
봉완성 시점은 다음봉시가가 수신될 때.

예제 : buy("b",Onclose);

Onclose는 생략가능

buy("b");

buy();

AtMarket

설명 : 신호타입

봉완성시에 신호가 발생하고 완성봉의 다음봉시가를 신호의 가격으로 사용하는 타입
봉완성 시점은 다음봉시가가 수신될 때.

예제 : buy("b",atmarket);

AtStop

설명 : 신호타입

봉완성시에 가격을 지정해 셋팅하고

다음봉에서 **지정한 가격과 같거나 불리한 가격이 발생**하면 즉시 신호를 발생할때 사용

→ Buy, ExitShort인 경우에는 지정한 가격 이상에서 신호발생

→ Sell, ExitLong인 경우에는 지정한 가격 이하에서 신호발생

예제 : ※ 신호가격을 반드시 지정해 주어야 한다.

시초가 대비 1포인트 이상 가격 발생시 매수

Buy("b",atstop,Dayopen+1);

진입가 대비 1포인트 이상 가격 발생시 매도포지션 청산

ExitShort("sx",atstop, EntryPrice +1);

시초가 대비 1포인트 이하 가격 발생시 매도
Sell("S",atstop,Dayopen-1);
진입가 대비 1포인트 이하 가격 발생시 매수포지션 청산
ExitLong("Bx",atstop,EntryPrice-1);

AtLimit

설명 : 신호타입

봉완성시에 가격을 지정해 셋팅하고

다음봉에서 **지정한 가격과 같거나 유리한 가격이 발생**하면 즉시 신호를 발생할때 사용

→ Buy, ExitShort인 경우에는 지정한 가격 이하에서 신호발생

→ Sell, ExitLong인 경우에는 지정한 가격 이상에서 신호발생

예제 : ※ 신호가격을 지정해 주어야 한다.

시초가 대비 1포인트 이하 시세 발생시 매수

Buy("b",atlimit,Dayopen-1);

진입가 대비 1포인트 이하 시세 발생시 매도포지션 청산

ExitShort("sx",atlimit,EntryPrice+1);

시초가 대비 1포인트 이상 시세 발생시 매도

Sell("S",atlimit,Dayopen+1);

진입가 대비 1포인트 이상 시세 발생시 매수포지션 청산

ExitLong("bx",atlimit, EntryPrice+1);

PercentStop

설명 : 강제청산의 수치값을 %(Percent)로 설정

예제 : SetStoploss(5,PercentStop); → 5% 손실 시 청산

PointStop

설명 : 강제청산의 수치값을 Point(원)로 설정

예제 : SetStoploss(5,PointStop) → 5 point 손실 시 청산

3) 수학적함수

Avg, Avglist

설명 : 리스트에 있는 모든 항목을 평균한 값을 구함.

항목은 모두 수치 값이어야 함.

함수 : avg(num1,num2,num3,num4,...)

예제 : Avg(10, 20, 30) → 20 반환

Avg(Open, HH, ma(c,5), dayclose(1))

→ 지정한 4개의 항목 값을 평균한다.

Avg(H, L) → 중간값

Avg(C,O) → 몸통 중간값

Max, Maxlist

설명 : 리스트에 있는 모든 항목 중 최대값을 구함.

항목은 모두 수치 값이어야 함.

함수 : max(num1,num2,num3,num4,...)

예제 : max(10, 20, 30) → 30 반환

max(Open, HH, ma(c,5), dayclose(1))

→ 지정한 4개의 항목 값 중 최대값을 구함 .

max(C,O) → 몸통 상단

H-max(C,O) → 위꼬리

Maxlist2

설명 : 리스트에 있는 모든 항목 중 두번째 큰 값을 구함

항목은 모두 수치 값이어야 함.

함수 : Maxlist2(num1,num2,num3,num4,...)

예제 : Maxlist2(10, 20, 30) → 20 반환

Maxlist2(Open, HH, ma(c,5), dayclose(1))

→ 지정한 4개의 항목 값 중 두번째 높은 값을 구함 .

NthMaxList

설명 : 리스트에 포함된 값 중에서 N번째 큰 값을 구함.

항목은 모두 수치 값이어야 함

함수 : NthMaxList(N,num1,num2,num3,num4,...)

예제 : NthMaxList (3,10, 20, 30, 40) → 20 반환

NthMaxList (3, Open, HH, ma(c,5), dayclose(1))

→ 지정한 4개의 항목 값 중 세번째 높은 값을 구함 .

Min, MinList

설명 : 리스트에 있는 모든 항목 중 최소값을 구함.

항목은 모두 수치값이어야 함

함수 : min(num1,num2,num3,num4,...)

예제 : min(10, 20, 30) → 10 반환

`min(Open, HH, ma(c,5), dayclose(1))`

→ 지정한 4개의 항목 값 중 최소값을 구함 .

`min(C,O)` → 몸통 하단

`min(C,O)-L` → 아래꼬리

MinList2

설명 : 리스트에 있는 모든 항목 중 두번째 작은 값을 구함

항목은 모두 수치 값이어야 함.

함수 : `MinList2(num1,num2,num3,num4,...)`

예제 : `MinList2(10, 20, 30)` → 20 반환

`MinList2(Open, HH, ma(c,5), dayclose(1))`

→ 지정한 4개의 항목 값 중 두번째 작은 값을 구함 .

NthMinList

설명 : 리스트에 포함된 값 중에서 N번째 작은 값을 구함.

항목은 모두 수치 값이어야 함

함수 : `NthMinList(N,num1,num2,num3,num4,...)`

예제 : `NthMinList(3,10, 20, 30,40)` → 30 반환

`NthMinList(3, Open, HH, ma(c,5), dayclose(1))`

→ 지정한 4개의 항목 값 중 세번째 작은 값을 구함 .

SumList

설명 : 리스트에 포함된 모든 값을 누적인 값을 구함.

항목은 모두 수치 값이어야 함

함수 : `SumList(N,num1,num2,num3,num4,...)`

예제 : `SumList(10, 20, 30,40)` → 100 반환

`SumList(Open, HH, ma(c,5), dayclose(1))`

→ 지정한 4개의 항목 값을 모두 합산 한 값을 구함 .

Abs, AbsValue

설명 : 주어진 수치값의 절대값을 구함.

함수 : `abs(num)`, `absValue(num)`,

예제 : `abs(-1.5)`, `AbsValue(-1.5)` → 1.5 반환

`Abs(100)`, `absValue(100)` → 100 반환

`Abs(C-O)`, `absValue(C-O)` → 몸통길이

Mod

설명 : num1을 num2로 나눈 나머지를 구함.

num1과 num2는 모두 수치값이어야 함

연산자 %와 같음

함수 : mod(num1,num2)

예제 : mod(10,3) → 10/3 → 1 반환

10%3 → 10/3 → 1 나눔

mod(dayindex,5) → 봉을 5봉 간격으로 구분할 경우 사용

mod(TimeToMinutes(stime),30) → 봉을 30분 간격으로 구분할 경우 사용

Round

설명 : num을 지정한 자릿수 만큼 반올림한 값을 구함.

함수 : Round(num,자릿수)

예제 : Round(99.555,2) → 99.56

Round(99.555,1) → 99.60

Round(99.555,0) → 100.00

Ceiling

설명 : 소수점 값들을 반올림한 정수값을 구함,

함수 : Ceiling (num)

예제 : Ceiling(10.6) → 11

Ceiling(-5.4) → -5

Floor

설명 : 소수점 값들을 내림한 정수값을 구함,

함수 : Floor(num)

예제 : Floor(10.6) → 10

Floor(-5.4) → -6

FracPortion

설명 : 정수가 아닌 분수나 수의 소수 아래의 값을 구함.

함수 : FracPortion(num)

예제 : FracPortion(1.56) → 0.56

Int, IntPortion

설명 : 주어진 수의 정수부분을 구함

함수 : IntPortion(num) , Int(num)

예제 : IntPortion (1.56), Int(1.56) \rightarrow 1,

Neg

설명 : 주어진 num을 음수 값으로 반환

함수 : neg(num)

예제 : neg(15) \rightarrow -15

neg(-2) \rightarrow -2

Pos

설명 : 주어진 num을 양수 값으로 반환, abs와 같음

함수 : pos(num)

예제 : pos(15) \rightarrow 15

pos(-2) \rightarrow 2

Pie

설명 : 원주율

함수 : Pie

예제 : value*Pie \rightarrow 변수 value에 Pie를 곱함

Pow, Power

설명 : num을 N값만큼 제공한 값을 구함

연산자 ^와 같음

함수 : Pow(num,N), Power(num,N)

예제 : Pow(10,3) \rightarrow 10의 3승 \rightarrow 1000

$10^3 \rightarrow$ 10의 3승 \rightarrow 1000

SqRt, SquareRoot

설명 : num의 제곱근을 구함

함수 : SqRt(num), SquareRoot(num)

예제 : SquareRoot(25) \rightarrow 5

세제곱근과 같은 내용은 함수로 제공되지 않는다.

세제곱근을 구할 경우 $25^{(1/3)}$ 과 같은 수식으로 풀어서 그 값을 구해야 한다.

Square

설명 : num의 제곱한 값을 구함.

num^2로 작성한 것과 같음

함수 : Square(num)

예제 : Square(16) → 256

Random

설명 : 0과 num사이에서 임의의 값을 반환

함수 : Random(num)

예제 : random(10) → 4,5, 5,23과 같이 0과 10사이의 임의의 수가 리턴됨

Exp, ExpValue

설명 : 승수로 거듭제곱하는 자연로그의 밑은 e의 값을 반환

함수 : exp(num), expvalue(num),

예제 : exp(8) → 2980.95

Log

설명 : num의 자연로그 값

함수 : log(num)

예제 : log(2) → 0.693147

Log10

설명 : num의 상용로그 값

함수 : log10(num)

예제 : log10(2) → 0.301030

Cos, Cosine

설명 : 각도 num의 코사인 값을 구함

함수 : Cos(num), Cosine(num)

예제 : Cosine(8) → 0.990268(8도의 코사인 값)

CosH

설명 : 각도 num의 하이퍼브릭 코사인 값

함수 : CosH(num)

예제 : CosH(8) → 1.009764(8도의 하이퍼블릭 코사인 값)

Acos

설명 : num의 아크코사인 값

함수 : Acos(num)

예제 : $\text{Acos}(1) \rightarrow 0$ (라디안 1의 아크 코사인 값)

Sin, Sine

설명 : num의 각을 가지는 삼각비의 사인 값

함수 : $\sin(\text{num})$, $\text{sine}(\text{num})$

예제 : $\sin(8) \rightarrow 0.14$ (8의 사인 값)

SinH

설명 : 각도 num의 하이퍼블릭 사인

함수 : $\sinh(\text{num})$

예제 : $\sinh(30) \rightarrow 0.55$ (30도의 하이퍼블릭 사인 값)

Asin

설명 : num의 아크사인 값

함수 : $\text{Asin}(\text{num})$

예제 : $\text{Asin}(1) \rightarrow 1.570796$ (라디안 1의 아크사인 값)

Tan, Tangent

설명 : num의 각을 가지는 삼각비 중 탄젠트의 값

함수 : $\tan(\text{num})$, $\text{tangent}(\text{num})$

예제 : $\tan(8) \rightarrow 0.14$

TanH

설명 : 각도 num의 하이퍼블릭 탄젠트

함수 : $\tanh(\text{num})$

예제 : $\tanh(1) \rightarrow 0.017452$ (1도의 하이퍼블릭 탄젠트 값)

Atan ,ArcTangent

설명 : num의 아크탄젠트값

함수 : $\text{atan}(\text{num})$, $\text{arctangent}(\text{num})$

예제 : $\text{Atan}(1) \rightarrow 0.78$ (라디안 1의 아크 탄젠트값)

Cotangent

설명 : 각도 num의 코탄젠트(탄젠트 역수)

함수 : $\text{cotangent}(\text{num})$

예제 : $\text{cotangent}(30) \rightarrow 1.73$

4) 분석함수

DayClose

설명 : N일전 일봉 종가

일봉 데이터에서 직접 그 값을 가져 온다.

차트상 최초일+99일의 일봉 데이터가 추가로 제공된다.

함수 : DayClose(N)

예제 : DayClose(0) → 당일종가, (0) 생략가능

DayClose(1) → 전일종가

※ 주종목의 값만 리턴하므로 data2(dayclose(1))과 같이 사용해서는 안됨.

※ 참조종목은 CloseD함수를 이용 → data2(CloseD(1)) → data2의 전일종가

※ 일간데이터함수(DayOpen, DayHigh, DayLow, DayClose, DayVolume, DayOI)는
당일의 일봉은 아직 미완성이므로 당일값은 차트에서 계산하고 과거일(완성봉)의
경우에는 일봉데이터에서 직접 가져오게 됩니다.

그러므로 현재 차트에서 당일분의 데이터가 모두 없고 일부만 있다면 해당 일부
데이터에서만 당일값을 계산해서 값을 리턴하므로 이용에 유의하시기 바랍니다.

DayHigh(N)

설명 : N일전 일봉 고가

일봉 데이터에서 직접 그 값을 가져 온다.

차트상 최초일+99일의 일봉 데이터가 추가로 제공된다.

함수 : DayHigh(N)

예제 : DayHigh(0) → 당일고가, (0) 생략가능

DayHigh(1) → 전일고가

※ 주종목의 값만 리턴하므로 data2(dayhigh(1))와 같이 사용해서는 안됨.

※ 참조종목은 HighD함수를 이용 → data2(highD(1)) → data2의 전일고가

※ 일간데이터함수(DayOpen, DayHigh, DayLow, DayClose, DayVolume, DayOI)는
당일의 일봉은 아직 미완성이므로 당일값은 차트에서 계산하고 과거일(완성봉)의
경우에는 일봉데이터에서 직접 가져오게 됩니다.

그러므로 현재 차트에서 당일분의 데이터가 모두 없고 일부만 있다면 해당 일부
데이터에서만 당일값을 계산해서 값을 리턴하므로 이용에 유의하시기 바랍니다.

DayLow(N)

설명 : N일전 일봉 저가

일봉 데이터에서 직접 그 값을 가져 온다.

차트상 최초일+99일의 일봉 데이터가 추가로 제공된다.

함수 : DayLow(N)

예제 : DayLow(0) → 당일저가, (0) 생략가능

DayLow(1) → 전일저가

※ 주종목의 값만 리턴하므로 data2(daylow(1))와 같이 사용해서는 안됨.

※ 참조종목은 LowD함수를 이용 → data2(LowD(1)) → data2의 전일저가

※ 일간데이터함수(DayOpen, DayHigh, DayLow, DayClose, DayVolume, DayOI)는

당일의 일봉은 아직 미완성이므로 당일값은 차트에서 계산하고 과거일(완성봉)의 경우에는 일봉데이터에서 직접 가져오게 됩니다.

그러므로 현재 차트에서 당일분의 데이터가 모두 없고 일부만 있다면 해당 일부 데이터에서만 당일값을 계산해서 값을 리턴하므로 이용에 유의하시기 바랍니다.

DayOpen(N)

설명 : N일전 일봉 시가

일봉 데이터에서 직접 그 값을 가져 온다.

차트상 최초일+99일의 일봉 데이터가 추가로 제공된다.

함수 : DayOpen(N)

예제 : DayOpen(0) → 당일시가, (0) 생략가능

DayOpen(1) → 전일시가

※ 주종목의 값만 리턴하므로 data2(dayOpen(1))와 같이 사용해서는 안됨.

※ 참조종목은 OpenD함수를 이용 → data2(OpenD(1)) → data2의 전일고가

※ 일간데이터함수(DayOpen, DayHigh, DayLow, DayClose, DayVolume, DayOI)는

당일의 일봉은 아직 미완성이므로 당일값은 차트에서 계산하고 과거일(완성봉)의 경우에는 일봉데이터에서 직접 가져오게 됩니다.

그러므로 현재 차트에서 당일분의 데이터가 모두 없고 일부만 있다면 해당 일부 데이터에서만 당일값을 계산해서 값을 리턴하므로 이용에 유의하시기 바랍니다.

DayOi(N)

설명 : N일전 일봉 미결제약정

일봉 데이터에서 직접 그 값을 가져 온다.

차트상 최초일+99일의 일봉 데이터가 추가로 제공된다.

함수 : DayOi(N)

예제 : DayOi(0) → 당일미결제약정, (0) 생략가능

DayOi(1) → 전일미결제약정

※ 주종목의 값만 리턴하므로 참조종목은 따로 수식으로 계산하여 사용해야 한다.

```

#data2의 전일 미결제약정
var : preDayOI(0,data2);

if data2(date) != data2(date[1]) Then
5   preDayOI = data2(OI[1]);

plot1(predayOI);
10

```

※ 일간데이터함수(DayOpen, DayHigh, DayLow, DayClose, DayVolume, DayOI)는 당일의 일봉은 아직 미완성이므로 당일값은 차트에서 계산하고 과거일(완성봉)의 경우에는 일봉데이터에서 직접 가져오게 됩니다.

그러므로 현재 차트에서 당일분의 데이터가 모두 없고 일부만 있다면 해당 일부 데이터에서만 당일값을 계산해서 값을 리턴하므로 이용에 유의하시기 바랍니다.

DayVolume(N)

설명 : N일전 일봉 거래량

일봉 데이터에서 직접 그 값을 가져 온다.

차트상 최초일+99일의 일봉 데이터가 추가로 제공된다.

함수 : DayVolume(N)

예제 : DayVolume(0) → 당일 거래량, (0) 생략가능

DayVolume(1) → 전일 거래량

※ 주종목의 값만 리턴하므로 참조종목은 따로 수식으로 계산하여 사용해야 한다.

```

# data2의 전일누적 거래량
var : dayvol(0,data2),predayvol(0,data2);

if data2(date) != data2(date[1]) then{
5   dayvol = 0; #초기화
   predayvol = dayvol[1]; #전일치 할당
}

DayVol = dayvol+data2(V); #당일 거래량 누적
10
plot1(predayvol);

```

※ 일간데이터함수(DayOpen, DayHigh, DayLow, DayClose, DayVolume, DayOI)는 당일의 일봉은 아직 미완성이므로 당일값은 차트에서 계산하고 과거일(완성봉)의 경우에는 일봉데이터에서 직접 가져오게 됩니다.

그러므로 현재 차트에서 당일분의 데이터가 모두 없고 일부만 있다면 해당 일부 데이터에서만 당일값을 계산해서 값을 리턴하므로 이용에 유의하시기 바랍니다.

CrossUp

설명 : Value1이 Value2를 상향돌파

함수 : CrossUp(Value1,Value2)

예제 : crossup(c,ma(c,20)) → 종가가 20이동평균을 상향돌파

※ 수식으로 풀어 작성하면 아래와 같다.

Value1 > value2 and value1[1] <= value2[1]

C > ma(c,20) and C[1] <= ma(c,20)[1]

CrossDown

설명 : Value1이 Value2를 하향이탈

함수 : CrossDown(Value1,Value2)

예제 : CrossUp(c,ma(c,20)) → 종가가 20이동평균을 하향이탈

※ 수식으로 풀어 작성하면 아래와 같다.

Value1 < value2 and value1[1] >= value2[1]

C < ma(c,20) and C[1] >= ma(c,20)[1]

Highest

설명 : Length 기간 동안 value값 중 가장 큰 값

함수 : Highest(Value, Length)

예제 : highest(H,20) → 20봉 중 최고가

Highest(H,BarsSinceEntry) 진입이후 최고가

NthHghest(N, Value, Length)

설명 : Length 기간 동안 value값 중 N번째 큰 값

함수 : NthHghest(N, Value, Length)

예제 : NthHghest(2,H,20) → 20봉 중 두번째 큰 고가

NthHighestBar

설명 : Length 기간 동안 value값 중 N번째 큰 값의 위치

봉갯수로 값이 리턴되어 몇 봉전 임을 나타낸다.

함수 : NthHighestBar (N,Value, Length)

예제 : NthHghestbar(1,H,20) → 가령 10이라는 수치가 리턴 된다면

20개봉 동안 제일 큰 고가는 현재로부터 10봉 전에 발생했음을 뜻한다.

```
#20개봉중 최고가가 발생한 봉의 이동평균값
var1 = NthHighestBar(1,H,20);
var2 = ma(c,5)[var1];
```

5

Lowest

설명 : Length 기간 동안 value값 중 가장 작은 값

함수 : Lowest (Value, Length)

예제 : Lowest (L,20) → 20봉 중 최저가

Lowest(L,BarsSinceEntry) 진입이후 최저가

NthLowest

설명 : Length 기간 동안 value값 중 N번째 작은 값

함수 : NthLowest (N, Value, Length)

예제 : NthLowest (2,L,20) → 20봉 중 두번째 작은 저가

NthLowestBar

설명 : Length 기간 동안 value값 중 N번째 작은 값의 위치

봉갯수로 값이 리턴되어 몇 봉전 임을 나타낸다.

함수 : NthLowestBar(N,Value, Length)

예제 : NthLowestBar(1,L,20) → 가령 10이라는 수치가 리턴 된다면

20개봉 동안 제일 작은 저가는 현재로부터 10봉 전에 발생했음을 뜻한다.

```
#20개봉중 최저가가 발생한 봉의 이동평균값
var1 = NthLowestBar(1,L,20);
var2 = ma(c,5)[var1];
```

5

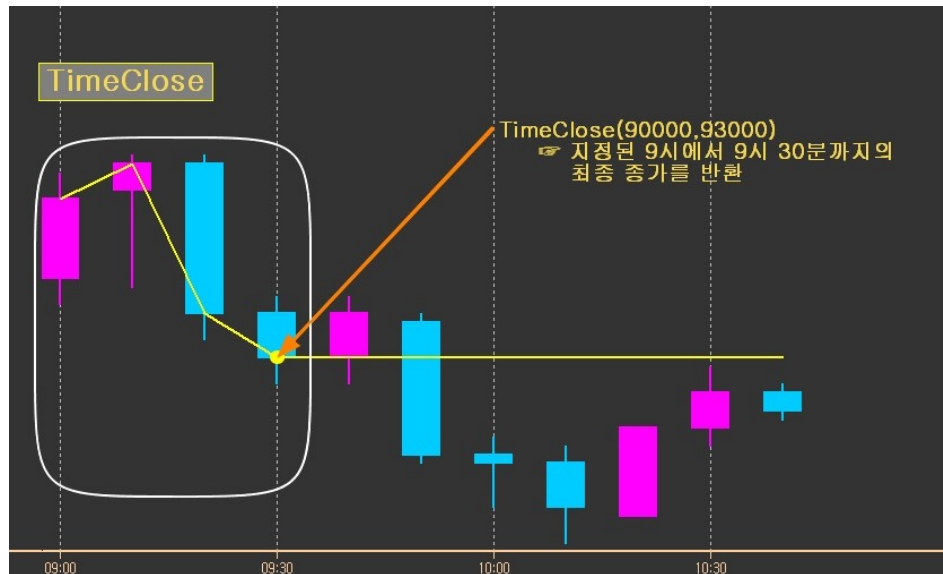
TimeClose

설명 : StartTime과 EndTime 으로 설정한 시간 범위 내의 최종 종가

시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeClose(StartTime,EndTime)

예제 : TimeClose(90000,93000) → 9시~9시 30분 봉까지의 종가



※ 모든 Time~함수들은 지정된 시간이 있는 봉을 모두 포함하므로 가령 5분봉 차트에서 TimeClose(90000,93000)로 지정하면 9시30분 봉까지 포함되므로 9시에서 9시 30분 이전으로 설정하고자 하면 TimeClose(90000,92959)로 지정해 주면 된다.

TimeCloseEx

설명 : 변동적으로 지정한 시간범위내의 최종 종가

시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeCloseEx(StartTime,EndTime)

예제 : TimeCloseEx(iff(Stime<110000,90000,100000),
iff(Stime<110000,95959,105959))

→ 봉 시작시간이 11시 이전일 경우 9시~ 9시59분 사이의 종가 반환.

→ 봉 시작시간이 11시 이후일 경우 10시~ 10시59분 사이의 종가 반환.



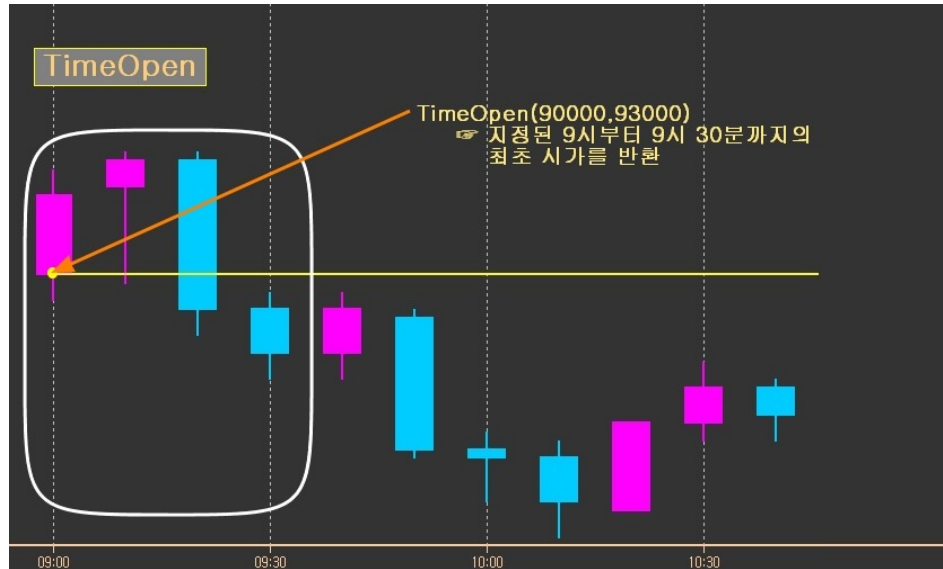
TimeOpen

설명 : StartTime과 EndTime 으로 설정한 시간 범위 내의 최초 시가

시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeOpen(StartTime,EndTime)

예제 : TimeOpen(90000,93000) → 9시에서 9시30분 사이의 최초 시가



TimeOpenEx

설명 : 변동적으로 지정한 시간 범위내의 최초 시가

시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeOpenEx(StartTime,EndTime)

예제 : TimeOpenEx(iff(Stime<110000,90000,100000),

iff(Stime<110000,95959,105959))

→ 봉시작시간이 11시 이전일 경우 9시~9시59분 사이의 최초 시가 반환

→ 봉시작시간이 11시 이후일 경우 10시~10시59분 사이의 최초 시가 반환.



TimeHigh

설명 : StartTime과 EndTime 으로 설정한 시간 범위 내의 최고가
 시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeHigh(StartTime,EndTime)

예제 : TimeHigh(90000,93000) → 9시에서 9시30분 사이의 최고가



TimeHighEx

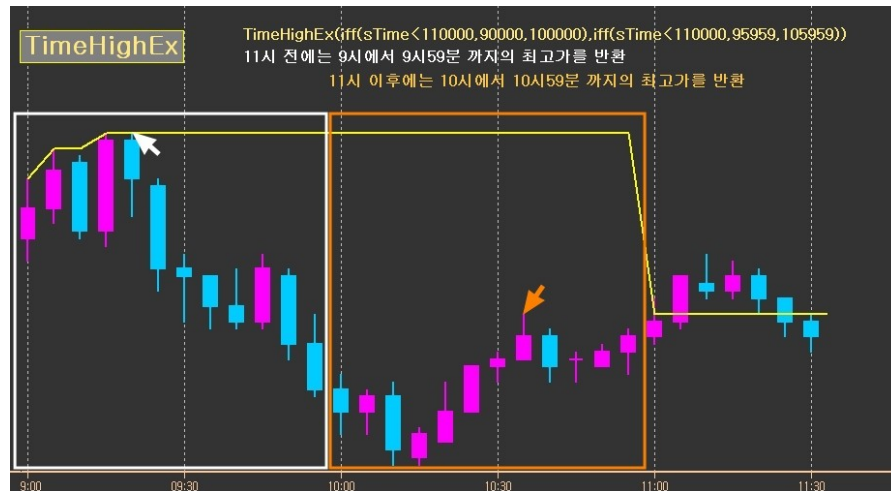
설명 : 변동적으로 지정한 시간 범위내의 최고가
 시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeHighEx(StartTime,EndTime)

예제 : TimeHighEx(iff(Stime<110000,90000,100000),
 iff(Stime<110000,95959,105959))

→ 봉 시작시간이 11시 이전일 경우 9시~9시59분 사이의 최고가 반환.

→ 봉 시작시간이 11시 이후일 경우 10시~ 10시59분 사이의 최고가 반환.

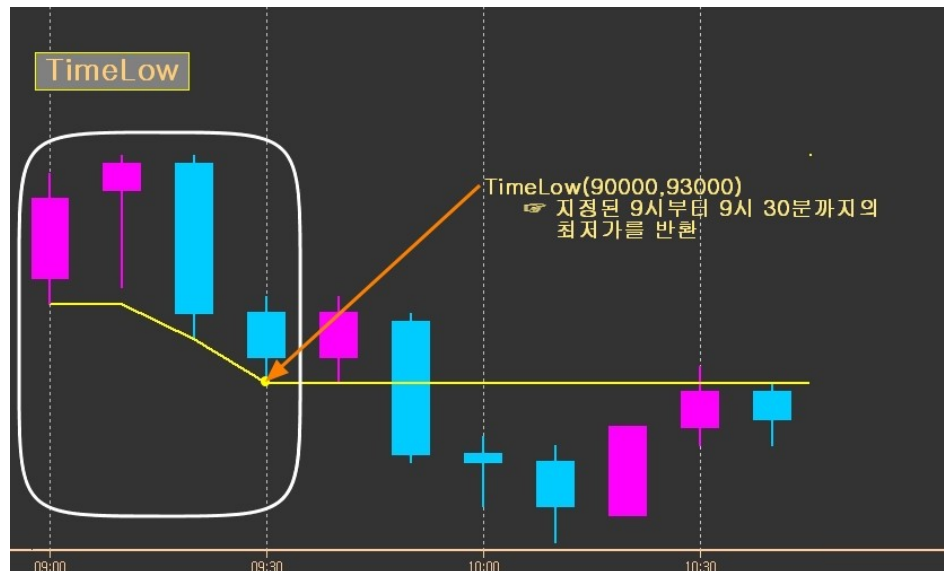


TimeLow

설명 : StartTime과 EndTime 으로 설정한 시간 범위 내의 최저가
시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeLow(StartTime,EndTime)

예제 : TimeLow(90000,93000) → 9시에서 9시30분의 최저가



TimeLowEx

설명 : 변동적으로 지정한 시간 범위내의 최저가

시간은 6자리로 설정(HHMMSS: HH시 MM분 SS초)

함수 : TimeLowEx(StartTime,EndTime)

예제 : TimeLowEx(iff(Stime<110000,90000,100000),
iff(Stime<110000,95959,105959))

→ 봉 시작시간이 11시 이전일 경우 9시에서 9시59분 사이 최저가 반환.

→ 봉 시작시간이 11시 이후일 경우 10시에서 10시59분 사이 최저가 반환.



Ema

설명 : 지수이동평균, Length를 평활계수로 하는 Value값 기준 지수이동평균

함수 : Ema(Value, Length)

예제 : ema(c,20) → 20기간 종가 지수이동평균

※ 지수이평 공식

지수 이동 평균값 = 금일 종가 * EP + 전일의 지수 이동평균 * (1-EP)

EP(평활계수) = 2/(기간 + 1)

최초 계산시 전일의 지수이동평균이 없으므로 전봉의 value가 전일의 지수이동평균값으로 사용됩니다..

Ma

설명 : 단순이동평균, Length기간 동안 Value값 기준 이동평균

함수 : Ma(Value, Length)

예제 : ma(c,20) → 20기간 종가 단순이동평균

※ 가중이동평균은 내장이 아닌 기본 사용자 함수로 제공되므로 사용자함수에서 설명이 있음.

Iff

설명 : 조건식의 만족여부에 따라 다른 값을 반환 하는 함수

참/거짓값은 같은 타입이어야 함.

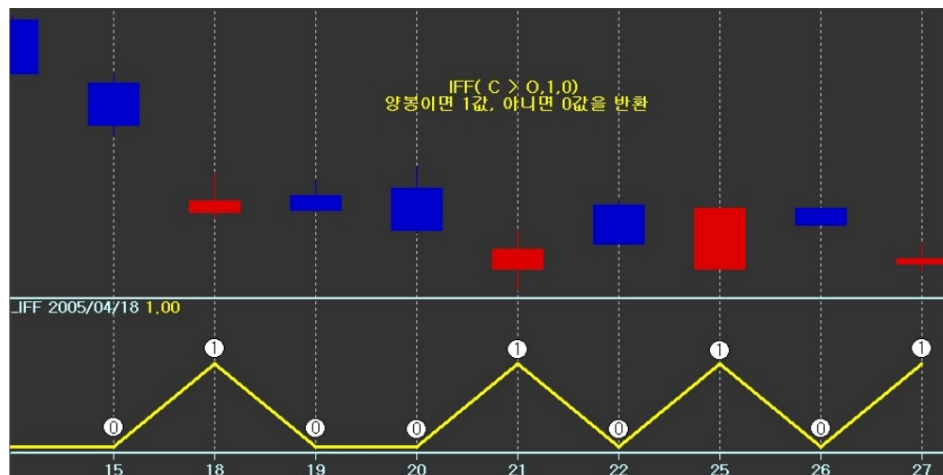
Iff(조건,1,0) → 0

Iff(조건,True,false) → 0

Iff(조건,1,false) → X

함수 : Iff(조건,참,거짓)

예제 : Iff(C > O,1,0) 해당 봉이 양봉이면 1, 아니면 0을 반환



NextBarOpen

설명 : 다음 봉 시가

※ 봉의 완성은 다음봉 시가가 수신되는 시점입니다.봉이 완성이 되면 다음봉 시작가격을 리턴합니다.

※ NextBarOpen을 매매식의 조건에서 이용할 경우 신호타입은 Atmarket, Atstop, Atlimit 만 사용하여야 한다. OnClose 타입을 사용하면 시뮬레이션과 실제매매의 결과가 다르게 나오므로 이 타입을 사용해서는 안된다.

함수 : NextbarOpen

예제 : 장 초반 시초가 수신 시 갭상승판단하여 즉시 매수

```
#시초가를 수신할 때 시초가가 전일종가보다 높으면 즉시 매수  
if stime == 151500 and NextBarOpen > DayClose Then  
    buy("b",AtMarket);
```

5



NextBarSdate

설명 : 다음 봉 시작시간

※ 봉의 완성은 다음봉 시가(**NextBarOpen**)가 수신되는 시점입니다.

다음봉 시가가 수신되어 봉이 완성될 때 다음봉 시작날짜를 리턴합니다.

(신호타입이 **Onclose**일때는 조건에 사용하지 않음)

함수 : NextbarSdate

NextBarStime

설명 : 다음 봉 시작시간

※ 봉의 완성은 다음봉 시가(**NextBarOpen**)가 수신되는 시점입니다.

다음봉 시가가 수신되어 봉이 완성될 때 다음봉 시작시간을 리턴합니다.

(신호타입이 **Onclose**일때는 조건에 사용하지 않음)

함수 : NextbarStime

Data1~Data99

설명 : 참조종목 함수

※ **Data1**은 주종목이므로 보통 생략한다.

※ 차트에 적용되어 있는 참조종목의 데이터를 사용하므로 수식 작성 시

참조로 하는 종목의 데이터 번호를 확인하고 사용해야 한다.

※ 참조종목은 차트 종목선택 버튼 중 오른쪽 버튼을 선택하여 지정가능 함

함수 : data2(value),data3(value),.....

예제 : data2(ma(c,20)) → 참조종목 data2의 20기간 이동평균

DayOfWeek

설명 : 지정한 날짜의 요일 값을 리턴하는 함수

날짜는 8자리로 표시해야 함 → YYYYMMDD(20100616)

일요일 : 0, 월요일 : 1, 화요일 : 2, 수요일 : 3, 목요일 : 4, 금요일 : 5, 토요일 : 6

함수 : DayOfWeek(날짜)

예제 : if DayOfWeek(sdate) < DayOfWeek(sdate[1]) then → 주가 변경되면

MinutesToTime

설명 : 주어진 분 단위를 시간으로 환산

함수 : MinutesToTime(분)

분은 0시를 기준으로 경과한 분단위로 입력한다.

예제 : MinutesToTime(540) → 90000 → 9시00분00초

MinutesToTime(550) → 91000 → 9시10분00초

TimeToMinutes

설명 : 주어진 시간을 0시를 기준으로 경과한 분단위로 계산.

함수 : TimeToMinutes(시간)

시간은 6자리로 입력해야 한다. → HHMMSS(102000)

예제 : 매수진입이후 30분 이상 경과했다면

if MarketPosition == 1 and

TimeToMinutes(stime) >= TimeToMinutes(EntryTime) + 30 Then

DateToJulian

설명 : 주어진 날짜를 Julian Date로 변경한다.

여기서 사용하는 Julian date의 기준은 1900년 1월 1일을 기준으로 한다.

즉 19000101은 0이 되며, 1일(하루) 증가 시 1이 증가한다

함수 : DateToJulian(날짜)

날짜는 8자리로 표시해야 함 → YYYYMMDD(20100616)

예제 : DateToJulian(20100531) → 40327

JulianToDate

설명 : 주어진 Julian Date로 날짜로 변경한다.

함수 : JulianToDate(JulianDate)

예제 : JulianToDate(40327) → 20100531

CurrentDate

설명 : 컴퓨터의 날짜를 가져옴

함수 : CurrentDate

예제 : 주로 지표를 당일에만 표시하고자 할 때 사용.

```
#당일에만 전일고가를 지표로 표시
if CurrentDate == sdate Then
    plot1(dayhigh(1));
```

5

CurrentTime

설명 : 컴퓨터의 시간을 가져옴

함수 : CurrentTime

IsNan(Value)

설명 : value의 값이 N/A이면 True 아니면 false를 리턴

모든 봉에서 계산 값이 도출되는 계산식이 아닐 경우 현재봉에서의 계산 여부를 판단할 때 사용한다.

함수 : IsNan(Value)

예제 : if isnan(value) == false then var1 = C;

→ value에 값이 할당되면 var1에 종가 저장

Nan

설명 : 매개변수에 N/A를 저장 함

함수 : nan

예제 : value = nan;

LastBarOnChart

설명 : 차트 마지막 봉임을 확인할 때 사용

차트상 마지막 봉에서만 어떤 행위를 하고자 할 때 사용

현재봉이 차트 마지막봉이 맞으면 1 아니면 0을 리턴

함수 : LastBarOnChart

예제 :

```
#검색식
#차트 마지막봉이 양봉이고 직전 10개봉 평균몸통길이의
#2배이상일 경우 고가에 점으로 표시
5 if LastBarOnChart == 1 and C > 0 and
    abs(C-0) > ma(abs(C-0),10)[1]*2 Then
    plot1(H);
```

LastCalcMMTime

설명 : 차트 마지막봉 시간을 밤 12시이후의 분단위로 계산하여 리턴

함수 : LastCalcMMTime

InStr

설명 : String1의 문자열 중에서 string2의 문자가 시작되는 위치 값을 리턴

String1에 string2로 지정한 문자가 없으면 0

String1에 string2로 지정한 문자가 다수이면 첫번째의 위치

함수 : InStr("string1","String2")

예제 : InStr("ABCDEFGF","C") → 3

LeftStr

설명 : String 문자열 내에서 가장 왼쪽부터 Num으로 지정한 부분까지의 문자열

함수 : LeftStr("String", Num)

예제 : LeftStr("ABCDEFGF ", 3) → ABC

RightStr

설명 : String 문자열 내에서 가장 오른쪽부터 Num으로 지정한 부분까지의 문자열

함수 : RightStr("String", Num)

예제 : RightStr("ABCDEFGF ", 3) → EFG

LowerStr

설명 : 지정한 string 문자열을 소문자로 리턴

함수 : LowerStr("String")

예제 : LowerStr("AbCdEfG") → abcdefg

UpperStr

설명 : 지정한 string 문자열을 대문자로 리턴

함수 : UpperStr("String")

예제 : UpperStr("AbCdEfG") → ABCDEFG

MidStr

설명 : String 문자열 내에서 Num1로 지정한 부분부터

num2 만큼 세어 문자열을 리턴

함수 : MidStr("String",Num1,Num2)

예제 : MidStr("ABCDEFGF" ,2, 3) → BCD

StrToNum

설명 : 문자열을 숫자로 리턴

함수 : StrToNum("String")

예제 : StrToNum("1234") → 1234

NumToStr

설명 : 숫자를 지정한 소수점 N자리까지 문자열로 리턴

함수 : NumToStr(Figure,N)

예제 : NumToStr(1234.123 , 2)→ "1234.12"

EL_DateStr

설명 : YYYYMMDD형식의 문자열로 날짜를 리턴

함수 : EL_DateStr(DD,MM,YYYY)

예제 : EL_DateStr(16,6,2010) → "20100616"

StrLen

설명 : string으로 지정한 문자열의 길이를 리턴

함수 : StrLen("String")

예제 : StrLen("ABCDEFGH") → 7

MaxBarsBack

설명 : 주식 계산에 필요한 봉갯수를 리턴

함수 : MaxBarsBack

예제 : ma(c,20) → MaxBarsBack 은 20

Spaces(Num)

설명 : num으로 지정된 수만큼 빈공간을 만든다.

messagelog나 Print함수 사용시 값과 값 사이 간격을 넓히고자 할 때 사용

함수 : Spaces(Num)

예제 : MessageLog("%.f%s%.2f%s%.2f",C,Spaces(5),H,Spaces(5),L);

NewLine

설명 : 새로운 행을 삽입한다.

messagelog나 Print함수 새로운 행으로 출력하고자 할 때 사용

함수 : NewLine

예제 : print("BB.txt", "%.2f%s%.2f%s%.2f",C,NewLine,H,NewLine,L);

NoValue

설명 : 화면표시 제거

함수 : NoValue

예제 : plot1(NoValue);

Extremes

설명 : 극단값과 위치

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : Extremes(Price, Length, HiLo, oExtremeVal, oExtremePosRaw)

※ Pricey : 기준값

※ Length : 기간

※ HiLo : 최대값 1, 최소값 -1

※ oExtremeVal: 극단값 리턴, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

※ oExtremePosRaw: 극단값 위치(방번호), NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : var1 = Extremes(H, 5, 1, Hv, Hx);

#최근 5개봉의 최고값을 Hv, 위치를 Hx에 저장 저장하고 수행결과를 var1에 저장

HarmonicMean

설명 : 조화평균

함수 : HarmonicMean(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : var1 = HarmonicMean(H, 5); #최근 5개봉의 조화평균을 var1에 저장

Mode

설명 : 최빈값

함수 : Mode(Price,Length, hiLo)

※ Price : 기준값

※ Length : 기간

※ HiLo : 동일빈도 발생시 리턴값, 큰값 : 1, 작은값 : -1

예제 : var1 = Mode(H, 5, 1); #5개봉 고가 중 최빈값(동일빈도이면 큰값)을 var1에 저장

NthExtremes

설명 : Nth번째 극단값과 위치

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : NthExtremes(Price, Length, Nth, HiLo, oExtremeVal, oExtremePosRaw)

※ Price : 기준값

※ Length : 기간

※ Nth : 순위

※ HiLo : 최대값 1, 최소값 -1

※ oExtremeVal : Nth번째 극단값, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

※ oExtremePosRaw : Nth번째 극단값 위치(방번호), NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : var1 = NthExtremes(C, 5, 2, 1, Hv, Hx);

#5개봉의 증가 중 2번째 최고값을 Hv, 위치를 Hx에 저장하고 수행결과를 var1에 저장

SummationRec

설명 : 역수합

함수 : SummationRec(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : var1 = SummationRec(C, 5); #5개봉 증가의 역수합을 var1에 저장

SummationSqr

설명 : 제곱합

함수 : SummationSqr(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : var1 = SummationSqr(C, 5); #5개봉 제곱합을 var1에 저장

TrimMean

설명 : 극단치 제외 평균

함수 : TrimMean(Price, Length, TrimPct)

※ Price : 기준값

※ Length : 기간

※ TrimPct: 제외 백분율 (10%이면 0.1)

예제 : var1 = TrimMean(C, 5, 0.2); # 극단값 20%를 제외한 5개봉 평균을 var1에 저장

Variance(Price, Size, DataType)

설명 : 모집단의 분산 혹은 표본집단의 분산

함수 : Variance(PriceArray, Length, DataType)

※ Price : 기준값

※ Length : 기간

※ DataType: 모집단 : 1, 표본 : -1

예제 : var1 = Variance(C, 5, 1); #5개봉 모집단의 분산을 var1에 저장

Covar

설명 : 공분산

함수 : Covar(Indep, Dep, Length)

※ Indep : 독립변수

※ Dep : 종속변수

※ Length : 기간

예제 : var1 = Covar(V1, V2, 5); #2개 변수의 5개봉 공분산을 var1에 저장

AvgDeviation

설명 : 편차평균

함수 : AvgDeviation(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : var1 = AvgDeviation(C, 5); #5개봉 편차평균을 var1에 저장

DevSqr

설명 : 편차 제곱합

함수 : DevSqr(Price, Length)

※ Price : 기준값

※ Length: 기간

예제 : var1 = DevSqr(C, 5); #5개봉 편차 제곱합을 var1에 저장

StdError

설명 : 표준오차

함수 : StdError(Price, Length)

※ Price : 기준값

※ Length: 기간

예제 : var1 = StdError(C ,5); #5개봉 표준오차를 var1에 저장

StdError2

설명 : 두 변수의 표준오차

함수 : StdError2(Indep, Dep, Length)

※ Indep : 독립변수

※ Dep : 종속변수

※ Length : 기간

예제 : var1 = StdError2(V1,V2, 5); #두 변수의 5개봉 표준오차를 var1에 저장

CoefficientR

설명 : 피어슨 상관계수R

함수 : CoefficientR(IndDep,Dep, Length)

※ IndDep : 독립변수

※ Dep : 종속변수

※ Length : 기간

예제 : var1 = CoefficientR(V1,V2,5); #두 변수의 5개봉 피어슨 상관계수R을 var1에 저장

RSquare

설명 : 피어슨 상관계수R의 제곱

함수 : RSquare(Indep, Dep, Length)

※ Indep : 독립변수

※ Dep : 종속변수

※ Length : 기간

예제 : var1 = RSquare(V1, V2, 5); #두 변수의 5개봉 피어슨 상관계수R 제곱을 var1에 저장

Skew

설명 : 왜도

함수 : Skew(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : var1 = Skew(C, 5); #5개봉 왜도를 var1에 저장

LinReg

설명 : 선형회귀선의 기울기, 각도, 절편, 예측값

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : LinReg(Price, Length, TgtPos, oLRSlope, oLRAngle,
oLRIntercept, oLRValueRaw)

- ※ Price : 기준값
- ※ Length : 기간
- ※ TgtPos : 예측지점, 미래는 음수 과거는 양수
- ※ oLRSlope : 선형회귀선의 기울기, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRAngle : 선형회귀선의 각도, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRIntercept : 선형회귀선의 절편, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRValueRaw : 선형회귀선의 예측값, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : var1 = LinReg(C, 5,-2, Slope, Angle, Intercept, Value);
#5개봉의 선형회귀선의 기울기는 Slope, 각도는 Angle, 편차는 Intercept,
예측값은 Value에 저장하고 수행결과를 var1에 저장

LinReg2

설명 : 두 변수의 선형회귀선의 기울기, 각도, 절편, 예측값
※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : LinReg2(Indep, Dep, Length, TgtPos, oLRSlope,
oLRAngle, oLRIntercept, oLRValueRaw)

- ※ Indep : 독립변수
- ※ Dep : 종속변수
- ※ Length : 기간
- ※ TgtPos : 예측지점, 미래는 음수 과거는 양수
- ※ oLRSlope : 선형회귀선의 기울기, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRAngle : 선형회귀선의 각도, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRIntercept : 선형회귀선의 절편, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRValueRaw : 선형회귀선의 값, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : var1 = LinReg2(V1, V2, 5,-2, Slope, Angle, Intercept, Value);
#5개봉 두 변수의 선형회귀선의 기울기는 Slope, 각도는 Angle, 편차는 Intercept,

예측값은 Value에 저장하고 수행결과를 var1에 저장

LinRegForecast

설명 : 선형회귀를 기반으로 주어진 X값에 대한 예측값

함수 : LinRegForecast(Price, Size, TgtPos)

※ Price : 기준값

※ Length : 기간

※ TgtPos : 예측지점, 미래는 음수 과거는 양수

예제 : var1 = LinRegForecast(C, 5, -5); #5개봉의 선형회귀선의 예측값을 var1에 저장

LinRegIntercept

설명 : 선형회귀선의 절편

함수 : LinRegIntercept(Price, Length)

※ Price : 기준값

※ Length: 기간

예제 : var1 = LinRegIntercept(C, 5); #5개봉 선형회귀선의 절편을 var1에 저장

LinRegSlope

설명 : 선형회귀선의 기울기

함수 : LinRegSlope(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : var1 = LinRegSlope(C, 5); #5개봉 선형회귀선의 기울기를 var1에 저장

NormalCumDensity(Price, Size)

설명 : 정규누적분포

함수 : NormalCumDensity(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : var1 = NormalCumDensity(C, 5); #5개봉 누적정규분포를 var1에 저장

NormalDensity(Price, Length)

설명 : 정규분포

함수 : NormalDensity(Price, Length)

※ Price : 기준값

※ Length : 기간

예제 : `var1 = NormalDensity(C, 5);` #5개봉 정규분포를 var1에 저장

Standardize(Price, Length, NumDevs)

설명 : 표준편차 및 평균에 따라 표준화된 값을 반환

※ 1차원 배열만 가능.

※ 계산은 항상 0번방부터 시작

함수 : `Standardize(Price, Length, NumDevs)`

※ Price : 기준값

※ Length : 기간

※ NumDevs : 표준편차 배수

예제 : `var1 = Standardize(C, 5, 2);` #5개봉 표준화값을 var1에 저장

InsertArray

설명 : 배열에 값 삽입.

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : `InsertArray(PriceArray, location, value)`

※ PriceArray : 대상배열

※ location: 삽입위치

※ value : 삽입값, 일반변수 or 상수 or 2차원 이하의 배열

삽입할 값은 대상배열변수보다 낮은 차원만 가능

예제 : 예1) 1차원 배열에 값 삽입

대상배열이 1차원이면 삽입할 값은 상수나 일반변수만 가능.

1차원 배열은 행이 하나이므로 삽입위치는 열번호가 되고

지정한 열에 값이 삽입되고 기존값은 다음열로 이동됨.

Array : `Arr1[5](0);`

대상배열	0	1	2	3	4 열
Arr1	91	78	23	40	64

`InsertArray(Arr1, 1, 10);`

삽입 후 대상배열

	0	1	2	3	4 열
Arr1	91	10	78	23	40

예2) 2차원 배열에 1차원 배열 삽입

대상배열이 2차원이면 행이 여러개 이므로 삽입위치는 행번호로 지정되고

대상배열의 지정한 행의 0열부터 값이 삽입되고 기존값은 다음행으로 이동

Array : Arr1[5](0), Arr2[4,5](0);

삽입할 배열
Arr1

	0	1	2	3	4 열
	91	78	23	40	64

대상배열
Arr2

	0	1	2	3	4 열
0	69	27	41	99	89
1	9	13	90	15	22
2	96	65	20	13	44
3	71	90	85	14	52

행

InsertArray(Arr2, 1, Arr1);

삽입 후 대상배열
Arr2

	0	1	2	3	4 열
0	69	27	41	99	89
1	91	78	23	40	64
2	9	13	90	15	22
3	96	65	20	13	44

행

삽입할 1차원 배열의 열갯수가 대상배열의 열갯수보다 크면
해당 부분은 버리고 삽입을 하게 됨.

Array : Arr1[7](0), Arr2[4,5](0);

삽입할 배열
Arr1

	0	1	2	3	4	5	6 열
	91	78	23	40	64	25	12

대상배열
Arr2

	0	1	2	3	4 열
0	69	27	41	99	89
1	9	13	90	15	22
2	96	65	20	13	44
3	71	90	85	14	52

행

InsertArray(Arr2, 1, Arr1);

삽입 후 대상배열
Arr2

	0	1	2	3	4 열
0	69	27	41	99	89
1	91	78	23	40	64
2	9	13	90	15	22
3	96	65	20	13	44

행

삽입할 1차원 배열의 열갯수가 대상배열의 열갯수보다 작으면

삽입할 1차원 배열의 마지막 열까지만 삽입되고
삽입된 열까지만 기준으로 다음행으로 이동됨.

Array : Arr1[3](0), Arr2[4,5](0);

삽입할 배열
Arr1

	0	1	2 열
Arr1	91	78	23

대상배열
Arr2

		0	1	2	3	4 열
0	69	27	41	99	89	
1	9	13	90	15	22	
2	96	65	20	13	44	
3	71	90	85	14	52	

행

InsertArray(Arr2, 1, Arr1);

삽입 후 대상배열
Arr2

		0	1	2	3	4 열
0	69	27	41	99	89	
1	91	78	23	15	22	
2	9	13	90	13	44	
3	96	65	20	14	52	

행

예3) 2차원 배열에 상수 삽입

대상배열이 2차원이면 행이 여러개 이므로 삽입위치는 행번호로 지정되고
지정한 행의 0열부터 마지막열까지 지정된 값이 삽입되고 기존값은 다음행으로 이동

Array : Arr2[4,5](0);

대상배열
Arr2

		0	1	2	3	4 열
0	69	27	41	99	89	
1	9	13	90	15	22	
2	96	65	20	13	44	
3	71	90	85	14	52	

행

InsertArray(Arr2, 1, 0);

삽입 후 대상배열
Arr2

		0	1	2	3	4 열
0	69	27	41	99	89	
1	0	0	0	0	0	
2	9	13	90	15	22	
3	96	65	20	13	44	

행

예4) 3차원 배열에 2차원 배열 삽입

대상배열이 3차원이면 면이 여러개 이므로 삽입위치는 면번호로 지정되고

지정한 면의 0행 0열부터 값이 삽입되고 기존값은 다음면으로 이동

Array : Arr2[4,5](0), Arr3[3,4,5](0);

삽입할배열

Arr2

	0	1	2	3	4 열
0	69	27	41	99	89
1	9	13	90	15	22
2	96	65	20	13	44
3	71	90	85	14	52

행

대상배열

Arr3

	0면	1	2	3	4 열
0	64	1	85	66	44
1	6	20	25	83	95
2	57	30	50	92	3
3	97	68	80	67	57

행

	0	1	2	3	4 열
0	44	34	5	45	83
1	62	29	62	1	78
2	65	75	42	0	59
3	95	64	95	74	96

행

	0면	1	2	3	4 열
0	53	2	22	14	30
1	55	59	50	97	37
2	33	38	47	17	39
3	15	12	53	64	26

행

InsertArray(Arr2, 1, 0);

삽입 후 대상배열

Arr3

	0면	1	2	3	4 열
0	64	1	85	66	44
1	6	20	25	83	95
2	57	30	50	92	3
3	97	68	80	67	57

행

	0	1	2	3	4 열
0	69	27	41	99	89
1	9	13	90	15	22
2	96	65	20	13	44
3	71	90	85	14	52

행

	0면	1	2	3	4 열
0	44	34	5	45	83
1	62	29	62	1	78
2	65	75	42	0	59
3	95	64	95	74	96

행

삽입할 2차원 배열의 행열갯수가 대상배열의 행열갯수보다 크면
해당 부분은 버리고 삽입을 하게 됨.

Array : Arr2[5,6](0), Arr3[3,4,5](0);

삽입할배열

Arr2

	0	1	2	3	4	5
0	69	27	41	99	89	32
1	9	13	90	15	22	28
2	96	65	20	13	44	76
3	71	90	85	14	52	14
4	67	1	49	32	2	19

행

대상배열

Arr3

	0면	1면	2면
0	64	44	53
1	6	62	55
2	57	65	33
3	97	95	15

행

	0	1	2	3	4	열
0	1	34	5	45	83	
1	20	29	62	1	78	
2	30	75	42	0	59	
3	68	64	95	74	96	

행

	0	1	2	3	4	열
0	2	22	14	30		
1	59	50	97	37		
2	38	47	17	39		
3	12	53	64	26		

행

InsertArray(Arr2, 1, 0);

삽입 후 대상배열

Arr3

	0면	1면	2면
0	64	44	53
1	6	62	55
2	57	65	33
3	97	95	15

행

	0	1	2	3	4	열
0	69	27	41	99	89	
1	9	13	90	15	22	
2	96	65	20	13	44	
3	71	90	85	14	52	

행

	0	1	2	3	4	열
0	44	34	5	45	83	
1	62	29	62	1	78	
2	65	75	42	0	59	
3	95	64	95	74	96	

행

삽입할 2차원 배열의 행열갯수가 대상배열의 행열갯수보다 작으면

삽입할 2차원 배열의 마지막 행열번호까지만 삽입되고

삽입된 마지막 행열번호 기준으로 다음면으로 이동됨.

Array : Arr2[3,4](0), Arr3[3,4,5](0);

삽입할배열

Arr2

	0	1	2	3 열
0	69	27	41	99
1	9	13	90	15
2	96	65	20	13

행

대상배열

Arr3

0면

	0	1	2	3	4 열
0	64	1	85	66	44
1	6	20	25	83	95
2	57	30	50	92	3
3	97	68	80	67	57

행

1면

	0	1	2	3	4 열
0	44	34	5	45	83
1	62	29	62	1	78
2	65	75	42	0	59
3	95	64	95	74	96

행

2면

	0	1	2	3	4 열
0	53	2	22	14	30
1	55	59	50	97	37
2	33	38	47	17	39
3	15	12	53	64	26

행

InsertArray(Arr2, 1, 0);

삽입 후 대상배열

Arr3

0면

	0	1	2	3	4 열
0	64	1	85	66	44
1	6	20	25	83	95
2	57	30	50	92	3
3	97	68	80	67	57

행

1면

	0	1	2	3	4 열
0	69	27	41	99	83
1	9	13	90	15	78
2	96	65	20	13	59
3	95	64	95	74	96

행

2면

	0	1	2	3	4 열
0	44	34	5	45	83
1	62	29	62	1	78
2	65	75	42	0	59
3	95	64	95	74	96

행

예5) 3차원 배열에 상수 삽입

대상배열이 3차원이면 면이 여러개 이므로 삽입위치는 면번호로 지정되고

지정한 면의 0행 0열부터 마지막 행열까지 모두 지정된 값이 삽입되고

기존값은 다음면으로 이동

Array : Arr3[3,4,5](0);

대상배열

Arr3

0면

	0	1	2	3	4 열
0	64	1	85	66	44
1	6	20	25	83	95
2	57	30	50	92	3
3	97	68	80	67	57

행

1면

	0	1	2	3	4 열
0	44	34	5	45	83
1	62	29	62	1	78
2	65	75	42	0	59
3	95	64	95	74	96

행

2면

	0	1	2	3	4 열
0	53	2	22	14	30
1	55	59	50	97	37
2	33	38	47	17	39
3	15	12	53	64	26

행

InsertArray(Arr3, 1, 0);

삽입 후 대상배열

Arr3

0면

	0	1	2	3	4 열
0	64	1	85	66	44
1	6	20	25	83	95
2	57	30	50	92	3
3	97	68	80	67	57

행

1면

	0	1	2	3	4 열
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0

행

2면

	0	1	2	3	4 열
0	44	34	5	45	83
1	62	29	62	1	78
2	65	75	42	0	59
3	95	64	95	74	96

행

AverageArray

설명 : 배열의 평균

함수 : AverageArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = averageArray(Arr, 5); #0~4번방의 평균

ExtremesArray

설명 : 배열의 극단값과 위치

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : ExtremesArray(PriceArray, Size, HiLo, oExtremeVal, oExtremePosRaw)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ HiLo : 최대값 1, 최소값 -1

※ oExtremeVal: 극단값 리턴, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

※ oExtremePosRaw: 극단값 위치(방번호), NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = ExtremesArray(Arr, 5, 1, Hv, Hx);

#0~4번방의 최고값을 Hv, 위치를 Hx에 저장 저장하고 수행결과를 var1에 저장

HarmonicMeanArray

설명 : 배열의 조화평균

함수 : HarmonicMeanArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = HarmonicMeanArray(Arr, 5); #0~4번방의 조화평균

HighestArray

설명 : 배열의 최고가

함수 : HighestArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = HighestArray(Arr, 5); #0~4번방의 최고가

LowestArray

설명 : 배열의 최저가

함수 : LowestArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = LowestArray(Arr, 5); #0~4번방의 최저가

ModeArray

설명 : 배열의 최빈값

함수 : ModeArray(PriceArray, Size, HiLo)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ HiLo : 동일빈도 발생시 리턴값, 큰값 : 1, 작은값 : -1

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = ModeArray(Arr, 5, 1); #0~4번방의 최빈값(동일빈도이면 큰값)을 var1에 저장

NthExtremesArray

설명 : 배열의 Nth번째 극단값과 위치

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : NthExtremesArray(PriceArray, Size, Nth, HiLo, oExtremeVal, oExtremePosRaw)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ Nth : 순위

※ HiLo : 최대값 1, 최소값 -1

※ oExtremeVal : Nth번째 극단값, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

※ oExtremePosRaw : Nth번째 극단값 위치(방번호), NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = NthExtremesArray(Arr, 5, 2, 1, Hv, Hx);
#0~4번방의 2번째 최고값을 Hv, 위치를 Hx에 저장하고 수행결과를 var1에 저장

NthHighestArray

설명 : 배열의 Nth번째 최고값과 위치

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : NthHighestArray(PriceArray, Size, Nth, oValue, oIndex)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ Nth : 순위

※ oValue : 최대값 리턴, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

※ oIndex : 최대값 위치(방번호), NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = NthHighestArray(Arr, 5, 2, Hv, Hidx);
#0~4번방의 2번째 최고값을 Hv, 위치를 Hidx에 저장하고 수행결과를 var1에 저장

NthLowestArray

설명 : 배열의 Nth번째 최저값과 위치

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : NthLowestArray(PriceArray, Size, Nth, oValue, oIndex)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ Nth : 순위

※ oValue : Nth번째 최저값 리턴, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

※ oIndex : Nth번째 최저값 위치(방번호), NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = NthLowestArray(Arr, 5, 2, Lv, Lidx);

#0~4번방의 2번째 최고값을 Lv, 위치를 Lidx에 저장하고 수행결과를 var1에 저장

SummationArray

설명 : 배열의 합계

함수 : SummationArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = SummationArray(Arr, 5); #0~4번방의 합계를 var1에 저장

SummationRecArray

설명 : 배열의 역수값의 합계

함수 : SummationRecArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = SummationRecArray(Arr, 5); #0~4번방의 역수값 합계를 var1에 저장

SummationSqrArray

설명 : 배열의 제곱합

함수 : SummationSqrArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = SummationSqrArray(Arr, 5); #0~4번방의 제곱합을 var1에 저장

TrimMeanArray

설명 : 배열의 극단치 제외 평균

함수 : TrimMeanArray(PriceArray, Size, TrimPct)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ TrimPct: 제외 백분율 (10%이면 0.1)

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = TrimMeanArray(Arr, 5, 0.2);
#0~4번방의 가격 중 극단값 20%를 제외한 평균을 var1에 저장

VarianceArray

설명 : 배열의 모집단의 분산 혹은 표본집단의 분산

함수 : VarianceArray(PriceArray, Size, DataType)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ DataType: 모집단 : 1, 표본 : -1

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = VarianceArray(Arr,5,1); #0~4번방의 모집단의 분산을 var1에 저장

CovarArray

설명 : 두 배열의 공분산

함수 : CovarArray(IndepArray,DepArray,Size)

※ IndepArray : 독립배열, 1차원 수치형 배열만 가능

※ DepArray : 종속배열, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr1[20](0),Arr2[20](0);
{Array 값 할당 로직}
var1 = CovarArray(Arr1, Arr2, 5); #2개 배열의 0~4번방의 공분산을 var1에 저장

AvgDeviationArray

설명 : 배열의 편차평균

함수 : AvgDeviationArray(PriceArray,Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = AvgDeviationArray(Arr ,5); #0~4번방의 편차평균을 var1에 저장

DevSqrArray

설명 : 배열의 편차 제공함

함수 : DevSqrArray(PriceArray,Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = DevSqrArray(Arr ,5); #0~4번방의 편차제공함을 var1에 저장

StdErrorArray

설명 : 배열의 표준오차

함수 : StdErrorArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = StdErrorArray(Arr ,5); #0~4번방의 표준오차를 var1에 저장

StdErrorArray2

설명 : 두 배열의 표준오차

함수 : StdErrorArray2(IndepArray, DepArray, Size)

※ IndepArray : 독립배열의 이름, 1차원 수치형 배열만 가능

※ DepArray : 종속배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr1[20](0), Arr2[20](0);

{Array 값 할당 로직}

var1 = StdErrorArray2(Arr1, Arr2, 5); #두 배열의 0~4번방의 표준오차를 var1에 저장

StandardDevArray

설명 : 배열의 모집단의 표준편차 혹은 표본집단의 표준편차 값을 반환

함수 : StandardDevArray(PriceArray, Size, DataType)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ DataType: 모집단 : 1, 표본 : 2

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = StandardDevArray(Arr,5,1); #0~4번방의 모집단 표준편차를 var1에 저장

CoefficientRArray

설명 : 두 배열의 피어슨 상관계수R

함수 : CoefficientRArray(IndDepArray,DepArray,Size)

※ IndDepArray : 독립배열의 이름, 1차원 수치형 배열만 가능

※ DepArray : 종속배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr1[20](0), Arr2[20](0);

{Array 값 할당 로직}

var1 = CoefficientRArray(Arr1,Arr2,5);

#두 배열의 0~4번방의 피어슨 상관계수R을 var1에 저장

RSquareArray

설명 : 두 배열의 피어슨 상관계수R의 제곱

함수 : RSquareArray(IndepArray, DepArray, Size)

※ IndepArray : 독립배열의 이름, 1차원 수치형 배열만 가능

※ DepArray : 종속배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr1[20](0), Arr2[20](0);

{Array 값 할당 로직}

var1 = RSquareArray(Arr1,Arr2,5);

#두 배열의 0~4번방의 피어슨 상관계수R 제곱을 var1에 저장

CorrelationArray

설명 : 두 배열의 상관계수

함수 : CorrelationArray(IndepArray, DepArray, Size)

※ IndepArray : 독립배열의 이름, 1차원 수치형 배열만 가능

※ DepArray : 종속배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr1[20](0), Arr2[20](0);

{Array 값 할당 로직}

var1 = CorrelationArray(Arr1,Arr2,5); #두 배열의 0~4번방의 상관계수를 var1에

저장

SortArray

설명 : 1차원 배열 정렬

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : SortArray(PriceArray, Size, HiLo)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ HiLo : 내림차순 : 1, 오름차순 : -1

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = SortArray(Arr,5,1); #0~4번방의 값을 내림차순으로 정렬하고 var1에 결과 저장

MedianArray

설명 : 배열을 오름차순으로 정렬하고 중간값 반환

함수 : MedianArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = MedianArray(Arr, 5); #0~4번방을 오름차순으로 정렬 후 중간값을 var1에 저장

QuartileArray

설명 : 배열의 사분위수 반환

함수 : QuartileArray(QRank, PriceArray, Size)

※ QRank : 분위수, 0~4의 정수 (0:0%, 1:25%, 2:50%, 3:75%, 4:100%)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = QuartileArray(2, Arr, 5);#0~4번방의 1사분위수(25%)값을 var1에 저장

NumericRankArray

설명 : 배열을 정렬 후 지정한 순위의 값을 반환

함수 : NumericRankArray(PriceToRank, PriceArray, Size, SortOrder)

※ PriceToRank : 순위

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size: 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ SortOrder: 내림차순 : 1, 오름차순 : -1

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = NumericRankArray(2, Arr, 5, 1);
#0~4번방을 내림차순으로 정렬 후 지정한 순위의 값을 var1에 저장

PercentRankArray

설명 : 배열에서 지정한 값의 백분율

함수 : PercentRankArray(PriceToRank, PriceArray, Size)

- ※ PriceToRank : 백분율을 계산할 값
- ※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능
- ※ Size: 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = PercentRankArray(40, Arr, 5); #0~4번방에서 40의 백분율을 var1에 저장

PercentileArray

설명 : 배열의 지정한 순위율에 해당하는 값

함수 : PercentileArray(PctRank, PriceArray, Size)

- ※ PctRank : 순위율, 25%면 0.25
- ※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능
- ※ Size: 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = PercentileArray(0.25, Arr, 5); #0~4번방을 25% 순위값을 var1에 저장

KurtosisArray

설명 : 배열의 첨도

함수 : KurtosisArray(PriceArray, Size)

- ※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능
- ※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);
{Array 값 할당 로직}
var1 = KurtosisArray(Arr, 5); #0~4번방의 첨도를 var1에 저장

SkewArray

설명 : 배열의 왜도

함수 : SkewArray(PriceArray, Size)

- ※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

- ※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = SkewArray(Arr, 5); #0~4번방의 왜도를 var1에 저장

LinRegArray

설명 : 배열의 선형회귀선의 기울기, 각도, 절편, 예측값

- ※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : LinRegArray(PriceArray, Size, TgtPos, oLRSlope, oLRAngle, oLRIntercept, oLRValueRaw)

- ※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

- ※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

- ※ TgtPos : 예측지점, 미래는 음수 과거는 양수

- ※ oLRSlope : 선형회귀선의 기울기, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

- ※ oLRAngle : 선형회귀선의 각도, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

- ※ oLRIntercept : 선형회귀선의 절편, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

- ※ oLRValueRaw : 선형회귀선의 값, NumericRef

(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = LinRegArray(Arr, 5, -2, Slope, Angle, Intercept, Value);

#0~4번방의 선형회귀선의 기울기는 Slope, 각도는 Angle, 편차는 Intercept,

예측값은 Value에 저장하고 수행결과를 var1에 저장

LinRegArray2

설명 : 두 배열의 선형회귀선의 기울기, 각도, 절편, 예측값

- ※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : LinRegArray2(IndepArray, DepArray Size, TgtPos, oLRSlope, oLRAngle, oLRIntercept, oLRValueRaw)

- ※ IndepArray : 독립배열의 이름, 1차원 수치형 배열만 가능

- ※ DepArray : 종속배열의 이름, 1차원 수치형 배열만 가능
- ※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)
- ※ TgtPos : 예측지점, 미래는 음수 과거는 양수
- ※ oLRSlope : 선형회귀선의 기울기, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRAngle : 선형회귀선의 각도, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRIntercept : 선형회귀선의 절편, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)
- ※ oLRValueRaw : 선형회귀선의 값, NumericRef
(함수의 매개변수로 지정한 변수에 값을 저장해 리턴)

예제 : Array : Arr1[20](0),Arr2[20](0);
 {Array 값 할당 로직}
 var1 = LinRegArray2(Arr1, Arr2, 5,-2, Slope, Angle, Intercept, Value);
 #두 배열 0~4번방의 선형회귀선의 기울기는 Slope, 각도는 Angle, 편차는 Intercept,
 예측값은 Value에 저장하고 수행결과를 var1에 저장

LinRegForecastArray

설명 : 배열의 선형회귀를 기반으로 주어진 X값에 대한 예측값

함수 : LinRegForecastArray(PriceArray, Size, TgtPos)

- ※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능
- ※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)
- ※ TgtPos : 예측지점, 미래는 음수 과거는 양수

예제 : Array : Arr[20](0);
 {Array 값 할당 로직}
 var1 = LinRegForecastArray (Arr, 5, -5);
 #0~4번방의 선형회귀선의 예측값을 var1에 저장

LinRegInterceptArray

설명 : 배열의 선형회귀선의 절편

함수 : LinRegInterceptArray(PriceArray, Size)

- ※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능
- ※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);
 {Array 값 할당 로직}
 var1 = LinRegInterceptArray(Arr, 5); #0~4번방의 선형회귀선의 절편을 var1에 저장

LinRegSlopeArray

설명 : 배열의 선형회귀선의 기울기

함수 : LinRegSlopeArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = LinRegSlopeArray(Arr, 5); #0~4번방의 선형회귀선의 기울기를 var1에 저장

NormalCumDensityArray

설명 : 배열의 정규누적분포

함수 : NormalCumDensityArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = NormalCumDensityArray(Arr, 5); #0~4번방의 누적정규분포

NormalDensityArray

설명 : 배열의 정규분포

함수 : NormalDensityArray(PriceArray, Size)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = NormalDensityArray(Arr, 5); #0~4번방의 정규분포

StandardizeArray

설명 : 배열의 표준편차 및 평균에 따라 표준화된 값을 반환

함수 : StandardizeArray(PriceArray, Size, NumDevs)

※ PriceArray : 배열의 이름, 1차원 수치형 배열만 가능

※ Size : 배열 사이즈(방갯수), 계산은 항상 0번방부터 시작(0번방부터 N개)

※ NumDevs : 표준편차수

예제 : Array : Arr[20](0);

{Array 값 할당 로직}

var1 = StandardizeArray(Arr, 5, 2); #0~4번방의 표준화값을 var1에 저장

Sort2DArray

설명 : 2차원 배열정렬.

지정한 범위를 0행을 기준으로 내림차순/오름차순으로 정렬.

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : Sort2DArray(Array, RowSize, ColSize, HiLo)

※ Array : 2차원 배열의 이름, 2차원 수치형 배열만 가능

※ RowSize : 행범위 사이즈, 시작행 포함 N개

※ ColSize : 열범위 사이즈, 시작열 포함 N개

※ HiLo : 내림차순 : 1, 오름차순 : -1

예제 : Array : Arr2[4,5](0);

{ Array 값 할당 로직 }

var1 = Sort2DArray(Arr2, 3, 3, 1);

2차원 배열변수(Arr2)의 0행0열~2행2열 영역을 내림차순으로 정렬하고

수행결과는 var1에 저장

Array : Arr2[4,5](0);

대상배열

Arr2

		0면				
		0	1	2	3	4 열
정렬기준행	0	64	1	85	66	44
	1	6	20	25	83	95
	2	57	30	50	92	3
	3	97	68	80	67	57
행						

Sort2DArray(Arr2, 3, 3, 1);

정렬 후 대상배열

Arr2

		0면				
		0	1	2	3	4 열
정렬 후	0	85	64	1	66	44
	1	25	6	20	83	95
	2	50	57	30	92	3
	3	97	68	80	67	57
행						

SortArray2D

설명 : 2차원 배열정렬.

시작행열을 기준으로 범위를 지정하고 시작행을 기준으로 내림차순/오름차순으로 정렬

※ 정상수행되면 1반환, 오류가 있으면 -1 반환

함수 : SortArray2D(Array,RowSize,ColSize,HiLo,StartRowNo,StartColNo)

- ※ Array : 2차원 배열의 이름, 2차원 수치형 배열만 가능
- ※ RowSize : 행범위 사이즈, 시작행 포함 N개
- ※ ColSize : 열범위 사이즈, 시작열 포함 N개
- ※ HiLo : 내림차순 : 1, 오름차순 : -1
- ※ StartRowNo: 시작 행번호
- ※ StartColNo: 시작 열번호

예제 : Array : Arr2[4,5](0);

{ Array 값 할당 로직 }

var1 = SortArray2D (Arr2,3,3,1,1,1);

2차원 배열변수(Arr2)의 1행1열~3행3열 영역을 내림차순으로 정렬하고
수행결과는 var1에 저장

Array : Arr2[4,5](0);

대상배열

Arr2

		0면				
		0	1	2	3	4 열
정렬기준행	0	64	1	85	66	44
	1	6	20	25	83	95
	2	57	30	50	92	3
	3	97	68	80	67	57
		행				

SortArray2D(Arr2,3,3,1,1,1);

정렬 후 대상배열

Arr2

		0면				
		0	1	2	3	4 열
정렬 후 대상배열	0	85	64	1	66	44
	1	25	83	25	20	95
	2	50	92	50	30	3
	3	97	67	80	68	57
		행				

5) 출력함수

Plot1~Plot99

설명 : 지표 출력 함수, 지표식과 검색식에서만 사용가능.

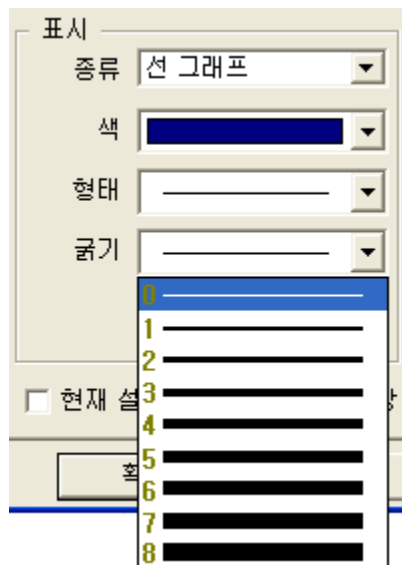
지표식은 기본 그래프 종류가 선그래프이고 검색식은 점그래프이다.

함수 : Plot1(출력대상, "이름",색상,def,굵기)

수치값만 출력대상이 될 수 있고 네번째 매개변수는 기능이 없으므로 def로 표기해야 한다.

굵기는 순서대로 0,1,2,3,4,5,6,7,8로 지정하면 속성창에서 기본으로 제공되는 선의 굵기

중에 선택 하는 것과 같다.



예제 : `plot1(ma(c,20), "이동평균",red,def,2);`

→ 20기간 이동평균을 굵기2의 붉은색으로 차트에 그려라

FixPlotShift

설명 : plot 이동 함수, 지표식과 검색식에서만 사용가능.

지정한 plot번호를 차트상 좌우 N개봉으로 이동합니다.

함수 : `FixPlotShift (PlotNum,Bar)`

PlotNum : plot번호

Bar : 이동할 봉갯수

(양수로 지정하면 우측으로, 음수로 지정하면 좌측으로 지정한 봉갯수 만큼 이동)

예제 : `var1 = (highest(H,52)+lowest(L,52))/2;`

`plot1(var1,"선행스팬2",RED);`

`FixPlotShift(1,25);` #plot1을 오른쪽으로 25개봉 이동



```
var1 = (highest(H,52)+lowest(L,52))/2;
plot1(var1,"선행스팬2",RED);
FixPlotShift(1,-25); #plot1을 왼쪽으로 25개봉 이동
```



PlotBaseLine1 ~ PlotBaseLine99

설명 : 지표 출력 함수. Plot과 같은 기능의 함수

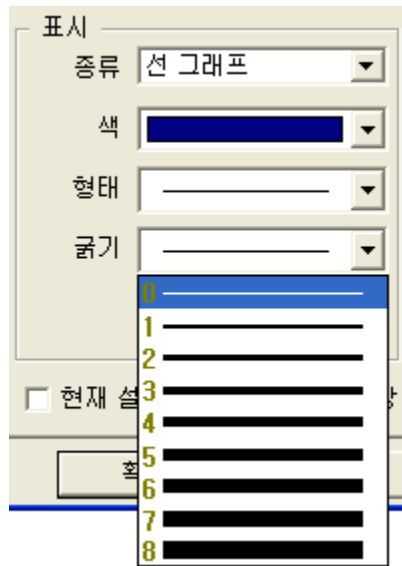
지표식, 검색식에서만 사용가능 하다.

지표식은 기본 그래프 종류가 선그래프이고 검색식은 점그래프이다.

함수 : PlotBaseLine1(출력대상, "이름", 색상, def, 굵기)

수치값만 출력대상이 될 수 있고 네번째 매개변수는 기능이 없으므로 def로 표기해야 한다.

굵기는 순서대로 0,1,2,3,4,5,6,7,8로 지정하면 속성창에서 기본으로 제공되는 선의 굵기 중에 선택 하는 것과 같다.



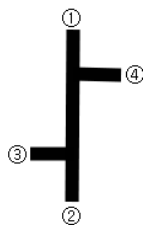
예제 : `plot1(ma(c,20), "이동평균",red,def,2);`
 → 20기간 이동평균을 굵기2의 붉은색으로 차트에 그려라

PlotPaintBar

설명 : 강조함수. 봉에 색을 덧 입힘

강조식에서만 사용가능

함수 : `PlotPaintBar(상단,하단,좌측표시,우측표시, "이름",색)`



※ 1로 지정한 상단에서 2로 지정한 하단까지 값까지 색을 칠하며
 좌측표시로 지정한 값은 3과 같이 우측표시로 지정한 값은 4와 같이
 나타난다.

예제 : 5-20-60 정배열 구간에 강조표시

```

input : Short(5),Mid(20),Long(60);

Value1 = ma(c, Short);
Value2 = ma(c, Mid);
5 Value3 = ma(c, Long);

if Value1 > Value2 and Value2 > value3 Then
begin
    PlotPaintBar( High,Low,Close,Open, "정배열",BLACK);
10 end
else
    NoPlot(1);

```

Print

설명 : txt나 csv 파일로 지정한 데이터 추출

실시간 차트에서만 추출가능.

함수 : Print("파일이름", "Expression", Data,...)

※ 파일이름 : 텍스트문서 이름

※ Expression : 표현 방식 (수치는 %.f를 문자열은 %s)

※ Data : 계산식 혹은 대상 데이터

예제 : 4.에스랭귀지 활용 → 데이터추출 참고

MessageLog

설명 : 디버깅창에 지정한 데이터 표시

함수 : MessageLog ("Expression", Data,...)

※ Expression : 표현 방식 (수치는 %.f를 문자열은 %s)

※ Data : 계산식 혹은 대상 데이터

예제 : 4.에스랭귀지 활용 → 디버깅 참고

ClearDebug

설명 : 디버깅창의 내용을 모두 삭제

연속적으로 MessageLog가 호출되면 기존의 표시된 내용에 다시 추가로

내용이 누적 표시되므로 MessageLog 발동시 먼저 디버깅창의 모든 내용을 삭제하고
표시하고자 할 때 사용한다.

함수 : ClearDebug

예제 :

```

var: value(0),cond(False);

value = abs(c-0);
cond = C < 0;
5 ClearDebug;
MessageLog("%.2f %s",value,cond);
#디버깅 화면의 기존 내용을 삭제한 후
#value는 소숫점2자리로 표시 cond는 스트링으로 표시

```

PlaySound

설명 : 소리출력

함수 : PlaySound("wav파일의 경로")

경로와 함께 파일명까지 지정해 주어야 한다.

예제 : condition1조건에 만족하면 지정된 경로의 wav파일을 실행

```

if Condition1 == true Then
    playsound("C:\예스트레이더31\WData\alert.wav");

```

Alert

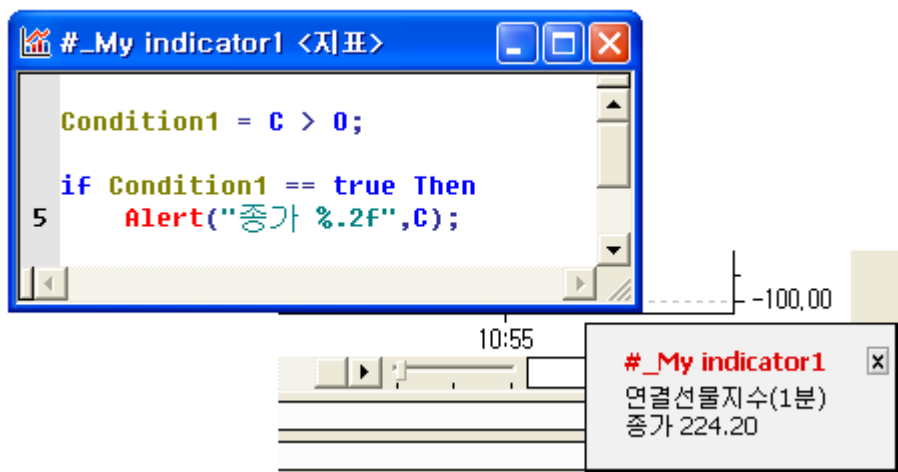
설명 : 알림 팝업창 생성

함수 : Alert("Expression", Data,...)

※ Expression : 표현 방식 (수치는 %.f를 문자열은 %s)

※ Data : 계산식 혹은 대상 데이터

예제 : condition1조건에 만족하면 알림 팝업창으로 생성하고 종가를 표시



AlertEnabled

설명 : 현재 알림팝업 설정 상태를 리턴

함수 : AlertEnabled

예제 : condition1조건에 만족하면 지정된 경로의 wav파일을 실행

SetAlert

설명 : 경보설정 변경

함수 : SetAlert(True/false)

예제 : If TRUE == AlertEnabled() then Alert("알림팝업 창 생성");

팝업 기능이 설정 되며 Alert() 함수의 조건이 만족하면 알림팝업창에 당 메시지를 출력한다

NoPlot

설명 : 해당 봉의 결과를 표시하지 않는다

함수 : Noplot(N)

※ N은 Plot number

예제 : 양봉일때는 종가를 출력하고 아니면 값을 표시하지 않는다

if C > O Then

plot1(C);

Else

NoPlot(1);

RGB

설명 : 색상 출력

색상예약어 중 원하는 색상이 없을 때 RGB값으로 색을 지정할 수 있다.

함수 : RGB(num1,num2,num3)

예제 : plot1(ma(c,20), "이동평균",RGB(255,182,193));

20이동평균값을 연한 핑크색으로 출력

Named	Numeric	Color Name	Decimal
		Light Pink	255,182,193
		Pink	255,192,203
		Crimson	220,20,60
		Lavender Blush	255,240,245
		Pale Violet Red	219,112,147
		Hot Pink	255,105,180
		Deep Pink	255,20,147

6) 추세선출력함수

TL_New

설명 : 기본차트 영역에 지정된 두개의 지점을 연결하는 추세선 개체를 생성합니다.

새로운 추세선 개체가 생성되면 고유의 식별번호가 자동 부여되며

아래와 같이 작성해 식별번호를 변수에 저장할수 있습니다.

```
var1 = TL_New(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)
```

함수 : TL_New(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)

※ StartDate는 추세선 개체의 시작점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ StartTime은 추세선 개체의 시작점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ StartValue는 추세선 개체의 시작점의 가격(Y축)입니다.

※ EndtDate는 추세선 개체의 끝점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ EndTime은 추세선 개체의 끝점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ EndValue는 추세선 개체의 끝점의 가격(Y축)입니다.

예제 : var : stok(0),stod(0),TL1(0);

```
stok = StochasticsK(20,10);
```

```
stod = StochasticsD(20,10,10);
```

```
#골드 발생
```

```
if crossup(stok,stod) Then{
```

```
    value1 = sdate; #날짜
```

```
    value2 = stime; #시간
```

```
    value3 = C; #종가
```

```
    value11 = value1[1];#이전 골드시 날짜
```

```
    value22 = value2[1];#이전 골드시 시간
```

```
    value33 = value3[1];#이전 골드시 종가
```

```
#이전 골드지점의 종가와 현재 골드지점의 종가를 연결하는 추세선 출력
```

```
TL1 = TL_New(value11,value22,value33,value1,value2,value3);
```

```
}
```

주의 : 추세선 개체의 식별번호는 미리 지정된 양식에 따라 자동 부여됩니다.

식별번호를 사용하기 위해서는 반드시 숫자형 변수에 할당하고 그 변수를 사용해야 합니다.

TL_New_Self

설명 : 지표 영역에 지정한 두개의 지점을 연결하는 추세선 개체를 생성합니다.

새로운 추세선 개체가 생성되면 고유의 식별번호가 자동 부여되며

아래와 같이 작성해 식별번호를 변수에 저장할수 있습니다.

```
var1 = TL_New_Self(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)
```

함수 : TL_New_Self(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)

※ StartDate는 추세선 개체의 시작점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ StartTime은 추세선 개체의 시작점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ StartValue는 추세선 개체의 시작점의 가격(Y축)입니다.

※ EndtDate는 추세선 개체의 끝점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ EndTime은 추세선 개체의 끝점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ EndValue는 추세선 개체의 끝점의 가격(Y축)입니다.

예제 : var : stok(0),stod(0),TL1(0);

```
stok = StochasticsK(20,10);
```

```
stod = StochasticsD(20,10,10);
```

```
#지표출력
```

```
plot1(stok);
```

```
plot2(stod);
```

```
#골드 발생
```

```
if crossup(stok,stod) Then{
```

```
    value1 = sdate; #날짜
```

```
    value2 = stime; #시간
```

```
    value3 = stok; #k값
```

```
    value11 = value1[1];#이전 골드시 날짜
```

```
    value22 = value2[1];#이전 골드시 시간
```

```
    value33 = value3[1];#이전 골드시 k값
```

```
#이전 골드지점의 k값과 현재 골드지점의 k값을 연결하는 추세선 출력
```

```
TL1 = TL_New_Self(value11,value22,value33,value1,value2,value3);
```

```
}
```

TL_Delete

설명 : 지정한 추세선 개체를 삭제합니다.

함수 : TL_Delete(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

예제 : TL_Delete(TL1);

식별번호가 TL1인 추세선 개체를 삭제합니다.

TL_SetBegin

설명 : 지정한 추세선 개체의 시작점을 새로 설정합니다.

함수 : TL_SetBegin(TL_ID,StartDate,StartTime,StartValue)

- ※ TL_ID는 추세선 개체의 식별번호이거나 식별번호를 뜻하는 수학적 표현입니다.
- ※ StartDate는 추세선 개체의 시작점 날짜입니다. 날짜는 YYYYMMDD 형식입니다.
- ※ StartTime은 추세선 개체의 시작점 시간입니다. 시간은 HHMMSS 형식입니다.
- ※ StartValue는 추세선 개체의 시작점 가격(Y축)입니다.

예제 : if Condition1 == true then

TL_SetBegin(TL1,sdate,stime,L);

특정조건이 만족하면 식별번호가 TL1인 추세선 개체의 시작점을
조건만족시점의 날짜, 시간, 지정한 값으로 변경합니다.

TL_SetEnd

설명 : 지정한 추세선 개체의 끝점을 새로 설정합니다.

함수 : TL_SetEnd(TL_ID,EndDate,EndTime,EndValue)

- ※ TL_ID는 추세선 개체의 식별번호
- ※ EndtDate는 추세선 개체의 끝점 날짜입니다. 날짜는 YYYYMMDD 형식입니다.
- ※ EndTime은 추세선 개체의 끝점 시간입니다. 시간은 HHMMSS 형식입니다.
- ※ EndValue는 추세선 개체의 끝점 가격(Y축)입니다.

예제 : if Condition2 == true then

TL_SetEnd(TL1,sdate,stime,L);

특정조건이 만족하면 식별번호가 TL1인 추세선 개체의 끝점을
조건만족시점의 날짜, 시간, 지정한 값으로 변경합니다.

TL_SetColor

설명 : 지정한 추세선 개체의 색상을 설정합니다.

함수 : TL_SetColor(TL_ID,Color)

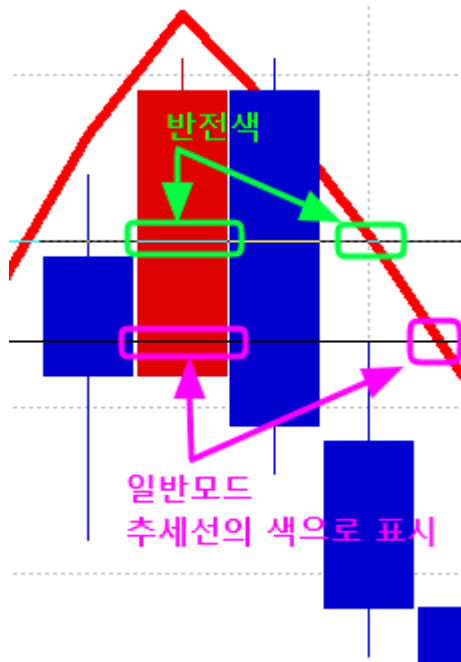
- ※ TL_ID는 추세선 개체의 식별번호
- ※ Color는 색상예약어나 RGB함수로 지정할 수 있습니다.

예제 : TL_SetColor(TL1,RED);

식별번호가 TL1인 추세선 개체의 색을 빨강색으로 설정합니다.

TL_SetDrawMode

설명 : 지정한 추세선 개체가 봉이나 다른 추세선/지표 등과 겹칠 경우 반전색으로 표시할지 여부를 설정합니다.



함수 : TL_SetDrawMode (TL_ID,Mode)

※ TL_ID는 추세선 개체의 식별번호

※ Mode는 0으로 설정하면 일반모드(지정된 추세선 색상)로 표시 되고 ,

1로 설정하면 반전색으로 표시, 기본값은 1

예제 : TL_SetDrawMode(TL1,1);

식별번호가 TL1인 추세선 개체가 봉이나 다른 추세선, 지표 등과 겹치면 반전색으로 표시합니다.

TL_SetSize

설명 : 지정한 추세선 개체의 굵기를 설정합니다.

함수 : TL_SetSize(TL_ID,Size)

※ TL_ID는 추세선 개체의 식별번호

※ Size는 추세선 개체의 굵기이며 1~5까지의 범위에서 정수로 지정 가능합니다.

※ 1번이 가장 얇고 5번이 가장 두껍습니다.

예제 : TL_SetSize(TL1,1);

식별번호가 TL1인 추세선 개체의 굵기를 1로 설정합니다.

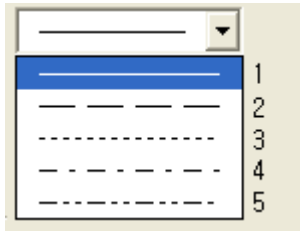
TL_SetStyle

설명 : 지정한 추세선 개체의 선의 형태를 설정합니다.

함수 : TL_SetStyle(TL_ID,Style)

※ TL_ID는 추세선 개체의 식별번호

※ Style은 속성화면에 제공하는 부분과 같습니다.위에서부터 1~5번입니다.



예제 : `TL_SetSize(TL1,1);`

식별번호가 TL1인 추세선 개체의 형태를 1번 형태로 설정합니다.

TL_SetExtLeft

설명 : 지정한 추세선 개체의 왼쪽 확장을 설정합니다.

함수 : `TL_SetExtLeft(TL_ID,ExtTF)`

※ TL_ID는 추세선 개체의 식별번호

※ ExtTF는 True로 설정하면 왼쪽으로 확장, false는 확장하지 않는다는 설정입니다.

예제 : `TL_SetExtLeft(TL1,True);`

식별번호가 TL1인 추세선 개체를 왼쪽으로 확장합니다.

TL_SetExtRight

설명 : 지정한 추세선 개체의 오른쪽 확장을 설정합니다.

함수 : `TL_SetExtRight(TL_ID,ExtTF)`

※ TL_ID는 추세선 개체의 식별번호

※ ExtTF는 True로 설정하면 오른쪽으로 확장, false는 확장하지 않는다는 설정입니다.

예제 : `TL_SetExtRight(TL1,True);`

식별번호가 TL1인 추세선 개체를 오른쪽으로 확장합니다.

TL_GetBeginDate

설명 : 지정한 추세선 개체의 시작점의 날짜를 리턴합니다.

함수 : `TL_GetBeginDate(TL_ID)`

※ TL_ID는 추세선 개체의 식별번호

※ YYYYMMDD형태로 리턴됩니다.

예제 : `var1 = TL_GetBeginDate(TL1);`

식별번호가 TL1인 추세선 개체의 시작점의 날짜를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 TL1인 추세선 개체의 시작점의 날짜가 2012년 04월 18일이면

리턴값은 20120418입니다.

TL_GetBeginTime

설명 : 지정한 추세선 개체의 시작점의 시간을 리턴합니다.

함수 : TL_GetBeginTime(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

※ HHMMSS형태로 리턴됩니다

예제 : var1 = TL_GetBeginTime(TL1);

식별번호가 TL1인 추세선 개체의 시작점의 시간을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 TL1인 추세선 개체의 시작점의 시간이 10시라면 리턴값은 100000입니다.

TL_GetBeginVal

설명 : 지정한 추세선 개체의 시작점(Y축)의 값을 리턴합니다.

함수 : TL_GetBeginVal(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

예제 : var1 = TL_GetBeginTime(TL1);

식별번호가 TL1인 추세선 개체의 시작점(Y축)의 값을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

TL_GetEndDate

설명 : 지정한 추세선 개체의 끝점의 날짜를 리턴합니다.

함수 : TL_GetEndDate(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

※ YYYYMMDD형태로 리턴됩니다.

예제 : var1 = TL_GetEndDate(TL1);

식별번호가 TL1인 추세선 개체의 끝점의 날짜를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 TL1인 추세선 개체의 끝점의 날짜가 2012년 04월 18일이면

리턴값은 20120418입니다.

TL_GetEndTime

설명 : 지정한 추세선 개체의 끝의 시간을 리턴합니다.

함수 : TL_GetEndTime(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

※ HHMMSS 형태로 리턴됩니다.

예제 : var1 = TL_GetEndTime(TL1);

식별번호가 TL1인 추세선 개체의 끝점의 시간을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 TL1인 추세선 개체의 끝점의 시간이 12시라면 리턴값은 120000입니다.

TL_GetEndVal

설명 : 지정한 추세선 개체의 끝점의 값(Y축)을 리턴합니다.

함수 : TL_GetEndVal(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

예제 : var1 = TL_GetEndVal(TL1);

식별번호가 TL1인 추세선 개체의 끝점의 값(Y축)을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 됩니다.

TL_GetValue

설명 : 지정한 추세선 개체의 특정일 특정시간의 값을 리턴합니다.

함수 : TL_Getvalue(TL_ID,날짜,시간)

※ TL_ID는 추세선 개체의 식별번호

※ 날짜는 YYYYMMDD 형식입니다.

※ 시간은 HHMMSS 형식입니다.

예제 : var1 = TL_Getvalue(TL1,20120418,110000);

식별번호가 TL1인 추세선 개체의 2012년 4월 18일 11시의 값을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

TL_GetExtLeft

설명 : 지정한 추세선 개체의 왼쪽 확장 여부를 리턴합니다.

함수 : TL_GetExtLeft(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

※ 지정한 추세선 개체가 왼쪽으로 확장이 설정되어 있으면 True,

설정되지 않았으면 False를 리턴합니다

예제 : Condition1 = TL_GetExtLeft(TL1);

식별번호가 TL1인 추세선 개체의 왼쪽 확장 여부를 condition1변수에 저장합니다.

리턴값은 논리형 변수에 저장해야 합니다.

TL_GetExtRight

설명 : 지정한 추세선 개체의 오른쪽 확장 여부를 리턴합니다.

함수 : TL_GetExtRight(TL_ID)

※ TL_ID는 추세선 개체의 식별번호

※ 지정한 추세선 개체가 오른쪽으로 확장이 설정되어 있으면 True,
설정되지 않았으면 False를 리턴합니다

예제 : `Condition1 = TL_GetExtRight(TL1);`

식별번호가 TL1인 추세선 개체의 오른쪽 확장 여부를 condition1변수에 저장합니다.
리턴값은 논리형 변수에 저장해야 합니다.

TL_GetColor

설명 : 지정한 추세선 개체에 설정 된 색상값을 리턴합니다.

함수 : `TL_GetColor(TL_ID)`

※ TL_ID는 추세선 개체의 식별번호

예제 : `var1 = TL_GetColor(TL1);`

식별번호가 TL1인 추세선 개체의 색상을 var1변수에 저장합니다.
리턴값은 수치형 변수에 저장해야 합니다.

TL_GetDrawMode

설명 : 지정한 추세선 개체의 반전색 표시여부를 리턴합니다.

함수 : `TL_GetDrawMode (TL_ID)`

※ TL_ID는 추세선 개체의 식별번호

예제 : `var1 = TL_GetDrawMode(TL1);`

식별번호가 TL1인 추세선 개체의 반전색 설정을 var1변수에 저장합니다.
리턴값은 수치형 변수에 저장해야 합니다.

TL_GetSize

설명 : 지정한 추세선 개체에 설정된 굵기를 리턴합니다.

함수 : `TL_GetSize(TL_ID)`

※ TL_ID는 추세선 개체의 식별번호

예제 : `var1 = TL_GetSize(TL1);`

식별번호가 TL1인 추세선 개체의 굵기를 var1에 저장합니다.
리턴값은 수치형 변수에 저장해야 합니다.

TL_GetStyle

설명 : 지정한 추세선 개체에 설정된 선의 형태를 리턴합니다.

함수 : `TL_GetStyle(TL_ID)`

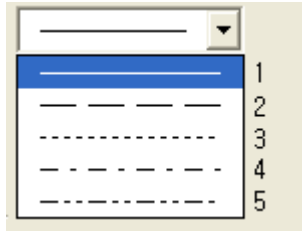
※ TL_ID는 추세선 개체의 식별번호

예제 : `var1 = TL_GetStyle(TL1);`

식별번호가 TL1인 추세선 개체의 형태를 var1에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

반환값은 1~5 입니다.



7) 텍스트출력함수

Text_New

설명 : 기본차트 영역의 지정된 위치에 새로운 텍스트 개체를 생성합니다.

새로운 텍스트 개체가 생성되면 고유의 식별번호가 자동 부여되며

아래와 같이 작성해 식별번호를 저장할수 있습니다.

```
var1 = Text_New(BarDate,BarTime,Price, "text")
```

함수 : Text_New(BarDate,BarTime,Price, "text")

※ BarDate는 텍스트 개체를 출력할 봉의 날짜입니다. 날짜 형식은 YYYYMMDD입니다.

※ BarTime은 텍스트 개체를 출력할 봉의 시간입니다. 시간 형식은 HHMMSS입니다.

※ Price는 가격위치(Y축)입니다.

※ Text는 출력할 문자열입니다. ""와 함께 표현해야 합니다.

숫자는 NumToStr 함수를 이용해 텍스트로 변경해 지정하면 됩니다.

예제 : `var : stok(0),stod(0),tx1(0);`

```
stok = StochasticsK(20,10);
```

```
stod = StochasticsD(20,10,10);
```

```
#지표출력
```

```
plot1(stok);
```

```
plot2(stod);
```

```
#골드 발생
```

```
if crossup(stok,stod) Then{
```

```
    value1 = sdate; #날짜
```

```
    value2 = stime; #시간
```

```
    value3 = C; #종가
```

```
    value4 = stok; #k값
```



```

        #기본차트 종가 위치에 골드봉 k값을 텍스트로 출력
        tx1 = Text_New(value1,value2,value3,NumToStr(value4,2));
    }

```

주의 : 텍스트의 식별번호는 미리 지정된 양식에 따라 부여됩니다.

10001부터 N번까지 번호가 부여되므로 추세선 개체의 식별번호를 사용하기 위해서는 반드시 숫자형 변수에 할당하고 그 변수를 사용해야 합니다.

Text_New_Self

설명 : 지표 영역의 지정한 위치에 새로운 텍스트 개체를 생성합니다.

새로운 텍스트 개체가 생성되면 고유의 식별번호가 자동 부여되며

아래와 같이 작성해 식별번호를 저장할수 있습니다.

```
var1 = Text_New_Self(BarDate,BarTime,Price, "text")
```

함수 : Text_New_Self(BarDate,BarTime,Price, "text")

※ BarDate는 텍스트 개체를 출력할 봉의 날짜입니다. 날짜 형식은 YYYYMMDD입니다.

※ BarTime은 텍스트 개체를 출력할 봉의 시간입니다. 시간 형식은 HHMMSS입니다.

※ Price는 가격위치(Y축)입니다.

※ Text는 출력할 문자열입니다. ""와 함께 표현해야 합니다.

숫자는 NumToStr 함수를 이용해 텍스트로 변경해 지정하면 됩니다.

예제 : var : stok(0),stod(0),tx1(0);

```
stok = StochasticsK(20,10);
```

```
stod = StochasticsD(20,10,10);
```

```
#지표출력
```

```
plot1(stok);
```

```
plot2(stod);
```

```
#골드 발생
```

```
if crossup(stok,stod) Then{
```

```
    value1 = sdate; #날짜
```

```
    value2 = stime; #시간
```

```
    value3 = stok; #k값
```

```
#지표의 k값 위치에 k값을 텍스트로 출력
```

```
tx1 = Text_New_Self(value1,value2,value3,NumToStr(value3,2));
```

}

주의 : 텍스트의 식별번호는 미리 지정된 양식에 따라 부여됩니다.

10001부터 N번까지 번호가 부여되므로 추세선 개체의 식별번호를 사용하기 위해서는 반드시 숫자형 변수에 할당하고 그 변수를 사용해야 합니다.

Text_Delete

설명 : 지정한 텍스트 개체를 삭제합니다.

함수 : Text_Delete(Text_ID)

※ Text_ID는 텍스트 개체의 식별번호

예제 : Text_Delete(Text1);

식별번호가 Text1인 텍스트를 삭제합니다.

Text_SetColor

설명 : 지정한 텍스트 개체의 색상을 설정합니다.

함수 : Text_SetColor(Text_ID,Color)

※ Text_ID는 텍스트 개체의 식별번호

※ Color는 색상예약어나 RGB함수로 지정할 수 있습니다.

예제 : Text_SetColor(Text1,red)

식별번호가 Text1인 텍스트의 색을 빨간색으로 설정합니다.

Text_SetBold

설명 : 지정한 텍스트 개체를 굵게 설정합니다.

함수 : Text_SetBold(Text_ID, Bold)

※ Text_ID는 텍스트 개체의 식별번호

※ Bold (0:보통, 1: 굵게)

예제 : Text_SetBold (Text1,r1)

식별번호가 Text1인 텍스트를 굵게 설정합니다.

Text_SetFont

설명 : 지정한 텍스트 개체의 폰트를 설정합니다.

함수 : Text_SetFont(Text_ID,Font)

※ Text_ID는 텍스트 개체의 식별번호

※ Font는 폰트명을 텍스트로 지정

예제 : Text_SetFont(Text1,"굴림")

식별번호가 Text1인 텍스트의 폰트를 굴림으로 설정합니다.

Text_SetSize

설명 : 지정한 텍스트 개체의 크기를 설정합니다.

함수 : Text_SetSize (Text_ID,Size)

※ Text_ID는 텍스트 개체의 식별번호

※ Size는 장수로 지정

예제 : Text_SetSize (Text1,10)

식별번호가 Text1인 텍스트의 크기를 10으로 설정합니다.

Text_SetLocation

설명 : 지정한 텍스트 개체의 위치를 변경합니다.

함수 : Text_SetLocation(Text_ID,BarDate,BarTime,Price)

※ Text_ID는 텍스트 개체의 식별번호

※ BarDate는 텍스트를 출력할 봉의 날짜입니다. 날짜 형식은 YYYYMMDD입니다.

※ BarTime은 텍스트를 출력할 봉의 시간입니다. 시간 형식은 HHMMSS입니다.

※ Price는 가격위치(Y축)입니다.

예제 : Text_SetLocation(Text1,20120420,100000,H+0.5)

식별번호가 Text1인 텍스트 개체를 2012년 4월 20일 10시 00분 봉의 고가+0.5 위치로 이동합니다.

Text_SetString

설명 : 지정한 텍스트 개체의 출력문자열을 새로운 문자열로 설정합니다.

함수 : Text_SetString(Text_ID, "New Text")

※ Text_ID는 텍스트 개체의 식별번호

※ New Text는 사용할 문자열입니다. ""와 함께 사용해야 합니다.

예제 : Text_SetString(Text1, "조건만족해제")

식별번호가 Text1인 텍스트 개체의 출력 문자열을 조건만족해제로 변경합니다.

Text_SetStyle

설명 : 지정한 텍스트 개체의 수평 및 수직 정렬 위치를 설정합니다

함수 : Text_SetStyle(Text_ID, Horizon, Vertical)

※ Text_ID는 텍스트 개체의 식별번호

※ Horizon은 수평 위치입니다. 0 왼쪽, 1 오른쪽, 2 중앙

※ Vertical은 수직 위치입니다. 0 상단, 1 하단, 2 중앙

예제 : Text_SetStyle(Text1, 2,2);

식별번호가 Text1인 텍스트 개체를 수평정렬은 중앙, 수직정렬은 중앙으로 설정합니다.

Text_GetColor

설명 : 지정한 텍스트 개체의 색상값을 리턴 합니다.

함수 : Text_GetColor(Text_ID)

※ Text_ID는 텍스트 개체의 식별번호

예제 : var1 = Text_GetColor(Text1);

식별번호가 Text1인 텍스트의 색상값을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Text_GetDate

설명 : 지정한 텍스트 개체의 출력 봉의 날짜를 리턴 합니다.

함수 : Text_GetDate(Text_ID)

※ Text_ID는 텍스트 개체의 식별번호

예제 : var1 = Text_GetDate(Text1);

식별번호가 Text1인 텍스트 개체의 출력 봉의 날짜를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Text_GetTime

설명 : 지정한 텍스트 개체의 출력 봉의 시간을 리턴 합니다.

함수 : Text_GetTime(Text_ID)

※ Text_ID는 텍스트 개체의 식별번호

예제 : var1 = Text_GetTime(Text1);

식별번호가 Text1인 텍스트 개체의 출력 봉의 시간을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Text_GetValue

설명 : 지정한 텍스트 개체의 출력위치값(Y축)을 리턴 합니다.

함수 : Text_GetValue(Text_ID)

※ Text_ID는 텍스트 개체의 식별번호

예제 : var1 = Text_GetValue(Text1);

식별번호가 Text1인 텍스트 개체의 출력위치값(Y축)을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Text_GetHStyle

설명 : 지정한 텍스트 개체의 수평 정렬 위치를 리턴 합니다.

함수 : Text_GetHSytle(Text_ID)

※ Text_ID는 텍스트 개체의 식별번호

예제 : `var1 = Text_GetHStyle(Text1);`

식별번호가 Text1인 텍스트 개체의 수평 정렬 위치를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

리턴값이 0이면 왼쪽, 1이면 오른쪽, 2는 중앙입니다.

Text_GetVStyle

설명 : 지정한 텍스트 개체의 수직 정렬 위치를 리턴 합니다.

함수 : `Text_GetVStyle(Text_ID)`

※ Text_ID는 텍스트 개체의 식별번호

예제 : `var1 = Text_GetVStyle(Text1);`

식별번호가 Text1인 텍스트 개체의 수직 정렬 위치를 var6변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

리턴값이 0이면 상단, 1이면 하단, 2는 중앙입니다.

Text_GetString

설명 : 지정한 텍스트 개체의 출력문자열을 리턴합니다.

함수 : `Text_GetString(Text_ID)`

※ Text_ID는 텍스트 개체의 식별번호

예제 : `var : str(" ");`

`str = Text_GetString(Text1);`

식별번호가 Text1인 텍스트의 출력문자열을 문자형 변수인 str에 저장합니다.

리턴값은 문자형 변수에 저장해야 합니다.

8) 박스출력함수

Box_New

설명 : 기본차트 영역에 지정된 두개의 지점을 기준으로 박스 개체를 생성합니다.



시작지점과 끝지점만 지정해 그리므로
 정사각형이나 직사각형 형태로만 출력이 가능합니다.
 새로운 박스 개체가 생성되면 고유의 식별번호가 자동 부여되며
 아래와 같이 작성해 식별번호를 변수에 저장할수 있습니다.

```
var1 = Box_New(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)
```

함수 : `Box_New(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)`

- ※ StartDate는 박스 개체의 시작점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.
- ※ StartTime은 박스 개체의 시작점의 시간입니다. 시간은 HHMMSS 형식입니다.
- ※ StartValue는 박스 개체의 시작점의 가격(Y축)입니다.
- ※ EndtDate는 박스 개체의 끝점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.
- ※ EndTime은 박스 개체의 끝점의 시간입니다. 시간은 HHMMSS 형식입니다.
- ※ EndValue는 박스 개체의 끝점의 가격(Y축)입니다.

예제 : `var : stok(0),stod(0),Box1(0);`

```
stok = StochasticsK(20,10);
stod = StochasticsD(20,10,10);
```

#골드 발생

```
if crossup(stok,stod) Then
{
    value1 = sdate; #날짜
    value2 = stime; #시간
```

```
value3 = C; #종가

#이전 데드지점과 현재 골드지점을 기준으로 박스를 출력
Box1 = Box_New(Value4,value5,value6,value1,value2,value3);
#박스 색상지정
Box_Setcolor(box1,MAGENTA);
#박스 채움설정
Box_Setfill(box1,true,100,true);
}

#데드 발생
if CrossDown(stok,stod) Then
{
    value4 = sdate; #날짜
    value5 = stime; #시간
    value6 = C; #종가

    #이전 골드지점과 현재 데드지점을 기준으로 박스를 출력
    Box1 = Box_New(value1,value2,value3,value4,value5,value6);
    #박스 색상지정
    Box_Setcolor(box1,CYAN);
    #박스 채움설정
    Box_Setfill(box1,true,100,true);
}

Plot1(stok);
Plot2(stoD);
PlotBaseLine1(20);
PlotBaseLine2(30);
```



주의 : 박스 개체의 식별번호는 미리 지정된 양식에 따라 자동 부여됩니다.

식별번호를 사용하기 위해서는 반드시 숫자형 변수에 할당하고 그 변수를 사용해야 합니다.

Box_New_Self

설명 : 수식 적용 영역에 지정된 두개의 지점을 기준으로 박스 개체를 생성합니다.

Box_New함수는 기본차트에만 출력 되지만 Box_new_self함수를 이용하면 지표가 출력되는 영역이나 참조데이터 영역에 박스를 출력할 수 있습니다.

새로운 박스 개체가 생성되면 고유의 식별번호가 자동 부여되며

아래와 같이 작성해 식별번호를 변수에 저장할수 있습니다.

var1 = Box_New_Self(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)

함수 : Box_New_Self(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)

※ StartDate는 박스 개체의 시작점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ StartTime은 박스 개체의 시작점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ StartValue는 박스 개체의 시작점의 가격(Y축)입니다.

※ EndtDate는 박스 개체의 끝점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ EndTime은 박스 개체의 끝점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ EndValue는 박스 개체의 끝점의 가격(Y축)입니다.

예제 : var : stok(0),stod(0),Box1(0);

stok = StochasticsK(20,10);

stod = StochasticsD(20,10,10);

#골드 발생

if crossup(stok,stod) Then

{

value1 = sdate; #날짜

value2 = stime; #시간

value3 = stok; #스토케스틱값

#이전 데드지점과 현재 골드지점을 기준으로 박스를 출력

Box1 = Box_New_self(Value4,value5,value6,value1,value2,value3);

#박스 색상지정

Box_Setcolor(box1,MAGENTA);

#박스 채움설정

Box_Setfill(box1,true,100,true);

}

#데드 발생

if CrossDown(stok,stod) Then

{

value4 = sdate; #날짜

value5 = stime; #시간

value6 = stok; #스토케스틱값

#이전 골드지점과 현재 데드지점을 기준으로 박스를 출력

Box1 = Box_New_self(value1,value2,value3,value4,value5,value6);

#박스 색상지정

Box_Setcolor(box1,CYAN);

#박스 채움설정

Box_Setfill(box1,true,100,true);

}

Plot1(stok);

Plot2(stoD);

PlotBaseLine1(20);

PlotBaseLine2(30);



주의 : 박스 개체의 식별번호는 미리 지정된 양식에 따라 자동 부여됩니다.

식별번호를 사용하기 위해서는 반드시 숫자형 변수에 할당하고 그 변수를 사용해야 합니다.

Box_Delete

설명 : 지정한 박스 개체를 삭제합니다.

함수 : Box_Delete(Box_ID)

※ Box_ID는 박스 개체의 식별번호

예제 : Box_Delete(Box1);

식별번호가 Box1인 박스 개체를 삭제합니다.

Box_SetColor

설명 : 지정한 박스 개체의 색상을 설정합니다.

함수 : Box_SetColor(Box_ID,Color)

※ Box_ID는 박스 개체의 식별번호

※ Color는 색상예약어나 RGB함수로 지정할 수 있습니다.

예제 : Box_SetColor(Box1,RED);

식별번호가 Box1인 박스 개체의 색을 빨강색으로 설정합니다.

Box_SetStyle

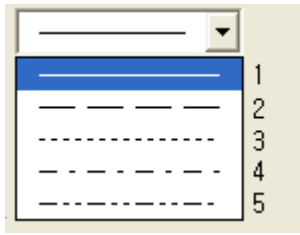
설명 : 지정한 박스 개체의 선의 형태를 설정합니다.

함수 : Box_SetStyle(Box_ID,Style)

※ Box_ID는 박스 개체의 식별번호

※ Style은 차트보조도구의 사각형 그리기의 속성화면에 제공하는 부분과 같습니다.

위에서부터 1~5번입니다.



예제 : `Box_SetStyle(Box1,1);`

식별번호가 Box1인 박스 개체의 형태를 1번 형태로 설정합니다.

Box_SetSize

설명 : 지정한 박스 개체의 굵기를 설정합니다.

함수 : `Box_SetSize(Box_ID,Size)`

※ Box_ID는 박스 개체의 식별번호

※ Size는 박스 개체의 굵기이며 1~5까지의 범위에서 정수로 지정 가능합니다.

※ 1번이 가장 얇고 5번이 가장 두껍습니다.

예제 : `Box_SetSize(Box1,1);`

식별번호가 Box1인 박스 개체의 굵기를 1로 설정합니다.

Box_SetFill

설명 : 지정한 박스 개체의 내부채움을 설정합니다.

내부채움은 `Box_SetColor`함수에서 지정한 색으로 채우게 됩니다.

함수 : `Box_SetFill(Box_ID,FillTF,TranNo,BackGround)`

※ Box_ID는 박스 개체의 식별번호

※ FillTF는 true로 설정하면 내부채움, false로 설정하면 내부채움을 하지 않습니다.

※ TranNo는 내부채움의 투명도입니다.

0~255 사이의 값으로 지정할 수 있고 255가 가장 진한색입니다.

생략가능하며 기본값은 50입니다.

※ BackGround는 true로 설정하면 봉뒤편 배경으로 배치되고, False로 지정하면 봉 위에 표시가 됩니다. 생략가능하며 기본값은 false입니다.

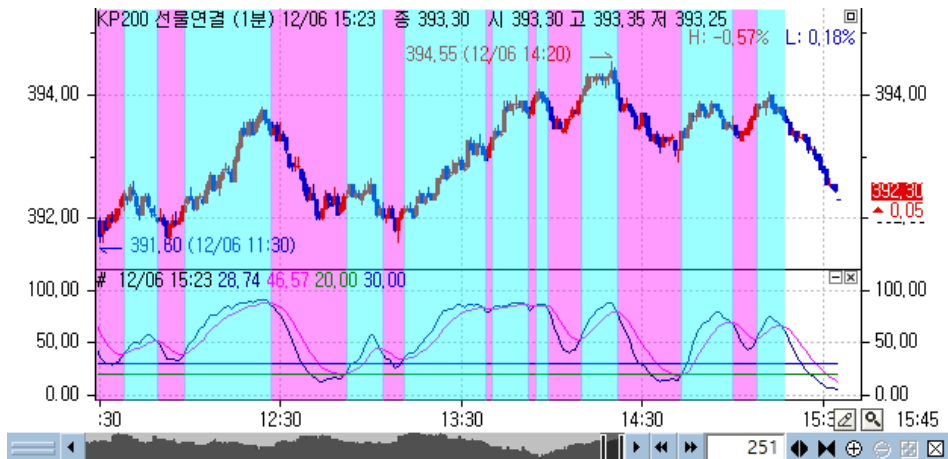
예제 : `Box_SetFill(Box1,true,100,true);`

식별번호가 Box1인 박스 개체를 내부채움을 하고 투명도는 100으로 봉뒤편에 표시

Box_SetExtFill

설명 : 지정한 박스 개체의 내부채움의 확장여부를 설정합니다.

확장하게 되면 박스가 차트 위아래 전체영역으로 채우게 됩니다.



함수 : `Box_SetExtFill(Box_ID,ExtTF)`

※ `Box_ID`는 박스 개체의 식별번호

※ `ExtTF`는 `true`로 설정하면 내부채움을 차트 위아래 전체영역으로 채우게 됩니다.

예제 : `Box_SetExtFill(Box1,true);`

식별번호가 `Box1`인 박스 개체를 내부채움을 확장

Box_SetBegin

설명 : 지정한 박스 개체의 시작점을 새로 설정합니다.

함수 : `Box_SetBegin(Box_ID,StartDate,StartTime,StartValue)`

※ `Box_ID`는 박스 개체의 식별번호이거나 식별번호를 뜻하는 수학적 표현입니다.

※ `StartDate`는 박스 개체의 시작점 날짜입니다. 날짜는 `YYYYMMDD` 형식입니다.

※ `StartTime`은 박스 개체의 시작점 시간입니다. 시간은 `HHMMSS` 형식입니다.

※ `StartValue`는 박스 개체의 시작점 가격(Y축)입니다.

예제 : `if Condition1 == true then`

`Box_SetBegin(Box1,sdate,stime,H);`

특정조건이 만족하면 식별번호가 `Box1`인 박스 개체의 시작점을

조건만족시점의 날짜, 시간, 지정한 값으로 변경합니다.

Box_SetEnd

설명 : 지정한 박스 개체의 끝점을 새로 설정합니다.

함수 : `Box_SetEnd(Box_ID,EndDate,EndTime,EndValue)`

※ `Box_ID`는 박스 개체의 식별번호

※ `EndtDate`는 박스 개체의 끝점 날짜입니다. 날짜는 `YYYYMMDD` 형식입니다.

※ `EndTime`은 박스 개체의 끝점 시간입니다. 시간은 `HHMMSS` 형식입니다.

※ EndValue는 박스 개체의 끝점 가격(Y축)입니다.

예제 : if Condition2 == true then

Box_SetEnd(Box1,sdate,stime,L);

특정조건이 만족하면 식별번호가 Box1인 박스 개체의 끝점을
조건만족시점의 날짜, 시간, 지정한 값으로 변경합니다.

Box_GetColor

설명 : 지정한 박스 개체의 색상을 리턴합니다.

함수 : Box_GetColor(Box_ID)

※ Box_ID는 박스 개체의 식별번호

예제 : var1 = Box_GetColor(Box1);

식별번호가 Box1인 박스 개체의 색을 var1에 저장합니다.

Box_GetStyle

설명 : 지정한 박스 개체의 선의 형태를 리턴합니다.

함수 : Box_GetStyle(Box_ID)

※ Box_ID는 박스 개체의 식별번호

예제 : Box_GetStyle(Box1);

식별번호가 Box1인 박스 개체의 선형태를 var1에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Box_GetSize

설명 : 지정한 박스 개체의 선굵기를 리턴합니다.

함수 : Box_GetSize(Box_ID)

※ Box_ID는 박스 개체의 식별번호

예제 : var1 = Box_GetSize(Box1);

식별번호가 Box1인 박스 개체의 선굵기를 var1에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Box_GetFill

설명 : 지정한 박스 개체의 내부채움을 설정을 리턴합니다.

지정한 박스 개체가 내부채움이 설정되어 있으면 1, 아니면 0이 리턴됩니다.

채움여부는 해당함수자체가 리턴하고

투명도나 백그라운드설정은 함수에 매개변수에 변수를 지정해서 값을 저장해 사용합니다.

함수 : Box_GetFill(Box_ID,oTranNo,oBackGround)

※ Box_ID는 박스 개체의 식별번호

※ oTranNo에 변수를 지정하면 해당 변수에 지정한 박스개체의 투명도값이 저장되며 생략이 가능합니다.

※ oBackGround에 변수를 지정하면 해당 변수에 지정한 박스개체의 배경배치설정이 저장되며 생략이 가능합니다. 배경배치설정이 되어 있으면 1, 아니면 0을 저장하게 됩니다.

예제 : var1 = box_GetFill(box1,Var2,Var3);

식별번호가 Box1인 박스 개체를 내부채움 여부는 var1에 저장하고

투명도는 var2, 배경배치설정은 var3에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Box_GetExtFill

설명 : 지정한 박스 개체의 내부채움의 확장여부를 리턴합니다.

확장되었으면1, 아니면 0을 리턴합니다.

함수 : Box_GetExitFill(Box_ID)

※ Box_ID는 박스 개체의 식별번호

예제 : var1 = Box_GetExtFill(Box1);

식별번호가 Box1인 박스 개체를 내부채움의 확장여부를 var1에 저장

리턴값은 수치형 변수에 저장해야 합니다.

Box_GetBeginDate

설명 : 지정한 박스 개체의 시작점의 날짜를 리턴합니다.

함수 : Box_GetBeginDate(Box_ID)

※ Box_ID는 박스 개체의 식별번호

※ YYYYMMDD형태로 리턴됩니다.

예제 : var1 = Box_GetBeginDate(Box1);

식별번호가 Box1인 박스 개체의 시작점의 날짜를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Box1인 박스 개체의 시작점의 날짜가 2012년 04월 18일이면

리턴값은 20120418입니다.

Box_GetBeginTime

설명 : 지정한 박스 개체의 시작점의 시간을 리턴합니다.

함수 : Box_GetBeginTime(Box_ID)

※ Box_ID는 박스 개체의 식별번호

※ HHMMSS형태로 리턴됩니다

예제 : var1 = Box_GetBeginTime(Box1);

식별번호가 Box1인 박스 개체의 시작점의 시간을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Box1인 박스 개체의 시작점의 시간이 10시라면 리턴값은 100000입니다.

Box_GetBeginVal

설명 : 지정한 박스 개체의 시작점(Y축)의 값을 리턴합니다.

함수 : Box_GetBeginVal(Box_ID)

※ Box_ID는 박스 개체의 식별번호

예제 : var1 = Box_GetBeginVal (Box1);

식별번호가 Box1인 박스 개체의 시작점(Y축)의 값을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Box_GetEndDate

설명 : 지정한 박스 개체의 끝점의 날짜를 리턴합니다.

함수 : Box_GetEndDate(Box_ID)

※ Box_ID는 박스 개체의 식별번호

※ YYYYMMDD형태로 리턴됩니다.

예제 : var1 = Box_GetEndDate(Box1);

식별번호가 Box1인 박스 개체의 끝점의 날짜를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Box1인 박스 개체의 끝점의 날짜가 2012년 04월 18일이면

리턴값은 20120418입니다.

Box_GetEndTime

설명 : 지정한 박스 개체의 끝점의 시간을 리턴합니다.

함수 : Box_GetEndTime(Box_ID)

※ Box_ID는 박스 개체의 식별번호

※ HHMMSS 형태로 리턴됩니다.

예제 : var1 = Box_GetEndTime(Box1);

식별번호가 Box1인 박스 개체의 끝점의 시간을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Box1인 박스 개체의 끝점의 시간이 12시라면 리턴값은 120000입니다.

Box_GetEndVal

설명 : 지정한 박스 개체의 끝점의 값(Y축)을 리턴합니다.

함수 : Box_GetEndVal(Box_ID)

※ Box_ID는 박스 개체의 식별번호

예제 : var1 = Box_GetEndVal(Box1);

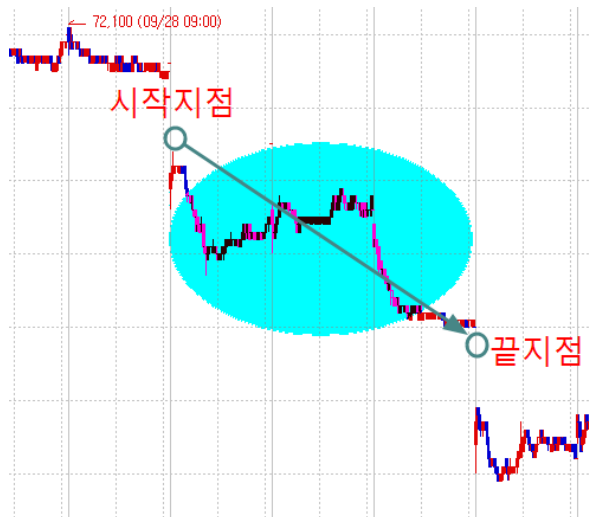
식별번호가 Box1인 박스 개체의 끝점의 값(Y축)을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 됩니다.

9) 타원출력함수

Circle_New

설명 : 기본차트 영역에 지정된 두개의 지점을 기준으로 타원 개체를 생성합니다.



새로운 타원 개체가 생성되면 고유의 식별번호가 자동 부여되며

아래와 같이 작성해 식별번호를 변수에 저장할수 있습니다.

var1 = Circle_New(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)

함수 : Circle_New(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)

※ StartDate는 타원 개체의 시작점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ StartTime은 타원 개체의 시작점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ StartValue는 타원 개체의 시작점의 가격(Y축)입니다.

※ EndtDate는 타원 개체의 끝점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ EndTime은 타원 개체의 끝점의 시간입니다. 시간은 HHMMSS 형식입니다.

※ EndValue는 타원 개체의 끝점의 가격(Y축)입니다.

예제 : var : stok(0),stod(0),Circle1(0);

stok = StochasticsK(20,10);

stod = StochasticsD(20,10,10);

#골드 발생

if crossup(stok,stod) Then

{

 value1 = sdate; #날짜

 value2 = stime; #시간

 value3 = C; #종가

 #이전 데드지점과 현재 골드지점을 기준으로 타원을 출력

 Circle1 = Circle_New(Value4,value5,value6,value1,value2,value3);

 #타원 색상 지정

 Circle_Setcolor(Circle1,MAGENTA);

 #타원 채움 설정

 Circle_setfill(Circle1,true,100,true);

}

#데드 발생

if CrossDown(stok,stod) Then

{

 value4 = sdate; #날짜

 value5 = stime; #시간

 value6 = C; #종가

 #이전 골드지점과 현재 데드지점을 기준으로 타원을 출력

 Circle1 = Circle_New(value1,value2,value3,value4,value5,value6);

 #타원 색상 지정

 Circle_Setcolor(Circle1,CYAN);

 #타원 채움 설정

 Circle_Setfill(Circle1,true,100,true);

}

```
Plot1(stok);
Plot2(stod);
PlotBaseLine1(20);
PlotBaseLine2(30);
```



주의 : 타원 개체의 식별번호는 미리 지정된 양식에 따라 자동 부여됩니다.

식별번호를 사용하기 위해서는 반드시 숫자형 변수에 할당하고 그 변수를 사용해야 합니다.

Circle_New_Self

설명 : 주식 적용 영역에 지정된 두개의 지점을 기준으로 타원 개체를 생성합니다.

Circle_New함수는 기본차트에만 출력 되지만 Circle_new_self함수를 이용하면 지표가 출력되는 영역이나 참조데이터 영역에 타원을 출력할 수 있습니다.

새로운 타원 개체가 생성되면 고유의 식별번호가 자동 부여되며

아래와 같이 작성해 식별번호를 변수에 저장할수 있습니다.

```
var1 = Circle_New_Self(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)
```

함수 : Circle_New_Self(StartDate,StartTime,StartValue,EndDate,EndTime,EndValue)

- ※ StartDate는 타원 개체의 시작점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.
- ※ StartTime은 타원 개체의 시작점의 시간입니다. 시간은 HHMMSS 형식입니다.
- ※ StartValue는 타원 개체의 시작점의 가격(Y축)입니다.
- ※ EndtDate는 타원 개체의 끝점의 날짜입니다. 날짜는 YYYYMMDD 형식입니다.
- ※ EndTime은 타원 개체의 끝점의 시간입니다. 시간은 HHMMSS 형식입니다.
- ※ EndValue는 타원 개체의 끝점의 가격(Y축)입니다.

예제 : var : stok(0),stod(0),Circle1(0);

```
stok = StochasticsK(20,10);
stod = StochasticsD(20,10,10);

#골드 발생
if crossup(stok,stod) Then
{
    value1 = sdate; #날짜
    value2 = stime; #시간
    value3 = stok; #스토캐스틱값

    #이전 데드지점과 현재 골드지점을 기준으로 타원을 출력
    Circle1 = Circle_New_self(Value4,value5,value6,value1,value2,value3);
    Circle_Setcolor(Circle1,MAGENTA);
    Circle_setfill(Circle1,true,100,true);
}

#데드 발생
if CrossDown(stok,stod) Then
{
    value4 = sdate; #날짜
    value5 = stime; #시간
    value6 = stok; #스토캐스틱값

    #이전 골드지점과 현재 데드지점을 기준으로 타원을 출력
    Circle1 = Circle_New_self(value1,value2,value3,value4,value5,value6);
    Circle_Setcolor(Circle1,CYAN);
    Circle_Setfill(Circle1,true,100,true);
}

Plot1(stok);
Plot2(stoD);
PlotBaseLine1(20);
PlotBaseLine2(30);
```



Circle_Delete

설명 : 지정한 타원 개체를 삭제합니다.

함수 : Circle_Delete(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

예제 : Circle_Delete(Circle1);

식별번호가 Circle1인 타원 개체를 삭제합니다.

Circle_SetColor

설명 : 지정한 타원 개체의 색상을 설정합니다.

함수 : Circle_SetColor(Circle_ID,Color)

※ Circle_ID는 타원 개체의 식별번호

※ Color는 색상예약어나 RGB함수로 지정할 수 있습니다.

예제 : Circle_SetColor(Circle1,RED);

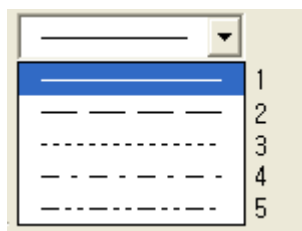
식별번호가 Circle1인 타원 개체의 색을 빨강색으로 설정합니다.

Circle_SetStyle

설명 : 지정한 타원 개체의 선의 형태를 설정합니다.

함수 : Circle_SetStyle(Circle_ID,Style)

※ Circle_ID는 타원 개체의 식별번호



※ Style은 차트보조도구의 원 그리기의 속성화면에 제공하는 부분과 같습니다.
위에서부터 1~5번입니다.

예제 : Circle_SetStyle(Circle1,1);

식별번호가 Circle1인 타원 개체의 형태를 1번 형태로 설정합니다.

Circle_SetSize

설명 : 지정한 타원 개체의 굵기를 설정합니다.

함수 : Circle_SetSize(Circle_ID,Size)

※ Circle_ID는 타원 개체의 식별번호

※ Size는 타원 개체의 굵기이며 1~5까지의 범위에서 정수로 지정 가능합니다.

※ 1번이 가장 얇고 5번이 가장 두껍습니다.

예제 : Circle_SetSize(Circle1,1);

식별번호가 Circle1인 타원 개체의 굵기를 1로 설정합니다.

Circle_SetFill

설명 : 지정한 타원 개체의 내부채움,투명도,배경배치를 설정합니다.

내부채움은 Circle_SetColor함수에서 지정한 색으로 채우게 됩니다.

함수 : Circle_SetFill(Circle_ID,FillTF,TranNo,BackGround)

※ Circle_ID는 타원 개체의 식별번호

※ FillTF는 true로 설정하면 내부채움, false로 설정하면 내부채움을 하지 않습니다.

※ TranNo는 내부채움의 투명도입니다.

0~255 사이의 값으로 지정할 수 있고 255가 가장 진한색입니다.

생략가능하고 기본값은 50입니다.

※ BackGround는 true로 설정하면 봉뒤 배경으로 배치되고, False로 지정하면 봉 위에 표시가 됩니다. 생략가능하고 기본값은 false입니다.

예제 : Circle_SetFill(Circle1,true,100,true);

식별번호가 Circle1인 타원 개체를 내부채움을 하고 투명도는 100으로 봉뒤에 표시

Circle_SetBegin

설명 : 지정한 타원 개체의 시작점을 새로 설정합니다.

함수 : Circle_SetBegin(Circle_ID,StartDate,StartTime,StartValue)

※ Circle_ID는 타원 개체의 식별번호이거나 식별번호를 뜻하는 수학적 표현입니다.

※ StartDate는 타원 개체의 시작점 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ StartTime은 타원 개체의 시작점 시간입니다. 시간은 HHMMSS 형식입니다.

※ StartValue는 타원 개체의 시작점 가격(Y축)입니다.

예제 : if Condition1 == true then

Circle_SetBegin(Circle1,sdate,stime,L);

특정조건이 만족하면 식별번호가 Circle1인 타원 개체의 시작점을
조건만족시점의 날짜, 시간, 지정한 값으로 변경합니다.

Circle_SetEnd

설명 : 지정한 타원 개체의 끝점을 새로 설정합니다.

함수 : Circle_SetEnd(Circle_ID,EndDate,EndTime,EndValue)

※ Circle_ID는 타원 개체의 식별번호

※ EndtDate는 타원 개체의 끝점 날짜입니다. 날짜는 YYYYMMDD 형식입니다.

※ EndTime은 타원 개체의 끝점 시간입니다. 시간은 HHMMSS 형식입니다.

※ EndValue는 타원 개체의 끝점 가격(Y축)입니다.

예제 : if Condition2 == true then

Circle_SetEnd(Circle1,sdate,stime,L);

특정조건이 만족하면 식별번호가 Circle1인 타원 개체의 끝점을
조건만족시점의 날짜, 시간, 지정한 값으로 변경합니다.

Circle_GetColor

설명 : 지정한 타원 개체의 색상을 리턴합니다.

함수 : Circle_GetColor(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

예제 : var1 = Circle_GetColor(Circle1);

식별번호가 Circle1인 타원 개체의 색을 var1에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Circle_GetStyle

설명 : 지정한 타원 개체의 선의 형태를 리턴합니다.

함수 : Circle_GetStyle(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

예제 : var1 = Circle_SetStyle(Circle1);

식별번호가 Circle1인 타원 개체의 선형태를 var1에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Circle_GetSize

설명 : 지정한 타원 개체의 선굵기를 리턴합니다.

함수 : Circle_GetSize(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

예제 : var1 = Circle_GetSize(Circle1);

식별번호가 Circle1인 타원 개체의 선굵기를 var1에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Circle_GetFill

설명 : 지정한 타원 개체의 내부채움을 설정을 리턴합니다.

지정한 타원 개체가 내부채움이 설정되어 있으면 1, 아니면 0이 리턴됩니다.

채움여부는 해당함수자체가 리턴하고

투명도나 백그라운드설정은 함수에 매개변수에 변수를 지정해서 저장해서 사용합니다.

함수 : Circle_GetFill(Circle_ID,oTranNo,oBackGround)

※ Circle_ID는 타원 개체의 식별번호

※ oTranNo에 변수를 지정하면 해당 변수에 지정한 타원개체의 투명도값이 저장되며 생략가능합니다.

※ oBackGround에 변수를 지정하면 해당 변수에 지정한 타원개체의 배경배치 설정이 저장됩니다. 배경배치 설정이 되어 있으면 1, 아니면 0을 저장하게 되며 생략가능합니다.

예제 : var1 = Circle_GetFill(Circle1,Var2,Var3);

식별번호가 Circle1인 타원 개체를 내부채움은 var1에 저장하고

투명도는 var2, BackGround설정은 var3에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Circle_GetBeginDate

설명 : 지정한 타원 개체의 시작점의 날짜를 리턴합니다.

함수 : Circle_GetBeginDate(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

※ YYYYMMDD형태로 리턴됩니다.

예제 : var1 = Circle_GetBeginDate(Circle1);

식별번호가 Circle1인 타원 개체의 시작점의 날짜를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Circle1인 타원 개체의 시작점의 날짜가 2012년 04월 18일이면
리턴값은 20120418입니다.

Circle_GetBeginTime

설명 : 지정한 타원 개체의 시작점의 시간을 리턴합니다.

함수 : Circle_GetBeginTime(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

※ HHMMSS형태로 리턴됩니다

예제 : var1 = Circle_GetBeginTime(Circle1);

식별번호가 Circle1인 타원 개체의 시작점의 시간을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Circle1인 타원 개체의 시작점의 시간이 10시라면 리턴값은 100000입니다.

Circle_GetBeginVal

설명 : 지정한 타원 개체의 시작점(Y축)의 값을 리턴합니다.

함수 : Circle_GetBeginVal(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

예제 : var1 = Circle_GetBeginVal(Circle1);

식별번호가 Circle1인 타원 개체의 시작점(Y축)의 값을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

Circle_GetEndDate

설명 : 지정한 타원 개체의 끝점의 날짜를 리턴합니다.

함수 : Circle_GetEndDate(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

※ YYYYMMDD형태로 리턴됩니다.

예제 : var1 = Circle_GetEndDate(Circle1);

식별번호가 Circle1인 타원 개체의 끝점의 날짜를 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Circle1인 타원 개체의 끝점의 날짜가 2012년 04월 18일이면
리턴값은 20120418입니다.

Circle_GetEndTime

설명 : 지정한 타원 개체의 끝점의 시간을 리턴합니다.

함수 : Circle_GetEndTime(Circle_ID)

※ Circle_ID는 타원 개체의 식별번호

※ HHMMSS 형태로 리턴됩니다.

예제 : `var1 = Circle_GetEndTime(Circle1);`

식별번호가 Circle1인 타원 개체의 끝점의 시간을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 합니다.

식별번호가 Circle1인 타원 개체의 끝점의 시간이 12시라면 리턴값은 120000입니다.

Circle_GetEndVal

설명 : 지정한 타원 개체의 끝점의 값(Y축)을 리턴합니다.

함수 : `Circle_GetEndVal(Circle_ID)`

※ Circle_ID는 타원 개체의 식별번호

예제 : `var1 = Circle_GetEndVal(Circle1);`

식별번호가 Circle1인 타원 개체의 끝점의 값(Y축)을 var1변수에 저장합니다.

리턴값은 수치형 변수에 저장해야 됩니다.

10) 그리드출력함수

Grid_New

설명 : 새로운 그리드(표) 객체를 생성합니다.

생성된 그리드는 텍스트가 지정된 셀만 표시가 됩니다.

함수 : `Grid_New(Position,columns,Rows,BackColor,FrameColor, FrameWidth, BorderColor, BorderWidth)`

※ Position : 그리드(표) 출력 위치

그리드 출력위치는 차트화면을 9개 지점으로 구분해 제공됩니다.

9개 지점 중 택일해서 출력할 수 있습니다.

상단좌측 : 0 or Position_Top_Left

상단우측 : 1 or Position_Top_Right

상단중심 : 2 or Position_Top_Center

중앙좌측 : 3 or Position_Middle_Left

중앙우측 : 4 or Position_Middle_Right

중앙중심 : 5 or Position_Middle_Center

하단좌측 : 6 or Position_Bottom_Left

하단우측 : 7 or Position_Bottom_Right

하단중심 : 8 or Position_Bottom_Center

기본차트 위에 적용할 때 위치



별도영역에 적용할 때 위치



- ※ Columns : column 개수 (1~30)
- ※ Rows : row 개수 (1~30)
- ※ BackColor : 배경색상
- ※ FrameColor : 외곽선 색상
- ※ FrameWidth : 외곽선 두께(0 ~ 20)
- ※ BorderColor : 셀경계선 색상
- ※ BorderWidth: 셀경계선 두께(0 ~ 20)

예제 : `var1 = Grid_New(Position_Top_Right, 2, 3, Green,Black, 1, Black, 1);`

2X3 그리드(표)를 생성하고 식별번호는 `var1`에 저장합니다.

리턴값은 수치형 변수에 저장해야 됩니다.

0.0	1.0
0.1	1.1
0.2	1.2

위치는 화면 상단우측,

배경색은 초록색이고, 외곽선은 검정색, 굵기는 1이고,

셀경계선은 검정색, 굵기는 1입니다.

Grid_SetPosition

설명 : 지정한 그리드(표)의 표시위치를 설정합니다.

함수 : `Grid_SetPosition(GridID, Position)`

※ GridID: 그리드(표) 식별번호

※ Position : 그리드(표) 출력 위치

그리드 출력위치는 차트화면을 9개 지점으로 구분해 제공됩니다.

9개 지점 중 택일해서 출력할 수 있습니다.

상단좌측 : 0 or `Position_Top_Left`

상단우측 : 1 or `Position_Top_Right`

상단중심 : 2 or `Position_Top_Center`

중앙좌측 : 3 or `Position_Middle_Left`

중앙우측 : 4 or `Position_Middle_Right`

중앙중심 : 5 or `Position_Middle_Center`

하단좌측 : 6 or `Position_Bottom_Left`

하단우측 : 7 or `Position_Bottom_Right`

하단중심 : 8 or `Position_Bottom_Center`

기본차트 위에 적용할 때 위치



별도영역에 적용할 때 위치



예제 : `Grid_SetPosition(GridID1, Position_Middle_right);`

식별번호 GridID1인 그리드(표)의 위치를 중앙우측으로 지정합니다.

Grid_SetBackColor

설명 : 지정한 그리드(표)의 배경색을 설정합니다.

함수 : `Grid_SetBackColor(GridID, BackColor)`

※ GridID: 그리드(표) 식별번호

※ BackColor : 배경 색상

예제 : Grid_SetBackColor(GridID1,Pink);

식별번호 GridID1인 그리드(표)의 배경색을 분홍색으로 지정합니다.

0.0	1.0
0.1	1.1
0.2	1.2

Grid_SetFrameColor

설명 : 지정한 그리드(표)의 외곽선색을 설정합니다.

함수 : Grid_SetFrameColor(GridID, FrameColor)

※ GridID: 그리드(표) 식별번호

※ FrameColor: 외곽선 색상

예제 : Grid_SetFrameColor(GridID1,Red);

식별번호 GridID1인 그리드(표)의 외곽선색을 빨강색으로 지정합니다.

0.0	1.0
0.1	1.1
0.2	1.2

Grid_SetFrameWidth

설명 : 지정한 그리드(표)의 외곽선 두께를 설정합니다.

함수 : Grid_SetFrameWidth(GridID, FrameWidth)

※ GridID: 그리드(표) 식별번호

※ FrameWidth : 두께 (0 ~ 20)

예제 : Grid_SetFrameWidth(GridID1,3);

식별번호 GridID1인 그리드(표)의 외곽선의 두께를 3으로 지정합니다.

0.0	1.0
0.1	1.1
0.2	1.2

Grid_SetBorderColor

설명 : 지정한 그리드(표)의 셀경계선 색상을 설정합니다.

함수 : Grid_SetBorderColor(GridID, BorderColor)

※ GridID: 그리드(표) 식별번호

※ BorderColor: 셀경계선 색상

예제 : Grid_SetBorderColor(GridID1,Blue);

식별번호 GridID1인 그리드(표)의 셀경계선 색상을 파란색으로 지정합니다.

0,0	1,0
0,1	1,1
0,2	1,2

Grid_SetBorderWidth

설명 : 지정한 그리드(표)의 셀경계선 두께를 설정합니다.

함수 : Grid_SetBorderWidth(GridID, BorderWidth)

※ GridID: 그리드(표) 식별번호

※ BorderWidth : 두께 (0 ~ 20)

예제 : Grid_SetBorderWidth(GridID1,3);

식별번호 GridID1인 그리드(표)의 셀경계선의 두께를 3로 지정합니다.

0,0	1,0
0,1	1,1
0,2	1,2

Grid_SetTextBold

설명 : 지정한 그리드(표)의 텍스트를 굵게 설정합니다.

함수 : Grid_SetTextBold(GridID, TextBold)

※ GridID: 그리드(표) 식별번호

※ TextBold : true : 굵게, false : 기본

예제 : Grid_SetTextBold(GridID1, true);

식별번호 GridID1인 그리드(표)의 텍스트를 굵게 설정합니다.

0,0	1,0
0,1	1,1
0,2	1,2

Grid_SetTextSize

설명 : 지정한 그리드(표)의 텍스트 크기를 설정합니다.

함수 : Grid_SetTextSize(GridID, TextSize)

※ GridID: 그리드(표) 식별번호

※ TextSize: 텍스트 크기 (0~50), 0은 기본크기

예제 : Grid_SetTextSize(GridID1, 20);

식별번호 GridID1인 그리드(표)의 텍스트의 크기를 20으로 설정합니다.

0,0	1,0
0,1	1,1
0,2	1,2

Grid_SetTransparency

설명 : 지정한 그리드(표)의 투명도를 설정합니다.

함수 : Grid_SetTransparency(GridID,Transparency)

※ GridID: 그리드(표) 식별번호

※ Transparency : 투명도(0~255), 값이 낮을수록 투명

예제 : Grid_SetTransparency(GridID1,100);

식별번호 GridID1인 그리드(표)의 투명도를 100으로 설정합니다.

0,0	1,0
0,1	1,1
0,2	1,2

Grid_Cell

설명 : 지정한 셀에 텍스트, 높이, 넓이, 텍스트색상, 배경색을 일괄 설정합니다.

함수 : Grid_Cell(GridID, Column, Row, Text, Width, Height, TextColor, BackColor)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ Text : 텍스트

※ Width : 셀넓이

※ Height : 셀높이

※ TextColor : 텍스트 색상

※ BackColor : 배경 색상

예제 : Grid_Cell(GridID1,0,0,"당일고가",0,0,Black,Yellow);

식별번호 GridID1인 그리드(표)의 셀(1,0)에 "당일고가" 텍스트를 표시.

셀 높이나 넓이는 글자크기에 따라 자동조절, 텍스트색상은 검정, 셀 배경은 노란색으로 지정

당일고가	1,0
0,1	1,1
0,2	1,2

Grid_CellSetBackColor

설명 : 지정한 셀에 배경색을 설정합니다.

함수 : Grid_CellSetBackColor(GridID, Column, Row, BackColor)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ BackColor : 배경 색상

예제 : Grid_CellSetBackColor(GridID1,1,0,Yellow);

식별번호 GridID1인 그리드(표)의 셀(1,0)의 배경색을 노란색으로 지정합니다.

당일고가	1.0
0.1	1.1
0.2	1.2

Grid_CellSetText

설명 : 지정한 셀에 텍스트를 지정합니다.

함수 : Grid_CellSetText(GridID, Column, Row, Text)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ Text : 텍스트, 숫자는 numtostr함수로 문자로 변환후 지정해야함.

예제 : Grid_CellSetText(GridID1,1,0,NumToStr(DayHigh(0),2));

식별번호 GridID1인 그리드(표)의 셀(1,0)에 당일고가값을 문자로 변경해 표시합니다.

당일고가	310.80
0.1	1.1
0.2	1.2

Grid_CellSetHeight

설명 : 지정한 셀의 높이를 설정합니다.

함수 : Grid_CellSetHeight(GridID, Column, Row, Height)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

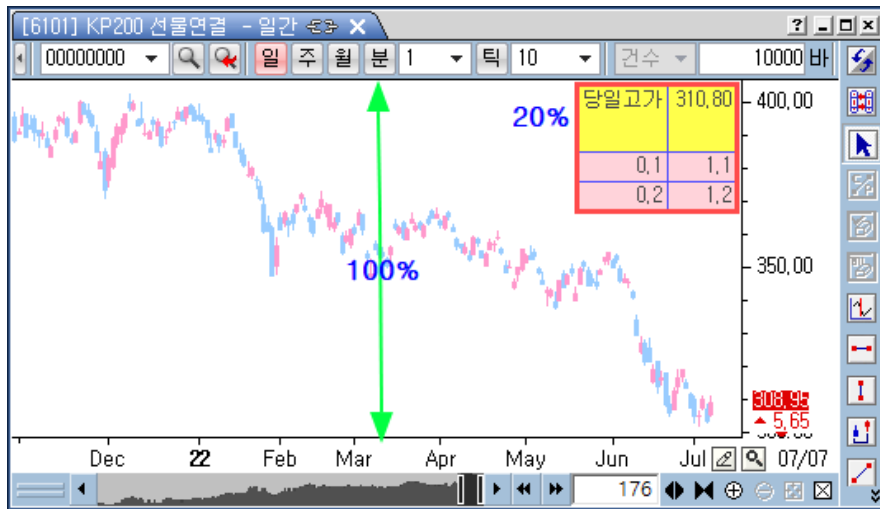
※ Height: 셀 높이.

0%~100% 사이에서 지정할 수 있고 0은 텍스트의 크기에 따라 자동조절.

적용영역의 상하단 폭을 기준으로 지정한 %만큼 셀 높이가 지정됩니다.

예제 : Grid_CellSetHeight(GridID1,1,0,20);

식별번호 GridID1인 그리드(표)의 셀(1,0)의 높이를 20%로 지정합니다.



Grid_CellSetWidth

설명 : 지정한 셀의 넓이를 설정합니다.

함수 : Grid_CellSetWidth(GridID, Column, Row, Width)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ Width 셀 넓이.

0%~100% 사이에서 지정할 수 있고 0은 텍스트의 크기에 따라 자동조절.

적용영역의 좌우 폭을 기준으로 지정한 %만큼 셀 넓이가 지정됩니다.

예제 : Grid_CellSetWidth (GridID1,1,0,15);

식별번호 GridID1인 그리드(표)의 셀(1,0)의 높이를 15%로 지정합니다.



Grid_CellSetTextBold

설명 : 지정한 셀의 텍스트를 굵게 지정합니다.

함수 : Grid_CellSetTextBold(GridID, Column, Row, TextBold)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ TextBold : true : 굵게, false : 기본

예제 : Grid_CellSetTextBold(GridID1, 0, 0, true);

식별번호 GridID1인 그리드(표)의 셀(0,0)의 텍스트를 굵게 표시합니다.

당일고가	310,80
0,1	1,1
0,2	1,2

Grid_CellSetTextColor

설명 : 지정한 셀의 텍스트 색상을 지정합니다.

함수 : Grid_CellSetTextColor(GridID, Column, Row, TextColor)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ TextColor : 텍스트색상

예제 : Grid_CellSetTextColor(GridID1, 0, 0, Red);

식별번호 GridID1인 그리드(표)의 셀(0,0)의 텍스트 색상을 빨강색으로 지정합니다.

당일고가	310,80
0,1	1,1
0,2	1,2

Grid_CellSetTextSize

설명 : 지정한 셀의 텍스트 크기를 지정합니다.

함수 : Grid_CellSetTextSize(GridID, Column, Row, TextSize)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ TextSize: 텍스트 크기(0~50), 0은 기본크기

예제 : Grid_CellSetTextSize(GridID1, 0, 0, Text_Font_Large);

식별번호 GridID1인 그리드(표)의 셀(0,0)의 텍스트를 크게 표시합니다.

당일고가	310,80
0,1	1,1
0,2	1,2

Grid_CellSetTextHAlign

설명 : 지정한 셀의 텍스트 수평정렬을 설정합니다.

함수 : Grid_CellSetTextHAlign(GridID, Column, Row, TextHAlign)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ TextHAlign: 텍스트 수평정렬

좌측 : 0 or Text_Align_Left

우측 : 1 or Text_Align_Right

중앙 : 2 or Text_Align_Center

예제 : Grid_CellSetTextHAlign(GridID1, 0, 0, Text_Align_Left);

식별번호 GridID1인 그리드(표)의 셀(0,0)의 텍스트 수평정렬을 좌측으로 설정합니다.

당일고가	310,80
0,1	1,1
0,2	1,2

Grid_CellSetTextVAlign

설명 : 지정한 셀의 텍스트 수직정렬을 설정합니다.

함수 : Grid_CellSetTextVAlign(GridID, Column, Row, TextVAlign)

※ GridID: 그리드(표) 식별번호

※ Column : Column 위치

※ Row : Row 위치

※ TextVAlign: 텍스트 수직정렬

상단 : 0 or Text_Align_Top

하단 : 1 or Text_Align_Bottom

중앙 : 2 or Text_Align_Center

예제 : Grid_CellSetTextVAlign(GridID1, 0, 0, Text_Align_Center);

식별번호 GridID1인 그리드(표)의 셀(0,0)의 텍스트 수직정렬을 중앙으로 설정합니다.

당일고가	310.80
0.1	1.1
0.2	1.2

Grid_Clear

설명 : 지정한 셀영역을 제거합니다.

함수 : Grid_Clear(GridID, StartColumn, StartRow, EndColumn, EndRow)

- ※ GridID: 그리드(표) 식별번호
- ※ StartColumn : 시작 Column 위치
- ※ StartRow : 시작 Row 위치
- ※ EndColumn : 종료 Column 위치
- ※ EndRow : 종료 Row 위치

예제 : Grid_Clear(GridID1,0,2,1,2);

식별번호 GridID1인 그리드(표)의 셀(0,2)~셀(1,2) 영역을 모두 제거합니다.

당일고가	310.80
0.1	1.1

Grid_MergeCells

설명 : 지정한 셀영역을 하나의 셀로 병합합니다.

이미 병합된 셀을 다시 병합하면 오류처리 됩니다.

함수 : Grid_MergeCells(GridID, StartColumn, StartRow, EndColumn, EndRow)

- ※ GridID: 그리드(표) 식별번호
- ※ StartColumn : 시작 Column 위치
- ※ StartRow : 시작 Row 위치
- ※ EndColumn : 종료 Column 위치
- ※ EndRow : 종료 Row 위치

예제 : Grid_MergeCells(GridID1,0,1,1,1);

식별번호 GridID1인 그리드(표)의 셀(0,1)~셀(1,1) 영역을 병합합니다.

당일고가	310.80
0.1	

Grid_Delete

설명 : 지정한 그리드(표)를 삭제합니다.

한번 삭제된 그리드는 이후로는 생성이 안됩니다.

함수 : Grid_Delete(GridID)

※ GridID: 그리드(표) 식별번호

예제 : Grid_Delete(GridID1)

식별번호 GridID1인 그리드(표)를 삭제합니다.

11) 잔고함수

잔고관련 함수는 사용자 PC의 “가원장” 기능을 이용하는 것으로서 주문 접수/체결 응답이 실제 원장과 가원장에 도달하고 처리되는 시차와 데이터의 유실 등으로 인해 차이가 발생할 수 있습니다.

사용자는 반드시 적절한 시점에 잔고조회 등을 통해 가원장의 상태를 원장의 상태와 동기화 될 수 있도록 조치 하여야 합니다.(기타 다른 매체나 단말을 통한 주문 및 입 출금 등 다른 요인도 존재합니다.)이런 현상에 대한 손실은 전적으로 사용자에게 귀속되며 회사는 이에 대한 보상이나 책임이 없습니다.(사용자는 반드시 여기에 동의 해야 만 사용할 수 있습니다.)

GetAccount

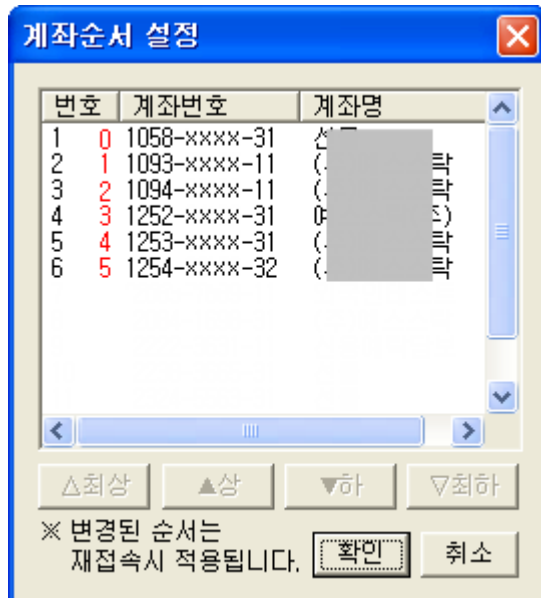
설명 : 계좌목록 의 계좌 중 지정한 순번의 계좌번호를 리턴

※ 계좌번호의 순번은 계좌순서 설정화면에서 위에서 아래로 0으로 해서 1씩 증가합니다.

※ 변수에 할당할 경우에는 문자열변수에 할당해 사용해야 합니다.

함수 : GetAccount(Accountindex)

예제 : GetAccount(1) → 계좌순서가 아래와 같다면 1093-xxxx-11



GetAccountStatus

설명 : 가원장 구축상태를 리턴(구축 :1, 미구축 : 0)

함수 : GetAccountStatus(계좌번호)

GetAccountType

설명 : 계좌종류, (1:위탁, 2:저축, 3:선/옵선)

함수 : GetAccountType(계좌번호)

예제 : var : AccountNum(""), AccountType(0);

AccountNum = getAccount(0);

AccountType = GetAccountType(AccountNum); → 계좌순번이 아래와 같다면 3

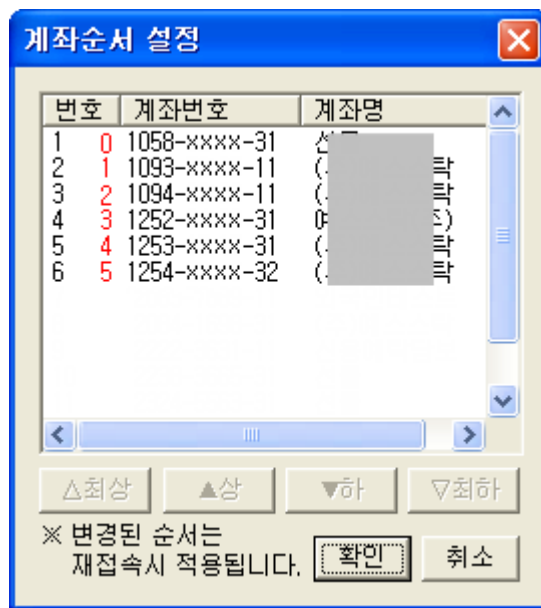


GetNumAccounts

설명 : 보유계좌의 총갯수

함수 : GetNumAccounts

예제 : value1 = GetNumAccounts; → 보유 계좌가 아래와 같다면 6



GetNumPositions

설명 : 지정한 계좌의 보유 총 종목수

함수 : GetNumPositions(계좌번호)

예제 : var : AccountNum(""), NumPositions(0);

AccountNum = getAccount(0);

NumPositions = GetNumPositions(AccountNum);

아래 그림과 같다면 NumPositions값은 14

번호	계좌번호	계좌명
1	0 1058-xxxx-31	선
2	1 1093-xxxx-11	(
3	2 1094-xxxx-11	(
4	3 1252-xxxx-31	0
5	4 1253-xxxx-31	(
6	5 1254-xxxx-32	(

계좌	1058-xxxx-31	선물	****	조회	다음
평가금액	45,942,000	4,680,000	60,982,000	-17,741,000	93,863,000
평가손익	45,942,000	4,680,000	22,772,500	4,312,900	77,707,400
손익률	1.00	0.25	59.60	19.56	1.19

종목코드	종목명	구분	수량	평균단가	현재가	평가손익	손익률
101G9000	KP200 F 1209	매수	38	242.23	244.65	45,942,000	1.00
101G0000	KP200 F 1212	매도	15	248.12	247.50	4,680,000	0.25
201G5272	C 201205 272.5	매수	10	0.41	0.01	-405,000	-97.59
201G6242	C 1206 242.5	매수	2	4.13	4.40	53,800	6.51
201G6245	C 1206 245.0	매수	1	3.83	3.05	-77,500	-20.26
201G6247	C 1206 247.5	매수	1	2.51	2.02	-49,000	-19.52
201G6255	C 1206 255.0	매도	9	5.25	0.41	4,356,000	92.19
201G6257	C 1206 257.5	매도	3	3.96	0.22	1,122,900	94.45
201G9245	C 1209 245.0	매도	3	10.55	11.40	-1,275,000	-8.06
301G6237	P 1206 237.5	매도	1	3.15	2.06	109,000	34.60
301G6245	P 1206 245.0	매수	1	4.75	4.80	5,000	1.05
301G6257	P 1206 257.5	매수	4	6.10	14.45	3,340,000	136.89
301G6270	P 1206 270.0	매수	19	15.89	26.50	20,155,200	66.75
301G7240	P 1207 240.0	매수	1	6.45	5.95	-250,000	-7.75

GetOpenOrderInitialMargin

설명 : 지정한 계좌의 위탁증거금

함수 : GetOpenOrderInitialMargin(계좌번호)

예제 : var : Account(""), InitialMargin(0);

```
Account = getAccount(0);
InitialMargin = GetOpenOrderInitialMargin(Account);
# 0순위 계좌의 위탁증거금
```

GetPositionSymbol

설명 : 지정한 계좌의 종목들 중에서 지정한 순번의 종목코드(단축코드)를 리턴

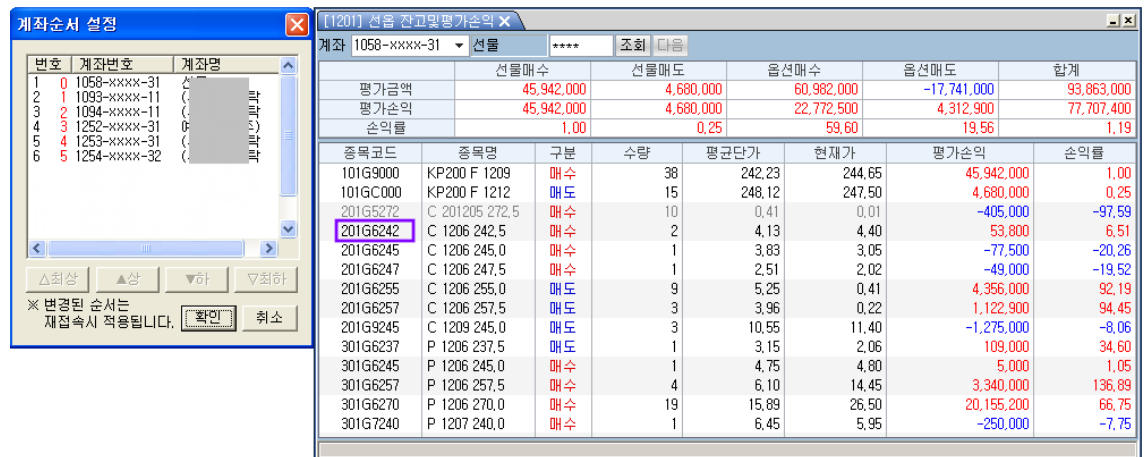
함수 : GetPositionSymbol(종목순번,계좌번호)

예제 : var : AccountNum(""),Symbol("");

```
AccountNum = getAccount(0);
```

```
Symbol = GetPositionSymbol(3,AccountNum);
```

#0순위 계좌의 3순위 종목의 종목코드



The screenshot shows a trading software window titled "[1201] 선물 잔고및평가손익 X". It displays account information for account number "1058-XXXX-31" and a list of positions. The left pane shows a list of positions with columns for rank, account number, and position name. The right pane shows a detailed table of positions with columns for position code, position name, division, quantity, average price, current price, average profit/loss, and profit/loss.

종목코드	종목명	구분	수량	평균단가	현재가	평가손익	손익률
101G9000	KP200 F 1209	매수	38	242.23	244.65	45,942,000	1.00
101GC000	KP200 F 1212	매도	15	248.12	247.50	4,680,000	0.25
201G5272	C 201205 272.5	매수	10	0.41	0.01	-405,000	-97.59
201G6242	C 1206 242.5	매수	2	4.13	4.40	53,800	6.51
201G6245	C 1206 245.0	매수	1	3.83	3.05	-77,500	-20.26
201G6247	C 1206 247.5	매수	1	2.51	2.02	-49,000	-19.52
201G6255	C 1206 255.0	매도	9	5.25	0.41	4,356,000	92.19
201G6257	C 1206 257.5	매도	3	3.96	0.22	1,122,900	94.45
201G9245	C 1209 245.0	매도	3	10.55	11.40	-1,275,000	-8.06
301G6237	P 1206 237.5	매도	1	3.15	2.06	109,000	34.60
301G6245	P 1206 245.0	매수	1	4.75	4.80	5,000	1.05
301G6257	P 1206 257.5	매수	4	6.10	14.45	3,340,000	136.89
301G6270	P 1206 270.0	매수	19	15.89	26.50	20,155,200	66.75
301G7240	P 1207 240.0	매수	1	6.45	5.95	-250,000	-7.75

GetPositionAveragePrice

설명 : 지정한 계좌의 지정한 종목의 평균가

함수 : GetPositionAveragePrice(종목코드,계좌번호,매수/매도구분)

※ 매수/매도구분은 생략가능합니다.(해외 종목에 필요한 항목입니다.)

예제 : var : AccountNum(""),Symbol(""),AveragePrice(0);

```
AccountNum = getAccount(0);
```

```
Symbol = GetPositionSymbol(4,AccountNum);
```

```
AveragePrice = GetPositionAveragePrice(Symbol,AccountNum);
```

0순위 계좌의 4순위 종목의 평균가

계좌순서 설정

번호	계좌번호	계좌명
1	0 1058-xxxx-31	선물
2	1 1093-xxxx-11	선물
3	2 1094-xxxx-11	선물
4	3 1252-xxxx-31	선물
5	4 1253-xxxx-31	선물
6	5 1254-xxxx-32	선물

※ 변경된 순서는 재접속시 적용됩니다. **확인** **취소**

[1201] 선물 잔고및평가손익

계좌: 1058-xxxx-31 선물 ***** 조회 다음

		선물매수	선물매도	옵션매수	옵션매도	합계
평가금액		45,942,000	4,680,000	60,982,000	-17,741,000	93,863,000
평가손익		45,942,000	4,680,000	22,772,500	4,312,900	77,707,400
손익률		1.00	0.25	59.60	19.56	1.19

종목코드	종목명	구분	수량	평균단가	현재가	평가손익	손익률
101G9000	KP200 F 1209	매수	38	242.23	244.65	45,942,000	1.00
101G0000	KP200 F 1212	매도	15	248.12	247.50	4,680,000	0.25
201G5272	C 201205 272.5	매수	10	0.41	0.01	-405,000	-97.59
201G6242	C 1206 242.5	매수	2	4.13	4.40	53,800	6.51
201G6245	C 1206 245.0	매수	1	3.83	3.05	-77,500	-20.26
201G6247	C 1206 247.5	매수	1	2.51	2.02	-49,000	-19.52
201G6255	C 1206 255.0	매도	9	5.25	0.41	4,356,000	92.19
201G6257	C 1206 257.5	매도	3	3.96	0.22	1,122,900	94.45
201G9245	C 1209 245.0	매도	3	10.55	11.40	-1,275,000	-8.06
301G6237	P 1206 237.5	매도	1	3.15	2.06	109,000	34.60
301G6245	P 1206 245.0	매수	1	4.75	4.80	5,000	1.05
301G6257	P 1206 257.5	매수	4	6.10	14.45	3,340,000	136.89
301G6270	P 1206 270.0	매수	19	15.89	26.50	20,155,200	66.75
301G7240	P 1207 240.0	매수	1	6.45	5.95	-250,000	-7.75

GetPositionMarketValue

설명 : 지정한 계좌의 지정한 종목의 현재가

함수 : GetPositionMarketValue(종목코드, 계좌번호)

예제 : var : AccountNum(""),Symbol(""),MarketValue(0);

AccountNum = getAccount(0);

Symbol = GetPositionSymbol(5,AccountNum);

MarketValue = GetPositionMarketValue(Symbol,AccountNum);

0순위 계좌의 5순위 종목의 현재가

계좌순서 설정

번호	계좌번호	계좌명
1	0 1058-xxxx-31	선물
2	1 1093-xxxx-11	선물
3	2 1094-xxxx-11	선물
4	3 1252-xxxx-31	선물
5	4 1253-xxxx-31	선물
6	5 1254-xxxx-32	선물

※ 변경된 순서는 재접속시 적용됩니다. **확인** **취소**

[1201] 선물 잔고및평가손익

계좌: 1058-xxxx-31 선물 ***** 조회 다음

		선물매수	선물매도	옵션매수	옵션매도	합계
평가금액		45,942,000	4,680,000	60,982,000	-17,741,000	93,863,000
평가손익		45,942,000	4,680,000	22,772,500	4,312,900	77,707,400
손익률		1.00	0.25	59.60	19.56	1.19

종목코드	종목명	구분	수량	평균단가	현재가	평가손익	손익률
101G9000	KP200 F 1209	매수	38	242.23	244.65	45,942,000	1.00
101G0000	KP200 F 1212	매도	15	248.12	247.50	4,680,000	0.25
201G5272	C 201205 272.5	매수	10	0.41	0.01	-405,000	-97.59
201G6242	C 1206 242.5	매수	2	4.13	4.40	53,800	6.51
201G6245	C 1206 245.0	매수	1	3.83	3.05	-77,500	-20.26
201G6247	C 1206 247.5	매수	1	2.51	2.02	-49,000	-19.52
201G6255	C 1206 255.0	매도	9	5.25	0.41	4,356,000	92.19
201G6257	C 1206 257.5	매도	3	3.96	0.22	1,122,900	94.45
201G9245	C 1209 245.0	매도	3	10.55	11.40	-1,275,000	-8.06
301G6237	P 1206 237.5	매도	1	3.15	2.06	109,000	34.60
301G6245	P 1206 245.0	매수	1	4.75	4.80	5,000	1.05
301G6257	P 1206 257.5	매수	4	6.10	14.45	3,340,000	136.89
301G6270	P 1206 270.0	매수	19	15.89	26.50	20,155,200	66.75
301G7240	P 1207 240.0	매수	1	6.45	5.95	-250,000	-7.75

GetPositionOpenPL

설명 : 지정한 계좌의 지정한 종목의 평가손익

함수 : GetPositionOpenPL (종목코드,계좌번호,매수/매도 구분)

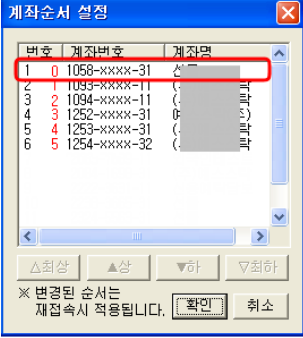
* 매수/매도구분은 생략가능합니다.(해의 종목에 필요한 항목입니다.)

예제 : var : AccountNum(""),Symbol(""),OpenPL(0);

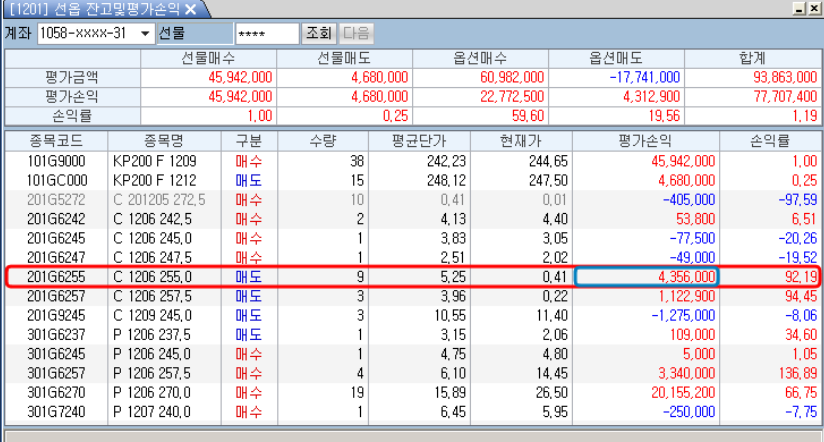
```

AccountNum = getAccount(0);
Symbol = GetPositionSymbol(6,AccountNum);
OpenPL = GetPositionOpenPL(Symbol,AccountNum);
# 0순위 계좌의 6순위 종목의 평가손익

```



번호	계좌번호	계좌명
1	0 1058-xxxx-31	선물
2	1 1093-xxxx-11	선물
3	2 1094-xxxx-11	선물
4	3 1252-xxxx-31	선물
5	4 1253-xxxx-31	선물
6	5 1254-xxxx-32	선물



계좌	선물매수	선물매도	옵션매수	옵션매도	합계
평가금액	45,942,000	4,680,000	60,982,000	-17,741,000	93,863,000
평가손익	45,942,000	4,680,000	22,772,500	4,312,900	77,707,400
손익률	1.00	0.25	59.60	19.56	1.19

종목코드	종목명	구분	수량	평균단가	현재가	평가손익	손익률
101G9000	KP200 F 1209	매수	38	242.23	244.65	45,942,000	1.00
101G0000	KP200 F 1212	매도	15	248.12	247.50	4,680,000	0.25
201G5272	C 201205 272.5	매수	10	0.41	0.01	-405,000	-97.59
201G6242	C 1206 242.5	매수	2	4.13	4.40	53,800	6.51
201G6245	C 1206 245.0	매수	1	3.83	3.05	-77,500	-20.26
201G6247	C 1206 247.5	매수	1	2.51	2.02	-49,000	-19.52
201G6255	C 1206 255.0	매도	9	5.25	0.41	4,356,000	92.19
201G6257	C 1206 257.5	매도	3	3.96	0.22	1,122,900	94.45
201G9245	C 1209 245.0	매도	3	10.55	11.40	-1,275,000	-8.06
301G6237	P 1206 237.5	매도	1	3.15	2.06	109,000	34.60
301G6245	P 1206 245.0	매수	1	4.75	4.80	5,000	1.05
301G6257	P 1206 257.5	매수	4	6.10	14.45	3,340,000	136.89
301G6270	P 1206 270.0	매수	19	15.89	26.50	20,155,200	66.75
301G7240	P 1207 240.0	매수	1	6.45	5.95	-250,000	-7.75

GetPositionQuantity

설명 : 지정한 계좌의 지정한 종목의 보유수량

함수 : GetPositionQuantity (종목코드,계좌번호,매수/매도 구분)

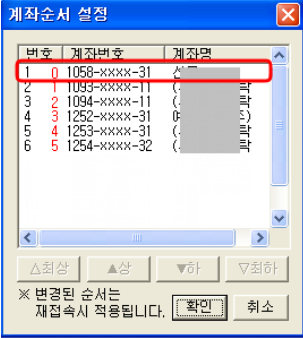
※ 매수/매도구분은 생략가능합니다.(해의 종목에 필요한 항목입니다.)

예제 : var : AccountNum(""),Symbol(""),Quantity(0);

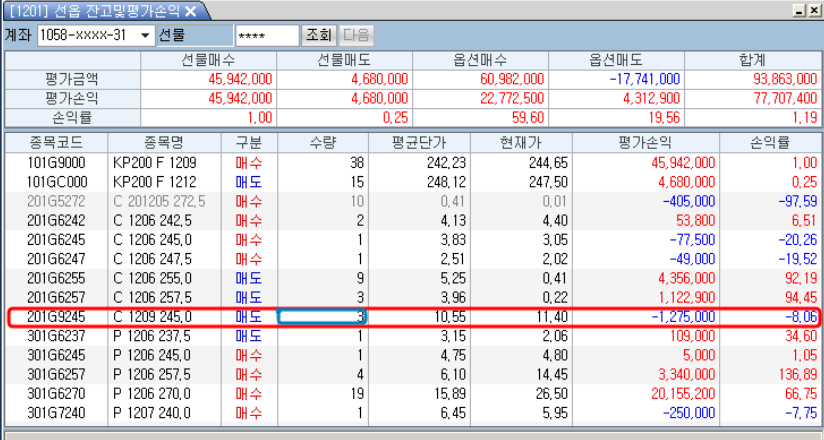
```

AccountNum = getAccount(0);
Symbol = GetPositionSymbol(4,AccountNum);
Quantity = GetPositionQuantity(Symbol,AccountNum);
# 0순위 계좌의 8순위 종목의 보유수량

```



번호	계좌번호	계좌명
1	0 1058-xxxx-31	선물
2	1 1093-xxxx-11	선물
3	2 1094-xxxx-11	선물
4	3 1252-xxxx-31	선물
5	4 1253-xxxx-31	선물
6	5 1254-xxxx-32	선물



계좌	선물매수	선물매도	옵션매수	옵션매도	합계
평가금액	45,942,000	4,680,000	60,982,000	-17,741,000	93,863,000
평가손익	45,942,000	4,680,000	22,772,500	4,312,900	77,707,400
손익률	1.00	0.25	59.60	19.56	1.19

종목코드	종목명	구분	수량	평균단가	현재가	평가손익	손익률
101G9000	KP200 F 1209	매수	38	242.23	244.65	45,942,000	1.00
101G0000	KP200 F 1212	매도	15	248.12	247.50	4,680,000	0.25
201G5272	C 201205 272.5	매수	10	0.41	0.01	-405,000	-97.59
201G6242	C 1206 242.5	매수	2	4.13	4.40	53,800	6.51
201G6245	C 1206 245.0	매수	1	3.83	3.05	-77,500	-20.26
201G6247	C 1206 247.5	매수	1	2.51	2.02	-49,000	-19.52
201G6255	C 1206 255.0	매도	9	5.25	0.41	4,356,000	92.19
201G6257	C 1206 257.5	매도	3	3.96	0.22	1,122,900	94.45
201G9245	C 1209 245.0	매도	3	10.55	11.40	-1,275,000	-8.06
301G6237	P 1206 237.5	매도	1	3.15	2.06	109,000	34.60
301G6245	P 1206 245.0	매수	1	4.75	4.80	5,000	1.05
301G6257	P 1206 257.5	매수	4	6.10	14.45	3,340,000	136.89
301G6270	P 1206 270.0	매수	19	15.89	26.50	20,155,200	66.75
301G7240	P 1207 240.0	매수	1	6.45	5.95	-250,000	-7.75

GetUnclearedDeposits

설명 : 지정한 계좌의 예수금, (선/오픈 경우 예탁총액, 위탁/저축인 경우 예수금)

함수 : GetUnclearedDeposits(계좌번호)

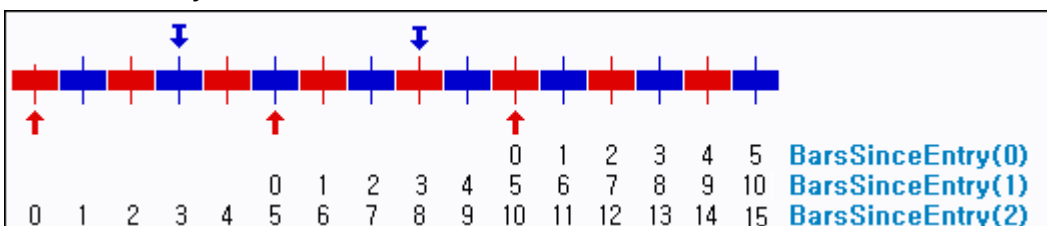
예제 : var : AccountNum(""),value(0);

AccountNum = getAccount(0); #계좌번호를 AccountNum에 저장

value = GetUnclearedDeposits(AccountNum); ##AccountNum으로 지정한 계좌의 예수금

12) 포지션함수

BarsSinceEntry



설명 : 진입 이후 경과된 봉 개수.

※ N은 진입 위치.진입위치는 현재를 기준으로 역순으로 지정.

※ 현재 진입은 0, 직전 진입 1,.....

※ (0)은 생략가능.

※ 신호가 발생한 봉을 0으로 하여 1씩 증가한다.

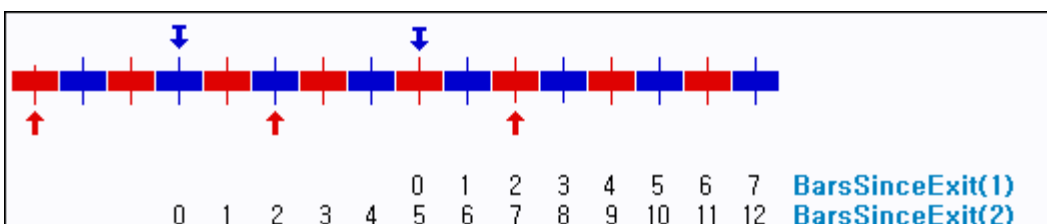
※ 피라미딩 상황에서는 포지션이 시작되는 첫 신호 이후 경과된 봉 수

함수 : BarsSinceEntry(N)

예제 : if BarsSinceEntry >= 10 then → 진입 이후 10개봉 경과

highest(H, BarsSinceEntry) → 진입 이후 최고가

BarsSinceExit



설명 : 청산 이후 경과된 봉 개수.

※ N은 청산 위치. 청산위치는 현재를 기준으로 역순으로 지정.

※ 최근 청산은 1, 직전 청산 2,.....

※ (0)이 사용될 수 없다.

함수 : BarsSinceExit(N)

예제 : if BarsSinceExit(1) >= 10 then → 청산 이후 10개봉 경과

Lowest(L, BarsSinceExit(1)) → 최근 청산 이후 최저가
highest(H, BarsSinceExit(1)) → 최근 청산 이후 최고가

CurrentContracts

설명 : 포지션에서 현재의 수량을 리턴.

함수 : CurrentContracts



- ① 신호 발생 후 CurrentContracts는 1
- ② 신호 발생 후 CurrentContracts는 3
- ③ 신호 발생 후 CurrentContracts는 5
- ④ 신호 발생 후 CurrentContracts는 7

CurrentEntries

설명 : 포지션에서 현재의 진입횟수를 리턴.

함수 : CurrentEntries



- 1 신호 발생 후 CurrentEntries는 1
- 2 신호 발생 후 CurrentEntries는 2
- ③ 신호 발생 후 CurrentEntries는 3
- ④ 신호 발생 후 CurrentEntries는 4

if CurrentEntries == 1 and condition1 == true then

Buy();

→ 최초 진입 후 첫번째 추가진입을 의미

AvgEntryPrice

설명 : 평균단가

※ 평균단가이므로 피라미딩시에만 의미가 있음.

※ 이전 거래 참조 안됨.

함수 : AvgEntryPrice

예제 : 만일 매수진입이 연속 발생해서

첫번째 매수의 진입가가 200.00 , 수량이 1 이고

두번째 매수의 진입가가 201.50, 수량이 2 이라면

평균단가는 $(200.00 \times 1 + 201.50 \times 2) / 3$ 과 같이 계산되어 201이다.

EntryDate

설명 : 진입날짜 리턴 함수

※ N은 진입 위치. 진입위치는 현재를 기준으로 역순으로 지정.

※ 최근 진입은 0, 직전 진입 1,.....

※ (0)은 생략가능

※ 피라미딩 상황에서는 포지션이 시작되는 첫진입의 날짜

함수 : EntryDate(N)

예제 : if sdate >= EntryDate+1 then → 진입이후 하루 이상 경과

EntryName

설명 : 진입명 리턴 함수

※ N은 진입 위치.진입위치는 현재를 기준으로 역순으로 지정.

※ 최근 진입은 0, 직전 진입 1,.....

※ (0)은 생략가능

※ 피라미딩 상황에서는 포지션이 시작되는 첫 진입의 이름

함수 : EntryName(N) == "진입명" or EntryName(N) != "진입명"

예제 : 일반적으로 진입별로 청산을 달리하고자 할 때와 같이

진입명칭을 확인해야 할 때 사용된다.

```

if crossup(ma(C,4),ma(c,9)) Then
    buy("b1");
if crossup(ma(C,9),ma(c,18)) Then
5    buy("b2");

if MarketPosition == 1 Then{
    if EntryName == "b1" Then
        ExitLong("bx1",AtStop,EntryPrice-1);
10    if EntryName == "b1" Then
        ExitLong("bx2",AtStop,EntryPrice-1);
}

```

IsEntryName

설명 : 진입명 리턴 함수

※ N은 진입 위치.진입위치는 현재를 기준으로 역순으로 지정.

※ 최근 진입은 0, 직전 진입 1,.....

※ (0)은 생략가능

※ 피라미딩 상황에서는 포지션이 시작되는 첫 진입의 이름

함수 : IsEntryName("진입명",N)

예제 : 일반적으로 진입별로 청산을 달리하고자 할 때와 같이

진입명칭을 확인해야 할 때 사용된다.

```

if crossup(ma(C,4),ma(C,9)) Then
    buy("b1");
if Crossup(ma(C,9),ma(C,18)) Then
5    buy("b2");

if MarketPosition == 1 Then{
    if IsEntryName("b1") == true Then
        exitlong("bx1",AtStop,EntryPrice-1);
10    if IsEntryName("b2") == true Then
        exitlong("bx2",AtStop,EntryPrice-2);
}

```

EntryPrice

설명 : 진입가격 리턴 함수

※ N은 진입 위치.진입위치는 현재를 기준으로 역순으로 지정.

※ 최근 진입은 0, 직전 진입 1,.....

※ (0)은 생략가능

※ 피라미딩 상황에서는 포지션이 시작되는 첫 진입의 이름

함수 : EntryPrice(N)

예제 : ExitLong("bx",atstop,entryprice-1); → 진입가에서 1포인트 하락시 청산

EntryTime

설명 : 진입시간 리턴 함수

※ N은 진입 위치.진입위치는 현재를 기준으로 역순으로 지정.

※ 최근 진입은 0, 직전 진입 1,.....

※ (0)은 생략가능

※ 피라미딩 상황에서는 포지션이 시작되는 첫 진입의 이름

함수 : EntryTime (N)

예제 : if TimeToMinutes(stime) <= TimeToMinutes(EntryTime)+30 Then

→ 진입 후 30분 이상 경과되지 않았다

ExitDate

설명 : 청산날짜 리턴 함수

※ N은 청산 위치. 청산위치는 현재를 기준으로 역순으로 지정.

※ 최근 청산은 1, 직전 청산 2,.....

※ (0)은 사용할 수 없음

함수 : ExitDate (N)

예제 : if sdate >= ExitDate(1)+ 1 then → 최근 청산 이후 하루 이상 경과

ExitName

설명 : 진입명 리턴 함수

※ N은 진입 위치.진입위치는 현재를 기준으로 역순으로 지정.

※ 최근 진입은 0, 직전 진입 1,.....

※ (0)은 생략가능

함수 : ExitName(N) == "청산명" or ExitName(N) != "청산명"

예제 : 일반적으로 진입별로 청산을 달리하고자 할 때와 같이 청산명칭을
확인해야 할 때 사용된다.

```

Condition1 = crossup(ema(C,5),ema(c,200));
Condition2 = CrossDown(ema(C,5),ema(c,200));

5 #청산이 손절이 아니면 Condition1로 진입
  if IsExitName("stoploss",1) == False and Condition1 == true Then
    buy("b1");

    #청산이 손절로 끝나면 직전 청산가에서 1포인트 상향시 진입
10 if ExitName(1)=="stoploss" and crossup(c,ExitPrice(1)+1) Then
    buy("b2");

    #청산1
    if Condition2 == True Then
15      exitlong("bx1");

    #손절
    SetStopLoss(1,PointStop);
    #당일청산
20 SetStopEndofday(144500);

```

IsExitName

설명 : 청산명 리턴 함수

※ N은 청산 위치. 청산위치는 현재를 기준으로 역순으로 지정.

※ 최근 청산은 1, 직전 청산 2,.....

※ (0)은 사용할 수 없음

함수 : IsExitName("청산명",N)

예제 : 일반적으로 직전 청산이 무엇인지를 확인해 이 후 진입을 달리하고자 할 때 사용된다.

```

Condition1 = crossup(ema(C,5),ema(c,200));
Condition2 = CrossDown(ema(C,5),ema(c,200));

5 #청산이 손절이 아니면 Condition1로 진입
  if IsExitName("stoploss",1) == False and Condition1 == true Then
    buy("b1");

    #청산이 손절로 끝나면 직전 청산가에서 1포인트 상향시 진입
10 if IsExitName("stoploss",1) == True and crossup(c,ExitPrice(1)+1) Then
    buy("b2");

    #청산1
    if Condition2 == True Then
15      exitlong("bx1");

    #손절
    SetStopLoss(1,PointStop);
    #당일청산
20 SetStopEndofday(144500);

```

ExitPrice

설명 : 청산가격 리턴 함수

※ N은 청산 위치. 청산위치는 현재를 기준으로 역순으로 지정.

※ 최근 청산은 1, 직전 청산 2,.....

※ (0)은 사용할 수 없음

함수 : ExitPrice(N)

예제 :

```
Condition1 = crossup(ema(C,5),ema(c,200));
Condition2 = CrossDown(ema(C,5),ema(c,200));

5 #청산이 손절이 아니면 Condition1로 진입
  if IsExitName("stoploss",1) == False and Condition1 == true Then
    buy("b1");

    #청산이 손절로 끝나면 직전 청산가에서 1포인트 상향시 진입
10 if IsExitName("stoploss",1) == True and crossup(c,ExitPrice(1)+1) Then
    buy("b2");

    #청산1
    if Condition2 == True Then
15     exitlong("bx1");

    #손절
    SetStopLoss(1,PointStop);
    #당일처안
20 SetStopEndofday(144500);
```

ExitTime

설명 : 청산시간 리턴 함수

※ N은 청산 위치. 청산위치는 현재를 기준으로 역순으로 지정.

※ 최근 청산은 1, 직전 청산 2,.....

※ (0)은 사용할 수 없음

함수 : ExitTime (N)

예제 : if TimeToMinutes(stime) >= TimeToMinutes(ExitTime(1))+30 Then

→ 청산 후 30분 이상 경과

MarketPosition

설명 : 포지션의 상태를 리턴

※ 매수포지션 1, 매도포지션 -1, 무포지션 0

※ N은 거래 위치. 포지션 위치는 현재를 기준으로 역순으로 지정.

※ 현재 거래 0, 직전 거래 1,.....

※ (0)은 생략가능.

함수 : MarketPosition(N)

예제 :

```

# 무포지션 상태일 때만 Condition1만족하면 매수진입
if MarketPosition == 0 and Condition1 == true Then
    buy();
5
# 매수포지션 상태일 때만 Condition1만족하면 매수진입
# 파라미딩 시 추가 매수분임을 의미
if MarketPosition == 1 and Condition1 == true Then
    buy();
10
# 매수포지션 상태가 아닐 때만 Condition1만족하면 매수진입
# 매도포지션 상태이거나 무포지션 상태일 때만 진입함을 의미
if MarketPosition != 1 and Condition1 == true Then
    buy();
15
# 무포지션 상태일 때만 Condition2 만족하면 매도진입
if MarketPosition == 0 and Condition2 == true Then
    Sell();
20
# 매도포지션 상태일 때만 Condition2 만족하면 매도진입
# 파라미딩 시 추가 매도분임을 의미
if MarketPosition == 1 and Condition2 == true Then
    Sell();
25
# 매도포지션 상태가 아닐 때만 Condition2 만족하면 매수진입
# 매수포지션 상태이거나 무포지션 상태일 때만 진입함을 의미
if MarketPosition != 1 and Condition2 == true Then
    buy();

```

MaxContracts

설명 : 포지션의 최대 누적 수량

- ※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.
- ※ 최근 거래는 0, 직전 거래 1,.....
- ※ (0)은 생략가능

함수 : MaxContracts(N)

예제 : if CurrentContracts < MaxContracts Then

- 현재 거래의 누적 수량이 최대 누적 수량보다 작다
- 일부 청산이 있었음을 의미

MaxEntries

설명 : 포지션의 최대 누적 진입횟수

- ※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.
- ※ 최근 거래는 0, 직전 거래 1,.....
- ※ (0)은 생략가능

함수 : MaxEntries (N)

예제 : if CurrentEntries < MaxEntries Then

→ 현재 거래의 누적 진입횟수가 최대 누적 진입횟수보다 작다

→ 일부 청산이 있었음을 의미

MaxPositionLoss

설명 : 포지션 최대 손실

※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.

※ 최근 거래는 0, 직전 거래 1,.....

※ (0)은 생략가능

함수 : MaxPositionLoss(N)

예제 : if MaxPositionLoss <= -1 Then

→ 포지션이 최대 손실이 -1포인트 이상이다.

※손익함수는 수량이 포함되어 계산된다.

선물에서

100에 1계약 매수진입 후 99까지 가격이 하락하였다면 MaxPositionLoss는 -1포인트.

100에 2계약 매수진입 후 99까지 가격이 하락하였다면 MaxPositionLoss는 -2포인트

위는 수수료와 슬리피지를 감안하지 않은 간단한 예시이며

시스템 트레이딩 설정창의 비용/수량 탭에서 수수료와 슬리피지를 설정하게 되면

진입중에는 진입수수료와 진입슬리피지가 포함되며 청산 후 청산수수료와 청산슬리피지가 반영되어 손익이 계산된다.

MaxPositionProfit

설명 : 포지션 최대 수익

※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.

※ 최근 거래는 0, 직전 거래 1,.....

※ (0)은 생략가능

함수 : MaxPositionProfit (N)

예제 : if MaxPositionProfit >= 2 Then

→ 포지션이 최대 수익이 2포인트 이상이다.

※손익함수는 수량을 포함하여 계산된다.

선물에서

100에 1계약 매수진입 후 101까지 가격이 상승하였다면 MaxPositionProfit은 1포인트.

100에 2계약 매수진입 후 101까지 가격이 상승하였다면 MaxPositionProfit는 2포인트

위는 수수료와 슬리피지를 감안하지 않은 간단한 예시이며

시스템 트레이딩 설정창의 비용/수량 탭에서 수수료와 슬리피지를 설정하게 되면

진입중에는 진입수수료와 진입슬리피지가 포함되며 청산 후에는 청산수수료와 청산슬리피지가 반영되어 손익이 계산된다.

OpenPositionProfit

설명 : 포지션의 손익

※ 이전 거래 참조 안됨

※ PositionProfit(0)과 같음

※ 현재 포지션의 값만 리턴(이전 거래의 손익이 필요하다면 PositionProfit 함수를 이용)

함수 : OpenPositionProfit

예제 : if OpenPositionProfit > 2 Then

→ 포지션이 수익이 2포인트 이상이다.

※손익함수는 수량이 포함되어 계산된다.

선물에서

100에 1계약 매수진입 후 현재 가격이 101이면 OpenPositionProfit은 1포인트.

100에 2계약 매수진입 후 현재 가격이 101이면 OpenPositionProfit은 2포인트

위는 수수료와 슬리피지를 감안하지 않은 간단한 예시이며

시스템 트레이딩 설정창의 비용/수량 탭에서 수수료와 슬리피지를 설정하게 되면

진입중에는 진입수수료와 진입슬리피지가 반영되어 손익이 계산된다.

PositionProfit

설명 : 포지션의 손익

※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.

※ 최근 거래는 0, 직전 거래 1,.....

※ (0)은 생략가능

함수 : PositionProfit (N)

예제 : 직전거래가 손실이면 수량 2개 수익이면 1개

```

if crossup(ma(C,5),ma(C,20)) Then
    buy("b1",OnClose,def,iff(PositionProfit>=0,1,2));
5 if CrossDown(ma(C,5),ma(C,20)) Then
    Sell("S",OnClose,def,iff(PositionProfit>=0,1,2));

```

당일 손익 계산

```

var : PLR(0),XCommission(0),XSlippage(0),OpenPL(0),dayPL(0),count(0);

XCommission = ((C*ExitCommission)/100)*CurrentContracts; #%설정
XSlippage = (ExitSlippage)*CurrentContracts; #Pt설정
5
PLR = 0;
count = 0;
for var1 = 1 to 10{
    if sdate == EntryDate(var1) Then{
10        count = count+1;
        PLR = PLR+PositionProfit(var1);
    }
}

15 if MarketPosition() == 0 Then{
    OpenPL = 0;
    dayPL = PLR;
}
Else{
20    OpenPL = (PositionProfit-(XCommission+XSlippage));
    dayPL = PLR+OpenPL;
}

if crossup(c,ma(c,20)) and dayPL < 1 Then
25    buy("b");

if CrossDown(c,ma(c,20)) Then{
    exitlong("s");
}
30
if MarketPosition() == 1 and dayPL >= 1 Then
    exitlong("XX");

```

※손익함수는 수량이 포함되어 계산된다.

선물에서

100에 1계약 매수진입 후 현재 가격이 101이면 PositionProfit은 1포인트.

100에 2계약 매수진입 후 현재 가격이 101이면 PositionProfit은 2포인트

위는 수수료와 슬리피지를 감안하지 않은 간단한 예시이며

시스템 트레이딩 설정창의 비용/수량 탭에서 수수료와 슬리피지를 설정하게 되면

진입중에는 진입수수료와 진입슬리피지가 포함되며 청산 후에는 청산수수료와 청산슬리피지가 반영되어 손익이 계산된다.

SlippageMethod

설명 : 슬리피지 설정 상태

시스템 트레이딩 설정의 비용 /수량 탭에서
슬리피지를 %로 설정했는지 Pt로 설정했는지를 리턴,
0이면 %, 1이면 Pt

함수 : SlippageMethod

CommissionMethod

설명 : 수수료 설정 상태

시스템 트레이딩 설정의 비용 /수량 탭에서
수수료를 %로 설정했는지 Pt로 설정했는지를 리턴,
0이면 %, 1이면 Pt

함수 : CommissionMethod

EntryCommission

설명 : 설정된 진입 수수료

시스템 트레이딩 설정의 비용/수량 탭에서 설정한 진입 수수료 값을 리턴

함수 : EntryCommission

EntrySlippage

설명 : 설정된 진입 슬리피지

시스템 트레이딩 설정의 비용/수량 탭에서 설정한 진입 슬리피지 값을 리턴

함수 : EntrySlippage

ExitSlippage

설명 : 설정된 청산 슬리피지

시스템 트레이딩 설정의 비용/수량 탭에서 설정한 청산 슬리피지 값을 리턴

함수 : ExitSlippage

ExitCommission

설명 : 설정된 청산 수수료

시스템 트레이딩 설정의 비용/수량 탭에서 설정한 청산 수수료 값을 리턴

함수 : ExitCommission

LatestEntryName

설명 : 가장최근 포지션의 진입명

※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.

※ 최근 거래는 0, 직전 거래 1,.....

※ (0)은 생략가능

함수 : LatestEntryName(N)

예제 : 주로 피라미딩 진입시 이용을 합니다.



LatestEntryName(0) → 현재 진행중 인 포지션의 가장 최근 진입의 이름을 리턴합니다.

LatestEntryName(1) → 직전 거래의 가장 마지막 진입의 이름이 리턴됩니다.

LatestEntryPrice

설명 : 가장최근 포지션의 진입가격

※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.

※ 최근 거래는 0, 직전 거래 1,.....

※ (0)은 생략가능

함수 : LatestEntryPrice(N)

예제 : 주로 피라미딩 진입시 이용을 합니다.



LatestEntryPrice(0) → 현재 진행중 인 포지션의 가장 최근 진입의 가격이 리턴합니다.

LatestEntryPrice(1) → 직전 거래의 가장 마지막 진입의 가격이 리턴됩니다.

LatestExitName

설명 : 포지션의 가장 최근 청산명

※ N은 거래 위치. 거래위치는 현재를 기준으로 역순으로 지정.

※ 최근 거래는 0, 직전 거래 1,.....

설명 : 지표/검색/강조에서만 사용가능(시스템의 보유수량을 리턴, 매수수량은 양수로, 매도수량은 음수로 리턴됨)

I_MarketPosition

설명 : 지표/검색/강조에서만 사용가능(시스템의 포지션상태를 리턴)

13) 전략성과함수

AvgBarsLosTrade

설명 : 청산 완료된 전체 거래 중 손실거래 평균 봉 개수

함수 : AvgBarsLosTrade

AvgBarsWinTrade

설명 : 청산 완료된 전체 거래 중 수익거래 평균 봉 개수

함수 : AvgBarsWinTrade

GrossLoss

설명 : 청산 완료된 전체 거래 중 손실거래의 손실 총합

함수 : GrossLoss

GrossProfit

설명 : 청산 완료된 전체 거래 중 수익거래의 수익 총합

함수 : GrossLoss

LargestLosTrade

설명 : 청산 완료된 전체 거래 중 손실거래에서 가장 큰 손실금액

함수 : LargestLosTrade

LargestWinTrade

설명 : 청산 완료된 전체 거래 중 수익거래에서 가장 큰 수익금액

함수 : LargestWinTrade

MaxConsecLosers

설명 : 청산 완료된 전체 거래 중 최대 연속 손실거래 횟수

함수 : MaxConsecLosers

MaxConsecWinners

설명 : 청산 완료된 전체 거래 중 최대 연속 수익거래 횟수

함수 : MaxConsecWinners

MaxContractsHeld

설명 : 청산 완료된 전체 거래 중 한 포지션이 보유한 최대 계약수

함수 : MaxContractsHeld

MaxIDDrawDown

설명 : 청산 완료된 전체 거래 중 최대 자본 인하분

함수 : MaxIDDrawDown

NetProfit

설명 : 청산 완료된 전체 거래의 최종 손익

함수 : NetProfit

예제 : 현재까지의 거래에서 손익을 가감하여 투입금액을 산정하여 수량계산

```
inputs: 최초투입금액(10000000);
var : capital(0);

5 # 최초투입금액에 현재까지의 손익을 합산하여 진입금액 계산
# 옵션과 같은 경우 손익이 포인트로 도출되므로 NetProfit에
# BigPointValue를 곱하여 금액으로 변경해 줌
capital = int(최초투입금액+(NetProfit*BigPointValue));

10 if CodeCategory == 1 Then{ #코스피 종목일 경우
    if BasePrice < 50000 Then #기준가격이 5만원 이하 10주 단위
        Var1 = int(int(capital/C)/10)*10;
    Else
        Var1 = int(capital/C);#5만원 이상 1주단위
15 }

    if CodeCategory == 2 Then #코스닥 종목일 경우 1주 단위
        Var1 = int(capital/C);

20 if CodeCategory == 6 Then #옵션 종목일 경우
    Var1 = int(capital/(C*BigPointValue)); #1계약 단위
```

NumLosTrades

설명 : 청산 완료된 전체 거래 중 손실 거래의 횟수

함수 : NumLosTrades

NumWinTrades

설명 : 청산 완료된 전체 거래 중 수익 거래의 횟수

함수 : NumWinTrades

PercentProfit

설명 : 청산 완료된 전체 거래의 승률

함수 : PercentProfit

TotalBarsLosTrades

설명 : 청산 완료된 전체 거래 중 손실거래의 전체 봉 개수

함수 : TotalBarsLosTrades

TotalBarsWinTrades

설명 : 청산 완료된 전체 거래 중 수익거래의 전체 봉 개수

함수 : TotalBarsWinTrades

TotalTrades

설명 : 청산 완료된 전체 거래의 횟수

함수 : TotalTrades

14) 사용자함수(기본 내장)

Accum

설명 : value를 누적.

※ 차트 상 전체의 value를 누적하게 된다.

※ Value는 수치 값이어야 한다.

함수 : Accum(Value)

예제 : accum(1) → 봉마다 1씩 누적하므로 index를 대체해서 사용되기도 한다.

AccumN

설명 : Length로 지정한 기간동안 value를 누적.

※ Value는 수치 값이어야 한다.

함수 : AccumN(Value, Length)

예제 : AccumN(C,20) → 20 개봉 증가

※ value에는 수치값만 와야 하므로 조건식의 만족횟수는 iff 함수와 함께 사용하여 조건만족 횟수를 계산하게 만들어야 한다.

AccumN(iff(C> O,1,0),20) → 20개봉 동안 양봉의 개수

Correlation

설명 : Period기간 동안의 num1과 num2의 상관관계

상관계수는 두 변수 간의 상호 종속관계를 특정해 주는 계수로서 $-1 \sim +1$ 사이의 값을 가진다.

+1에 가까울수록 강한 비례관계를 가지고 -1 에 가까울수록 강한 반비례 관계를 가진다

함수 : Correlation(num1,num2,Period)

예제 : plot1(Correlation(ma(v,5),ma(C,5),10));

→ 10개봉 동안 거래량 5이평과 종가 5이평과의 상관관계

CloseD(N)

설명 : N일전 일간 종가

※ 차트의 데이터를 분석하여 일간 종가를 계산

※ 차트 날짜 이상의 이전일 값은 계산할 수 없다.

※ 참조종목의 일간 종가는 꼭 closeD를 사용해야 한다. Data2(CloseD(1))

함수 : CloseD(N)

예제 : CloseD(2) → 2일전 종가 → 차트상 2일전 데이터가 없을 경우 N/A

CloseD(0) → 당일 종가 → (0)을 생략하지 못한다.

HighD

설명 : N일전 일간 최고가

※ 차트의 데이터를 분석하여 일간 최고가를 계산

※ 차트 날짜 이상의 이전일 값은 계산할 수 없다.

※ 참조종목의 일간 최고가는 꼭 HighD를 사용해야 한다. Data2(HighD (1))

함수 : HighD(N)

예제 : HighD(2) → 2일전 최고가 → 차트상 2일전 데이터가 없을 경우 N/A

HighD(0) → 당일 최고가 → (0)을 생략하지 못한다.

LowD

설명 : N일전 일간 최저가

※ 차트의 데이터를 분석하여 일간 최저가를 계산

※ 차트 날짜 이상의 이전일 값은 계산할 수 없다.

※ 참조종목의 일간 최저가는 꼭 LowD를 사용해야 한다. Data2(LowD(1))

함수 : LowD (N)

예제 : LowD (2) → 2일전 최저가 → 차트상 2일전 데이터가 없을 경우 N/A

LowD(0) → 당일 최저가 → (0)을 생략하지 못한다.

OpenD

설명 : N일전 일간 시초가

※ 차트의 데이터를 분석하여 일간 시초가를 계산

※ 차트 날짜 이상의 이전일 값은 계산할 수 없다.

※ 참조종목의 일간 시초가는 꼭 OpenD를 사용해야 한다. Data2(OpenD(1))

함수 : OpenD(N)

예제 : CloseD(2) → 2일전 시초가 → 차트상 2일전 데이터가 없을 경우 N/A

OpenD(0) → 당일 시초가 → (0)을 생략하지 못한다.

DayIndex

설명 : 분봉에서 당일의 봉번호

※ 차트의 주기가 일간이상이면 0값만 리턴한다.

※ 첫봉을 0으로 해서 1씩 증가한다.

함수 : DayIndex

예제 : if dayindex == 0 then → 당일 첫봉

highest(H,dayindex+1) → 당일 현재시점까지의 최고가

(첫 봉이 0이므로 dayindex에 1을 더해줘서 계산한다);

ma(c,dayindex+1) → 당일 이평

C[dayindex] → 첫봉의 증가

CountIF

설명 : Period기간 동안 condition으로 지정한 조건만족 횟수를 구함

함수 : CountIF(condition,Period)

※ condition : 조건

※ Period : 기간

예제 : CountIf(C>O,10); → 10개 안에 양봉의 개수

if crossup(c,ma(c,20)) and countif(crossdown(c,ma(c,20)),5) < 1 then

Buy();

→ 최근 5개봉안에 종가가 20이동평균을 하향이탈한 경우가 없고

현재 종가가 20이평을 상향돌파하면 매수

MedianPrice

설명 : 고가와 저가의 중간가격

함수 : MedianPrice

예제 : $C > O$ and $\max(C,O) < \text{MedianPrice}$ → 중간값 아래에 몸통이 위치한 양봉
 $\text{Ma}(\text{MedianPrice},20)$ → 중간가 기준 20기간 이동평균

MRO

설명 : Period기간 동안 condition으로 지정한 조건을 만족한 봉들 중에서 현재로부터 N번째 가깝게 만족한 봉의 위치가 몇 개봉 이전인지 봉갯수를 리턴해 준다.
만족한 경우가 없으면 -1값 리턴

함수 : $\text{MRO}(\text{Condition}, \text{Period}, N)$

※ Condition : 조건

※ Period : 기간

※ N : 현재봉에서 가까운 순서로 1, 2, 3,....

예제 : $\text{MRO}(\text{crossup}(\text{ma}(c,5), \text{ma}(c,20)), 30, 1)$

→ 최근 30개 봉 안의 5-20 골든크로스 중 현재로부터 가장 가까운 골든크로스의 위치(값이 5가 나온다면 5개봉 전에 발생했음을 의미)

```
# 5-20 골든크로스
Condition1 = crossup(ma(c,5),ma(c,20));
# 가장최근 골든크로스 위치(현재면 0값 리턴)
5 var1 = MRO(Condition1,30,1);
# 직전 골든크로스 위치
var2 = MRO(Condition1,30,2);

if Condition1 == true and #1
10   var1 != -1 and var2 != -1 and #2
    var1 < 5 and var2 > 10 and #3
    H[var1] < H[var2] Then #4
    buy();

15 #1 현재 봉에서 골든크로스 발생
    #2 지정한 기간동안 2번의 골든크로스가 있었음을 의미
    #3 var1은 5봉이내 var2는 10봉이상 이전
    #4 최근 골든크로스의 고가가 직전 골든크로스의 고가보다 작다
```

Range

설명 : 봉길이, 고가와 저가의 차이

함수 : Range

예제 : $\text{ma}(\text{Range},20)$ → 봉길이의 20기간 이동평균

$\text{Range} > \text{ma}(\text{Range},20)*2$

→ 현재 봉의 봉길이가 직전 20봉 동안의 평균길이의 2배

STD

설명 : value의 Length 기간 표준편차

함수 : STD(value,Length)

예제 : STD(c,20)

SwingHigh

설명 : 최근 Period 기간 동안 발생된 Value의 SwingHigh 중 현재로부터 Nth번째 가까운 위치에 발생한 Swinghigh의 값을 리턴한다

함수 : SwingHigh(Nth,value,leftbar,Rightbar, Period)

※ Nth : 순위, N번째

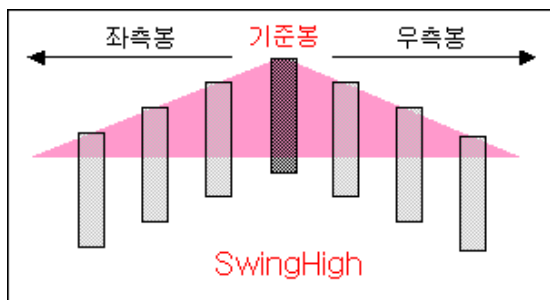
※ value : SwingHigh 계산에 사용하고자 하는 가격

※ Leftbar : 기준봉의 좌측 봉 갯수

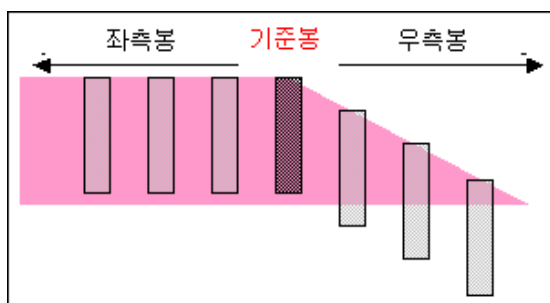
※ Rightbar : 기준봉의 우측 봉 갯수

※ Period : SwingHigh를 찾는 대상 봉 갯수

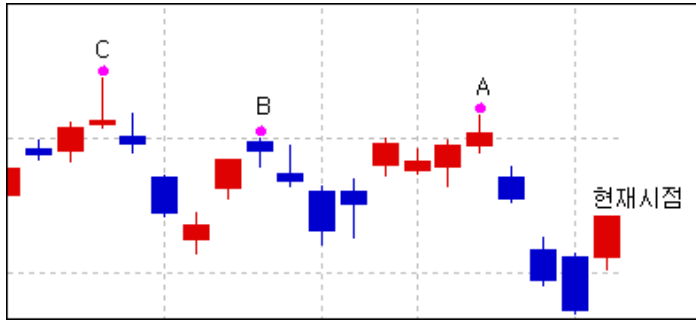
참고 : SwingHigh는 기준봉의 값이 기준봉의 좌측봉들의 값보다 크거나 같고 기준봉의 우측봉들의 값보다 큰 것을 말하는데 이를 그림으로 표시하면 다음과 같다.



※ 좌측봉들의 값은 기준봉의 값보다 작거나 같으면 되므로 아래와 같이 좌측봉들의 값은 기준봉의 값과 모두 같고 우측봉들의 값은 작은 경우에도 SwingHigh는 성립된다. (좌측봉들의 값 \leq 기준봉 > 우측봉들의 값)



※ 아래 그림과 같이 스윙하이가 발생하였다면



현재시점을 기준으로

SwingHigh(1,H,3,3,20)는 A의 고가를

SwingHigh(2,H,3,3,20)는 B의 고가를

SwingHigh(3,H,3,3,20)는 C의 고가를 리턴하고

20개봉 동안 3번만 발생했으므로 Nth가 4이상이면 모두 -1값을 리턴한다.

SwingHighBar

설명 : 최근 Period 기간 동안 발생된 Value의 SwingHigh 중 현재로부터 Nth번째 가까운 위치에 발생한 Swinghigh가 현재 봉을 기준으로 몇 개봉 이전인지 봉갯수를 리턴해 준다.

함수 : SwingHighBar(Nth,value,leftbar,Rightbar, Period)

※ Nth : 순위, N번째

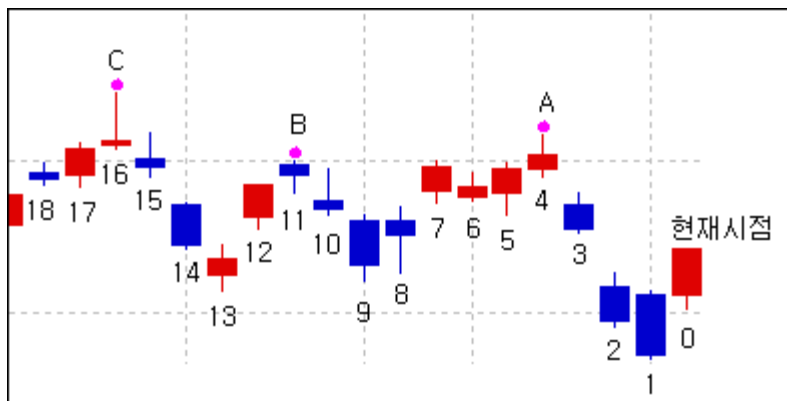
※ value : SwingHighBar 계산에 사용하고자 하는 가격

※ Leftbar : 기준봉의 좌측 봉 갯수

※ Rightbar : 기준봉의 우측 봉 갯수

※ Period : SwingHighBar를 찾는 대상 봉 갯수

참고 : 아래 그림과 같이 스윙하이가 발생하였다면



현재시점을 기준으로

SwingHighBar(1,H,3,3,20)는 4를

SwingHighBar(2,H,3,3,20)는 11을

SwingHighBar(3,H,3,3,20)는 16을 리턴하고

20개봉 동안 3번만 발생했으므로 Nth가 4이상이면 모두 -1값을 리턴한다.

SwingLow

설명 : 최근 Period 기간 동안 발생된 Value의 SwingLow 중 현재로부터 Nth번째 가까운 위치에 발생한 SwingLow의 값을 리턴한다

함수 : SwingLow(Nth,value,leftbar,Rightbar, Period)

※ Nth : 순위, N번째

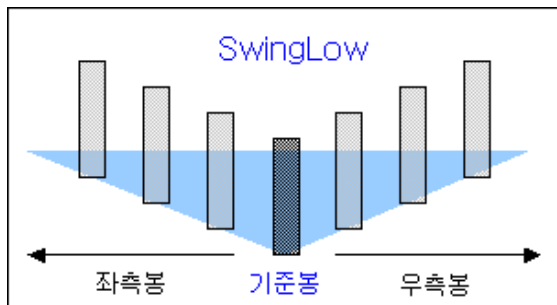
※ value : SwingLow 계산에 사용하고자 하는 가격

※ Leftbar : 기준봉의 좌측 봉 갯수

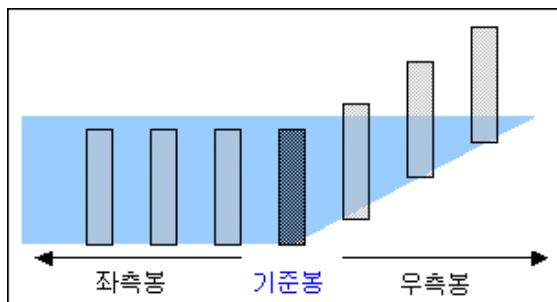
※ Rightbar : 기준봉의 우측 봉 갯수

※ Period : SwingLow를 찾는 대상 봉 갯수

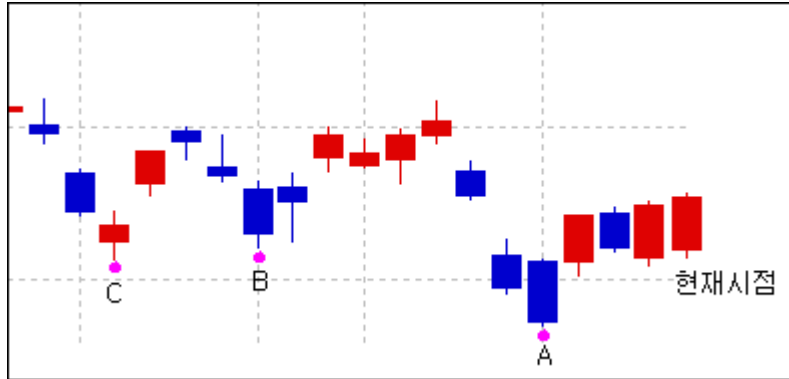
참고 : SwingLow는 기준봉의 값이 기준봉의 좌측봉들의 값보다 작거나 같고 기준봉의 우측봉들의 값보다는 작은 것을 말하는데 이를 그림으로 표시하면 다음과 같다.



※ 좌측봉들의 값은 기준봉의 값보다 크거나 같으면 되므로 아래와 같이 좌측봉들의 값은 기준봉의 값과 모두 같고 우측봉들의 값은 큰 경우에도 SwingLow는 성립된다. (좌측봉들의 값 \geq 기준봉 < 우측봉들의 값)



※ 아래 그림과 같이 SwingLow가 발생하였다면



현재시점을 기준으로

SwingLow(1,L,3,3,20)는 A의 저가를

SwingLow(2,L,3,3,20)는 B의 저가를

SwingLow(3,L,3,3,20)는 C의 저가를 리턴하고

20개봉 동안 3번만 발생했으므로 Nth가 4이상이면 모두 -1값을 리턴한다.

SwingLowBar

설명 : 최근 Period 기간 동안 발생한 Value의 SwingLow 중 현재로부터 Nth번째 가까운 위치에 발생한 SwingLow가 현재 봉을 기준으로 몇 개봉 이전인지 봉갯수를 리턴해 준다.

함수 : SwingLowBar(Nth,value,leftbar,Rightbar, Period)

※ Nth : 순위, N번째

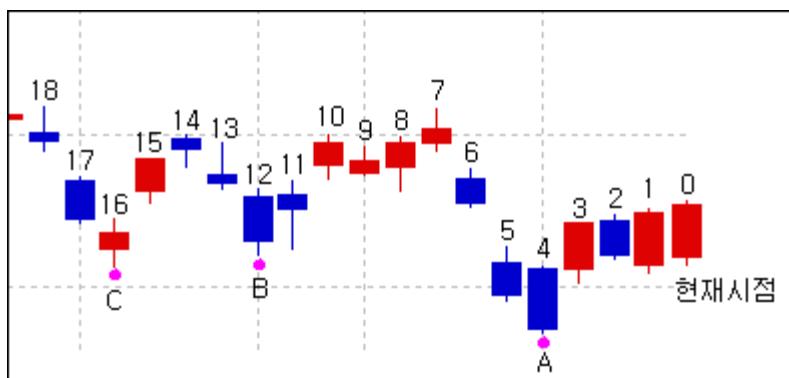
※ value : SwingLowBar 계산에 사용하고자 하는 가격

※ Leftbar : 기준봉의 좌측 봉 갯수

※ Rightbar : 기준봉의 우측 봉 갯수

※ Period : SwingLowBar를 찾는 대상 봉 개수

참고 : 아래 그림과 같이 SwingLow가 발생하였다면



현재시점을 기준으로

SwingLowBar(1,L,3,3,20)는 4를

SwingLowBar(2,L,3,3,20)는 12를

SwingLowBar(3,L,3,3,20)는 16을 리턴하고

20개봉 동안 3번만 발생했으므로 Nth가 4이상이면 모두 -1값을 리턴한다.

TrueHigh

설명 : 진고가

현재봉의 고가가 전봉의 종가보다 크거나 같으면 현재봉의 고가를

현재봉의 고가가 전봉의 종가보다 작으면 전봉의 종가를 리턴

함수 : TrueHigh

예제 : $\text{TrueHigh} > \text{highest}(\text{TrueHigh}, 20)[1]$ → 최근 20개봉 진고가 중 최고가를 갱신

TrueLow

설명 : 진저가

현재봉의 저가가 전봉의 종가보다 크면 전봉의 종가를

현재봉의 저가가 전봉의 종가보다 작거나 같으면 현재봉의 저가를 리턴

함수 : TrueLow

예제 : $\text{TrueLow} < \text{lowest}(\text{TrueLow}, 20)[1]$ → 최근 20개봉 진저가 중 최저가를 갱신

TrueRange

설명 : 진봉길이

$\text{TrueRange} = \text{TrueHigh} - \text{TrueLow}$

함수 : TrueRange

예제 : $\text{TrueRange} \geq \text{ma}(\text{TrueRange}, 20)[1] * 2$

→ 직전 20개봉 진봉길이 보다 현재 진봉길이가 2배이상 크다

WC

설명 : 가중종가. 종가에 가중치를 둔 평균가격

$(\text{종가} * 2 + \text{고가} + \text{저가}) / 4$

함수 : WC

예제 : $\text{ma}(\text{WC}, 20)$ → 가중종가 20기간 이동평균

AccDist

설명 : Accumulation_Distribution 지표함수

함수 : AccDist

계산 : $((\text{종가} - \text{저가}) - (\text{고가} - \text{종가})) / (\text{고가} - \text{저가}) * \text{거래량의 차트 전체기간 합}$

ADX

설명 : ADX(Average Directional Movement Index) 지표함수

함수 : ADX(Period)

※ Period : 기간

계산 : Directional Movement Index의 이동평균값으로 구한다.

$PDM = (\text{당일고가} - \text{전일고가} > 0) \text{ 이고}$

$(\text{당일고가} - \text{전일고가} > \text{전일저가} - \text{당일저가}) \text{ 이면,}$

$(\text{당일고가} - \text{전일고가}) \text{ 이고, 아니면 } 0.$

$MDM = (\text{전일저가} - \text{당일저가} > 0) \text{ 이고}$

$(\text{당일고가} - \text{전일고가} < \text{전일저가} - \text{당일저가}) \text{ 이면,}$

$(\text{전일저가} - \text{당일저가}) \text{ 이고, 아니면 } 0.$

$PDMMA = (+DM) \text{의 Period 일 이동평균}$

$MDMMA = (-DM) \text{의 Period 일 이동평균}$

$DX = PDMMA - MDMMA \text{의 절대값} / (PDMMA + MDMMA) \times 100$

$ADX = DX \text{의 Period 일 이동평균}$

ATR

설명 : ATR 지표함수, Average True Range

함수 : ATR(Period)

※ Period : 기간

계산 : $ATR = \text{TrueRange}$ 의 Period 기간 이동평균 값

Average

설명 : value값의 Period기간 이동평균

ma와 같음

함수 : Average(value,Period)

※ value : 기준값

※ Period : 기간

AverageFC

설명 : value값의 Period기간 이동평균,

ma나 average와 같이 단순이평을 구하는 함수로 계산방식이 다름.

이전봉까지의 합산값을 기억했다가 현재값을 추가하고

가장 과거의 값을 빼는 방식으로 계산.

함수 : AverageFC(value,Period)

※ value : 기준값

※ Period : 기간

BollBandDown

설명 : 볼린저밴드 하단

함수 : BollBandDown(Period,D)

계산 : Period기간 종가 이동평균-Period기간 표준편차*D

※ Period : 이동평균의 기간

※ D : 승수값

BollBandUp

설명 : 볼린저밴드 상단

함수 : BollBandUp(Period,D)

※ Period : 이동평균의 기간

※ D : 승수값

계산 : Period기간 종가 이동평균+Period기간 표준편차*D

BW

설명 : 바이너리웨이브(BinaryWave) 지표함수

함수 : BW(maPeriod, RocPeriod, stoPeriod1, stoPeriod2)

※ maPeriod : 이동평균 기간

※ RocPeriod : Price ROC 기간

※ StoPeriod1 : StochasticsK 기간1

※ StoPeriod2 : StochasticsK 기간2

계산 : 바이너리 웨이브는 MACD,지수이동평균, ROC, 스토캐스틱 총 4개의 지표를 이용하여 만든 지표입니다. 각각 지표의 상태에 따라 값을 -1, 0, 1 값만 가지도록 하여 바이너리 웨이브값이 -4에서 4까지만 표현되도록 한것입니다.

A = macd가 macdSignal 보다 크면 1, 아니면 -1

B = 종가가 n일 이동평균보다 크면 1, 아니면 -1

C = ROC가 0이상이면 1, 아니면 -1

D = 스토캐스틱이 50이상이면 1, 아니면 -1

BW = A+B+C+D

CCI

설명 : CCI(Commodity Channel Index) 지표함수

함수 : CCI(Period)

※ Period : 기간

계산 : value1 = (고가+저가+종가)/3

value2 = value1의 n일 이동평균값
value3 = (value1-value2)의 절대값
value4 = value3의 n일 이동평균값
 $CCI = (value1 - value2) / (value4 * 0.015)$

CMO

설명 : CMO(Chandre Momentum Oscillator) 지표함수

함수 : CMO(Period)

※ Period : 기간

계산 : 모멘텀 = 증가 - n일전 증가

$CMO = (N\text{기간 상승모멘텀} - N\text{기간 하락모멘텀})\text{의 합} /$
 $(N\text{기간 상승모멘텀} + N\text{기간 하락모멘텀})\text{의 합} * 100$

CO

설명 : Chaikin's Oscillator 지표함수

함수 : CO

계산 : value = ((증가-저가)-(고가-종가))/(고가-저가)*거래량;

value1 = value의 누적

value2 = value1의 3일 지수이동평균값

value3 = value1의 10일 지수이동평균값

$CO = value2 - value3$

CSar

설명 : 종가 파라볼릭

종가 파라볼릭은 극단 값에 사용되는 고가와 저가 대신 종가를 이용

함수 : CSar(AF,AFMAX)

※ AF : 가속도

※ AFMAX : 가속최대치

계산 : 익일 SAR = 당일 SAR + AF(EP - 당일SAR)

AF = 가속변수

EP = 종가

AF는 0.02에서 새로 출발해 내일의 SAR값을 계산한다. 이때 AF값을 임의로 조정할 수도 있는데, AF값을 크게 하면 추세의 변화에 민감

Sar

설명 : 파라볼릭

함수 : Sar(AF,AFMAX)

※ AF : 가속도

※ AFMAX : 가속최대치

계산 : 익일 SAR = 당일 SAR + AF(EP - 당일SAR)

AF = 가속변수

EP = 극단가격

AF는 0.02에서 새로 출발해 내일의 SAR값을 계산한다. 이때 AF값을 임의로 조정할 수도 있는데, AF값을 크게 하면 추세의 변화에 민감

CV

설명 : Chaikin's Volatility 지표함수

함수 : CV(Period)

※ Period : 기간

계산 : A = (고가-저가)의 Period 일 지수이동평균값

B = A의 Period 일전 값

CV = (A-B)/B*100

Detrend

설명 : Detrend 지표함수

함수 : Detrend(value,Period)

※ value : 데이터 및 계산식

※ Period : 기간

계산 : AvgVal = Period 기간 이동평균값

DFact = (N*0.5) + 1

Detrend = C - AvgVal의 Dfact 이전 봉의 값

Disparity

설명 : 이격도

함수 : DisParity(Period)

※ Period : 이동평균 기간

계산 : 주가/ Period 기간 이동평균값 × 100

DiMinus

설명 : Directional Moving Average Index Minus 지표함수

함수 : DiMinus(Period)

※ Period : 기간

계산: $DMPlus = (\text{당일고가} - \text{전일고가}) > 0$ 이고
(당일고가-전일고가 > 전일저가-당일저가) 이면
(당일고가-전일고가)이고 아니면 0.
 $DMMinus = (\text{전일저가} - \text{당일저가}) > 0$ 이고
(당일고가-전일고가 < 전일저가-당일저가) 이면
(전일저가-당일저가)이고 아니면 0.
 $TR = (\text{고가} - \text{저가}), (\text{전일종가} - \text{당일고가})$ 의 절대값,
(전일종가-당일저가)의 절대값 중 최대값
 $DMnPlus = (+DM)$ 의 Period 기간 이동평균값
 $DMnMinus = (-DM)$ 의 Period 기간 이동평균값
 $TRn = TR$ 의 Period 기간 이동평균값
 $DIPlus = DMnPlus/TRn$
 $DIMinus = DMnMinus/TRn$

Diplus

설명 : Directional Moving Average Index Plus 지표함수

함수 : DiPlus (Period)

※ Period : 기간

예제 : $DMPlus = (\text{당일고가} - \text{전일고가}) > 0$ 이고
(당일고가-전일고가 > 전일저가-당일저가) 이면
(당일고가-전일고가)이고 아니면 0.
 $DMMinus = (\text{전일저가} - \text{당일저가}) > 0$ 이고
(당일고가-전일고가 < 전일저가-당일저가) 이면
(전일저가-당일저가)이고 아니면 0.
 $TR = (\text{고가} - \text{저가}), (\text{전일종가} - \text{당일고가})$ 의 절대값,
(전일종가-당일저가)의 절대값 중 최대값
 $DMnPlus = (+DM)$ 의 Period 기간 이동평균값
 $DMnMinus = (-DM)$ 의 Period 기간 이동평균값
 $TRn = TR$ 의 Period 기간 이동평균값
 $DIPlus = DMnPlus/TRn$
 $DIMinus = DMnMinus/TRn$

DMI

설명 : Directional Movement Index 지표함수

함수 : DMI(Period)

※ Period : 기간

계산 : $DMPlus = (당일고가 - 전일고가) > 0$ 이고
(당일고가-전일고가 > 전일저가-당일저가) 이면
(당일고가-전일고가)이고 아니면 0.
 $DMMinus = (전일저가 - 당일저가) > 0$ 이고
(당일고가-전일고가 < 전일저가-당일저가) 이면
(전일저가-당일저가)이고 아니면 0.
 $TR = (고가 - 저가), (전일종가 - 당일고가)의 절대값,$
(전일종가-당일저가)의 절대값 중 최대값
 $DMnPlus = (+DM)의 Period 기간 이동평균값$
 $DMnMinus = (-DM)의 Period 기간 이동평균값$
 $TRn = TR의 Period 기간 이동평균값$
 $DIPlus = DMnPlus/TRn$
 $DIMinus = DMnMinus/TRn$
 $DMI = (DIPlus - DIMinus)의 절대값 / (DIPlus + DIMinus) * 100$

EnvelopeDown

설명 : Envelope Trading Bands 하한

함수 : EnvelopeDown(period,percent)

※ Period : 기간

※ Percent : % 수치

계산 : $Period일\ 주가\ 이동\ 평균 - (Period일\ 주가\ 이동\ 평균 \times Percent\ (\%))$

EnvelopeUp

설명 : Envelope Trading Bands 상한

함수 : EnvelopeUp(period,percent)

※ Period : 기간

※ Percent : % 수치

계산 : $Period일\ 주가\ 이동\ 평균 + (Period일\ 주가\ 이동\ 평균 \times Percent\ (\%))$

EOM

설명 : Ease Of Momentum 지표함수

함수 : EOM

계산 : $MP = (금일고가 + 금일저가) / 2 - (전일고가 + 전일저가) / 2$

$BR = (거래량 / 10000) / (금일고가 - 금일저가)$

$EOM = MP / BR$

KeltnerChannel

설명 : Keltner Channel 지표함수

함수 : KeltnerChannel(value, Period, ATRs)

※ value : 데이터 및 계산

※ Period : 기간

※ ATRs : ATR 배수

계산 : TH = 전일종가가 현재고가보다 크면 전일종가 아니면 현재고가

TL = 전일종가가 현재저가보다 작으면 전일종가 아니면 현재저가

TR = TH - TL

ATR = TR의 Period 일 이동평균값

Keltner = 증가 Period 기간 이동평균값 + ATR값 * ATRs

** Keltner 상단 = Keltner + (ATR*배수);

** Keltner 하단 = Keltner - (ATR*배수);

LRL

설명 : LRL(Linear Regression Line) 지표함수

함수 : LRL(value,Period)

※ value : 데이터 및 계산

※ Period : 기간

계산 : a1= 봉위치

a2 = 봉위치제곱

a3 = 봉위치 Period 기간 평균

a4 = 봉위치제곱 Period 기간 평균

a5 = 봉위치 Period 기간 평균 제곱

a6 = 증가*봉위치

a7 = value6의 Period 기간 이동평균

a8 = 증가 Period 기간 이동평균

LRL = (a7 - a3 * a8) / (a4 - a5) * (a1 - a3) + a8

LRS

설명 : Linear Regression Slope 지표함수

함수 : LRS (Value, Period)

※ value : 데이터 및 계산식

※ Period : 기간

계산 : SumBars = 기간값 * (기간값 - 1) * .5

$$\text{SumSqrBars} = (\text{기간값} - 1) * \text{기간값} * (2 * \text{기간값} - 1) / 6$$

$$\text{Sum1} = \text{Period 기간 value의 합산, 현재를 기준으로 과거치에 1씩 가중}$$

$$\text{SumY} = \text{Period 기간 value의 합산}$$

$$\text{Sum2} = \text{SumBars} * \text{SumY}$$

$$\text{Num1} = \text{기간값} * \text{Sum1} - \text{Sum2}$$

$$\text{Num2} = \text{SumBars} * \text{SumBars} - \text{기간} * \text{SumSqrBars};$$

$$\text{LRS} = \text{Num1} / \text{Num2};$$

MACD

설명 : MACD(Moving Average Convergence & Divergence) 지표함수

함수 : MACD(Short, Long)

※ Short : 단기 지수이동평균 기간

※ Long : 장기 지수이동평균 기간

계산 : 단기 지수 이동 평균 - 장기 지수 이동 평균

MACD_OSC

설명 : MACD Oscillator 지표함수

함수 : MACD_OSC(ShorPeriod, LongPeriod, Period);

※ ShortPeriod : 단기 지수이동평균 기간

※ LongPeriod : 장기 지수이동평균 기간

※ Period : 시그널을 만들기 위한 지수이동평균 기간

계산 : MACD = 단기 지수이동평균 - 장기 지수이동평균

MACD 시그널 = MACD를 Period 기간 지수이동평균

MACD 오실레이터 = MACD-MACD 시그널

MassIndex

설명 : Mass Index 지표함수

함수 : MassIndex(Period1,Period2)

※ Period1 : 지수이동평균 기간

※ Period2 : 합산기간

계산 : value1 = (고가-저가)의 Period1 기간 지수이동평균

value2 = value1의 Period1 기간 지수이동평균

MI = value1/value2의 Period2 기간 동안의 합

MFI

설명 : Money Flow Index 지표함수

함수 : MFI(Period);

※ Period : 기간

계산 : $TP = (\text{고가} + \text{저가} + \text{종가}) / 3$

$MF = TP \times \text{거래량}$

PMF = 당일 TP > 전일 TP일 때 MF 아니면 0

NMF = 당일 TP < 전일 TP일 때 MF 아니면 0

Value1 = PMF의 Period 기간 누적

value2 = NMF의 Period 기간 누적

$MFI = 100 - 100 / (1 + (\text{value1} / \text{value2}))$;

Momentum

설명 : 모멘텀, N일전 종가대비 현재가 비율

함수 : Momentum(N)

※ N : 기간

계산 : $\text{당일 종가} / \text{n일전 종가} * 100$

NVI

설명 : Negative Volume Index 지표함수

함수 : NVI

계산 : 최초값 100

현재 거래량이 직전봉 거래량보다 작으면

$NVI = \text{직전봉 NVI} + ((\text{현재봉 종가} - \text{직전봉 종가}) / \text{직전봉 종가} * \text{전일 NVI})$

현재 거래량이 직전봉 거래량보다 크거나 같다면

$NVI = \text{직전봉 NVI}$

OBV

설명 : On Balance Volume 지표함수

함수 : OBV

계산 : 당일 종가 > 전일 종가 이면 $OBV = \text{전일 OBV} + \text{당일 거래량}$

당일 종가 < 전일 종가 이면 $OBV = \text{전일 OBV} - \text{당일 거래량}$

당일 종가 = 전일 종가 이면 $OBV = \text{전일 OBV}$

OSCP

설명 : Price Oscillator 지표함수

함수 : OSCP(Period1, Period2)

※ Period1 : 단기 이동평균 기간

※ Period2 : 장기 이동평균 기간

계산 : $(\text{단기이동평균} - \text{장기이동평균}) / \text{단기이동평균} * 100$

OSCV

설명 : Volume Oscillator 지표함수

함수 : OSCV(Period1, Period2)

※ Period1 : 단기 거래량 이동평균 기간

※ Period2 : 장기 거래량 이동평균 기간

계산 : $((\text{단기거래량 이동평균} - \text{장기거래량 이동평균}) / \text{단기거래량 이동평균}) * 100$

PercentR

설명 : %R 지표함수

함수 : PercnetR(Period)

※ Period : 기간

계산 : A1 = Period 기간 최고가

B1 = Period 기간 최저가

C1 = A1 - B1

%R = $100 - ((A1 - \text{종가}) / C1) * 100$

PROC

설명 : Price Rate Of Change 지표함수

함수 : PROC(Period)

※ Period : 기간

계산 : $(\text{현재종가} - \text{Period 기간 전 종가}) / \text{Period 기간 전 종가} * 100$

PVI

설명 : Positive Volume Index 지표함수

함수 : PVI

계산 : 최초값 100

현재 거래량이 직전봉 거래량보다 크면

$PVI = \text{전일 PVI} + ((\text{현재종가} - \text{직전봉 종가}) / \text{직전봉 종가} * \text{직전봉 PVI})$

현재 거래량이 직전봉 거래량보다 작거나 같다면

$PVI = \text{직전봉 PVI}$

PVT

설명 : Price Volume Trend 지표함수

함수 : PVT

계산 : 직전봉 PVT + ((현재 증가- 직전봉 증가)/ 직전봉 증가)*현재 거래량
*** 최초값 0

RSI

설명 : Relative Strength Index 지표함수

함수 : RSI(Period)

※ Period : 기간

계산 : A1 = Period 기간 상승폭의 합계

B1 = Period 기간 하락폭의 합계

$RSI = A1 / (A1 + B1) * 100$

Simrido

설명 : 심리도 지표함수

함수 : Simrido(Period)

※ Period : 기간

계산 : (Period 기간 동안의 증가상승 횟수) / Period 기간 * 100

FastK

설명 : 스토케스틱 FastK 지표함수, 기본적 K

함수 : FastK(Period)

※ Period : 최고/최저가 계산 기간

계산 : H1 = Period 기간 최고가

L1 = Period 기간 최저가

$FastK = (현재증가 - L1) / (H1 - L1) * 100;$

FastD

설명 : 스토케스틱 FastD 지표함수, FastK를 지수이동평균 한 값

*** FastD 와 SlowK 와 StochasticsK는 같은 내용의 함수

함수 : FastD(Period1, Period2)

※ Period1 : FastK 계산 기간값

※ Period2 : Fast K를 지수이동평균 할 기간

계산 : H1 = Period1 기간 최고가

L1 = Period1 기간 최저가

$FastK = (현재증가 - L1) / (H1 - L1) * 100$

FastD = FastK를 Period2기간 지수이동평균 한 값

SlowK

설명 : 스토캐스틱 SlowK 지표함수, FastK를 지수이동평균 한 값
*** FastD 와 SlowK 와 StochasticsK는 같은 내용의 함수

함수 : SlowK(Period1,Period2)

※ Period1 : FastK 계산 기간값

※ Period2 : Fast K를 지수이동평균 할 기간

계산 : H1 = Period1 기간 최고가

L1 = Period1 기간 최저가

FastK = (현재종가- L1) / (H1 - L1)*100

SlowK = FastK를 Period2기간 지수이동평균 한 값

StochasticsK

설명 : 스토캐스틱 SlowK 지표함수, FastK를 지수이동평균 한 값
*** FastD 와 SlowK 와 StochasticsK는 같은 내용의 함수

함수 : StochasticsK(Period1,Period2)

※ Period1 : FastK 계산 기간값

※ Period2 : FastK를 지수이동평균 할 기간

계산 : H1 = Period1 기간 최고가

L1 = Period1 기간 최저가

FastK = (현재종가- L1) / (H1 - L1)*100

StochasticsK = FastK를 Period2기간 지수이동평균 한 값

StochasticsD

설명 : Stochastics SlowD 지표함수, SlowK를 지수이동평균 한 값

함수 : StochasticsD(Period1,Period2,Period)

※ Period1 : 최고/최저가 계산 기간

※ Period2 : Fast K를 지수이동평균할 기간

※ Period3 : SLowK를 지수이동평균할 기간

계산 : H1 = Period1 기간 최고가

L1 = Period1 기간 최저가

FastK = (현재종가- L1) / (H1 - L1)*100

StochasticsK = FastK를 Period2 기간 지수이동평균 한 값

StochasticsD = StochasticsK Period3 기간 지수이동평균 한 값

SMI

설명 : Stochastics Momentum Index 지표함수

함수 : SMI(Period1,Period2,Period3)

※ Period1 : 기간1

※ Period2 : 기간2

※ Period3 : 기간3

계산 : Lval = Period1 기간 최저가

Hval = Period1 기간 최고가

SM = 종가-(Lval+Hval)/2;

value1 = SM의 Period2 기간 지수이동평균

value2 = (Lval-Hval)의 Period2 기간 지수이동평균

value3 = value1의 Period3 기간 지수이동평균/2

value4 = value2의 Period3 기간 지수이동평균/2;

SMI = (value3/value4)*100;

SONAR

설명 : Sonar Momentum 지표함수

함수 : SONAR(Period)

※ Period : 기간

계산 : 현재 Period 기간 지수이동평균 - Period 일전 Period 기간 지수이동평균

TRIX

설명 : TRIX(Triple Exponential smoothed moving averages) 지표함수

함수 : TRIX(Period);

※ Period : 기간

계산 : A1 = 종가의 Period 기간 지수이동평균 값

B1 = A1의 값에 대해 Period 기간 지수이동평균 값

C1 = B1의 값에 대해 Period 기간 지수이동평균 값

TRIX = (현재 C1-직전봉 C1)/(직전봉 C1)*100

TSI

설명 : True Strength Index 지표함수

함수 : TSI(value, Period1,Period2,Period3)

※ value : 데이터 및 계산식

※ Period1 : 기간1

※ Period2 : 기간2

※ Period3 : 기간3

계산 : $VV = \text{value} - \text{직전봉 value}$

$A1 = VV \text{의 Period1 기간 지수이동평균}$

$A2 = A1 \text{의 Period2 기간 지수이동평균}$

$A3 = A2 \text{의 Period3 기간 지수이동평균}$

$B1 = VV \text{ 절대값의 Period1 기간 지수이동평균}$

$B2 = B1 \text{ 절대값의 Period1 기간 지수이동평균}$

$B3 = B2 \text{ 절대값의 Period1 기간 지수이동평균}$

$TSI = A3/B3*100$

TXAverage

설명 : 3중 지수이동평균 함수 (Triple Exponential Moving Average)

함수 : $TXAverage(\text{value}, \text{Period1}, \text{Period2}, \text{Period3})$

※ value : 데이터 및 계산식

※ Period1 : 기간1

※ Period2 : 기간2

※ Period3 : 기간3

계산 : $A1 = \text{value} \text{의 Period1 기간 지수이동평균}$

$A2 = A1 \text{의 Period2 기간 지수이동평균}$

$TXAverage = A2 \text{의 Period3 기간 지수이동평균}$

UltimateOsc

설명 : Ultimate Oscillator 지표함수

함수 : $UltimateOsc(\text{Period1}, \text{Period2}, \text{Period3})$

※ Period1 : 기간1

※ Period2 : 기간2

※ Period3 : 기간3

계산 : $\text{SumTR1} = \text{TrueRange의 Period1 기간 합산}$

$\text{SumTR2} = \text{TrueRange의 Period2 기간 합산}$

$\text{SumTR3} = \text{TrueRange의 Period3기간 합산}$

$\text{Diff} = \text{종가} - \text{TrueLow}$

$\text{SumDiff1} = \text{Diff의 Period1 기간 합산}$

$\text{SumDiff2} = \text{Diff의 Period2 기간 합산}$

$\text{SumDiff3} = \text{Diff의 Period3 기간 합산}$

$\text{Diff1} = (\text{SumDiff1} / \text{SumTR1}) * 4$

$\text{Diff2} = (\text{SumDiff2} / \text{SumTR2}) * 2$

$\text{Diff3} = (\text{SumDiff3} / \text{SumTR3})$

$$\text{UltimateOsc} = (\text{Diff1} + \text{Diff2} + \text{Diff3}) / 7$$

VHF

설명 : Vertical Horizontal Filter 지표함수

함수 : VHF(Period)

※ Period : 기간

계산 : HCP = Period 기간 동안의 증가 최고값

LCP = Period 기간 동안의 증가 최저값

VHF = (HCP - LCP)의 절대값 / (증가-전봉증가)의 절대값의 Period 기간 합

VR

설명 : Volume Ratio 지표함수

함수 : VR(Period)

※ Period : 기간

계산 : value1 = 현재증가가 직전봉 증가보다 크면 현재거래량 아니면 0값

value2 = 현재증가가 직전봉 증가보다 작으면 현재거래량 아니면 0값

value3 = 현재증가가 직전봉 증가와 같으면 현재거래량 아니면 0값

$$\text{VR} = ((\text{Period 기간 value1 합계}) + (\text{Period 기간 value3 합계}/2)) /$$

$$((\text{Period 기간 value2 합계}) + (\text{Period 기간 value3 합계}/2)) * 100;$$

VROC

설명 : Volume Rate Of Change 지표함수

함수 : VROC(Period)

※ Period : 기간

계산 : (현재거래량 - Period 기간 이전봉 거래량) /

Period 기간 이전봉 거래량*100

WILLA

설명 : Williams' Accumulation/Distribution 지표함수

함수 : WILLA

계산 : A1 = 증가가 직전봉 대비 상승하면 증가에서 Truehigh를 뺀 값, 증가가

직전봉 증가 대비 하락하면 증가에서 TrueLow를 뺀 값

WILLA = A1의 차트 전체 누적

WILLR

설명 : Williams' %R

함수 : WILLR(Period);

※ Period : 기간

계산 : (Period 기간 최고가 - 현재종가) /

(Period 기간 최고가 - Period 기간 최저가) *100

WMA

설명 : 가중이동평균

함수 : WMA (value,Period);

※ value : 데이터 및 계산식

※ Period : 기간

계산 : $(P_1 \times n_1 + P_2 \times n_2 + \dots P_n \times n_n) / (n_1 + n_2 + n_3 \dots n_n)$

BullishEngulfing

설명 : 상승장악형 봉패턴

함수 : BullishEngulfing(Period)

※ Period : 추세파악을 위한 이동평균 기간

계산 : Period 기간 이동평균 하락.

현재봉 양봉, 직전봉 음봉.

현재봉 종가는 직전봉 시가보다 크고 현재봉 시가는 직전봉 종가보다 작음.

BearishEngulfing

설명 : 하락장악형 봉패턴

함수 : BearishEngulfing(Period)

※ Period : 추세파악을 위한 이동평균 기간

계산 : Period 기간 이동평균 상승

현재봉 음봉 직전봉 양봉

현재봉 종가는 직전봉 시가보다 작고 현재봉 시가는 직전봉 종가보다 큼

C_Doji

설명 : 십자봉 패턴

함수 : C_Doji(Percent)

※ Percent : 봉 길이 대비 몸통의 비율

계산 : 몸통의 길이가 봉의 길이 대비 Percent 이하

DarkCloud

설명 : 흑운형 패턴

함수 : DarkCloud(Period)

※ Period : 몸통길이의 평균 기간

계산 : 직전봉 양봉, 최근 Period 기간 몸통길이의 평균보다 큰 직전봉 몸통길이가
직전봉 고가-직전봉 종가나 직전봉 시가-직전봉 저가 보다 크고 현재봉
시가는 직전봉 고가보다 크고 종가가 직전봉 중심가격보다 큰 패턴

EveningStar

설명 : 석별형 패턴

함수 : EveningStar(Period)

※ Period : 몸통길이의 평균 기간

계산 : 현재봉은 음봉이고 2봉전은 양봉이고
직전봉의 종가와 시가가 2봉전의 종가보다 크고
2봉전 몸통길이는 Period기간 평균 몸통길이보다 크고
현재 종가가 2봉전 종가보다 작은 패턴

Hammer

설명 : 망치형 패턴

함수 : Hammer(Period,Tails)

※ Period : LRL 기간

※ tails : 몸통길이의 배수

계산 : 몸통의 하단이 중간 값보다는 높게 위치하고 종가와 시가는 같지 않으며
아래꼬리의 길이가 몸통길이의 tails배보다 길고 윗꼬리는 몸통길이보다 작고
LRL은 하향 중인 패턴

HangingMan

설명 : 교수형 패턴

함수 : HangingMan (Period,Tails)

※ Period : LRL 기간

※ tails : 몸통길이의 배수

계산 : 몸통의 하단이 중간 값보다는 높게 위치하고 종가와 시가는 같지 않으며
아래꼬리의 길이가 몸통길이의 tails배보다 길고 윗꼬리는 몸통길이보다 작고
LRL은 상승 중인 패턴

MorningStar

설명 : 석별형 패턴

함수 : MorningStar(Period)

※ Period : 몸통길이의 평균 기간

계산 : 현재봉은 양봉이고 2봉전은 음봉이고

직전봉의 증가와 시가가 2봉전의 증가보다 작고

2봉전 몸통길이는 Period기간 평균 몸통길이보다 크고

현재 증가가 2봉전 증가보다 큰 패턴

15) 선언

외부변수 선언문

Input , Inputs
외부로부터 입력 받을 변수를 선언
Input : Period1(10); Inputs : Period(10);

내부변수 선언문

var , vars, variable, variables
수식 내에서 사용될 변수를 선언
Var : value(0), cond(false), Str(" "); Vars : value(0), cond(false), Str(" "); Variable : value(0), cond(false), Str(" "); Variables : value(0), cond(false), Str(" ");

배열변수 선언문

Array , Arrays
배열 변수를 선언
Array : value[10](0),cond[10](false),str[10](""); Arrays : value[10](0),cond[10](false),str[10]("");

논리형 내부변수 선언

True , False
변수를 논리형으로 지정
Var : cond1(True); → cond1변수를 논리형으로 지정하고 초기값은 true로 지정 Var : cond1(false); → cond1변수를 논리형으로 지정하고 초기값은 true로 지정

사용자함수에서 외부변수를 논리형으로 지정

Logical TrueFalse	논리형으로 선언 이전값 참조 가능 내부에서 변수 값 변경 불가능	Input : cond(Logical); Input : cond(TrueFalse);
LogicalArray TrueFalseArray	논리형 배열 선언 이전값 참조 가능 내부에서 변수 값 변경 불가능	Input : cond[N](LogicalArray); Input : cond[N](TrueFalseArray);
LogicalArrayRef TrueFalseArrayRef	논리형 배열 참조 선언 이전값 참조 가능 내부에서 변수 값 변경 가능	Input : cond[N](LogicalArrayRef); Input : cond[N](TrueFalseArrayRef);
LogicalRef TrueFalseRef	논리형 참조 선언 이전값 참조 가능 내부에서 변수 값 변경 가능	Input : cond(LogicalRef); Input : cond(TrueFalseRef);
LogicalSeries TrueFalseSeries	논리형 Series 선언 이전값 참조 가능 내부에서 변수 값 변경 불가능	Input : cond(LogicalSeries); Input : cond(TrueFalseSeries);
LogicalSimple TrueFalseSimple	논리형 Simple 선언 이전값 참조 불가능 내부에서 변수 값 변경 불가능	Input : cond(LogicalSimple); Input : cond(TrueFalseSimple);

사용자함수에서 외부변수를 수치형으로 지정

Numeric	수치형으로 선언 이전값 참조 가능 변수 값 내부 변경 불가능	Input : value(Numeric);
NumericArray	수치형 배열 선언 이전값 참조 가능 변수 값 내부 변경 불가능	Input : value[N](NumericArray);
NumericArrayRef	수치형 배열 참조 선언 이전값 참조 가능 변수 값 내부 변경 가능	Input : value[N](NumericArrayRef);
NumericRef	수치형 참조 선언 이전값 참조 가능 변수 값 내부 변경 가능	Input : value(NumericRef);
NumericSeries	수치형 Series 선언 이전값 참조 가능 변수 값 내부 변경 불가능	Input : value(NumericSeries);

NumericSimple	수치형 Simple 선언 이전값 참조 불가능 변수 값 내부 변경 불가능	Input : value(NumericSimple);
---------------	---	---------------------------------

사용자함수에서 외부변수를 문자형으로 지정

String	문자형으로 선언 이전값 참조 가능 변수 값 내부 변경 불가능	Input : Str(String);
StringArray	문자형 배열 선언 이전값 참조 가능 변수 값 내부 변경 불가능	Input : Str[N](StringArray);
StringArrayRef	문자형 배열 참조 선언 이전값 참조 가능 변수 값 내부 변경 가능	Input : Str[N](StringArrayRef);
StringRef	문자형 참조 선언 이전값 참조 가능 변수 값 내부 변경 가능	Input : Str(StringRef);
StringSeries	문자형 Series 선언 이전값 참조 가능 변수 값 내부 변경 불가능	Input : Str(StringSeries);
StringSimple	문자형 Simple 선언 이전값 참조 불가능 변수 값 내부 변경 불가능	Input : Str(StringSimple);

16) 색상에약어

색상에약어	색	RGB	색상에약어	색	RGB
Black		0,0,0	Indigo		75,0,130
DarkGray		105,105,105	Purple		128,0,128
Gray		128,128,128	DarkMagenta		139,0,139
LightGray		211,211,211	Magenta		255,0,255
Silver		192,192,192	LightMagenta		255,128,255
Bwhite		128,128,128	DeepPink		255,20,147
DarkBlue		0,0,139	Pink		255,192,203
Blue		0,0,255	LightPink		255,182,193
LightBlue		173,216,230	Violet		238,130,238
Navy		0,0,128	Plum		221,160,221
DarkCyan		0,139,139	Lavander		230,230,250
Cyan		0,255,255	Coral		255,127,80
LightCyan		224,255,255	Chocolate		210,105,30
Teal		0,128,128	Salmon		250,128,114
Turquoise		64,224,208	Tomato		255,99,71
DarkGreen		0,100,0	Crimson		220,20,60
Green		0,128,0	DarkRed		139,0,0
LightGreen		144,238,144	RED		255,0,0
Lime		0,255,0	LightRed		255,204,203
Olive		128,128,0	DarkBrown		92,64,51
DarkYellow		155,135,12	Brown		165,42,42
Yellow		255,255,0	Maroon		128,0,0
LightYellow		255,255,224	White		255,255,255
Beige		245,245,220	LBlue		0,0,128
Gold		255,215,0	LCyan		0,128,128
Orange		255,135,0	LGreen		0,128,0
Khaki		240,230,140	LYellow		128,128,0
Ivory		255,255,240	LMagenta		128,0,128

17) 예약어

DEF	default	buy("b1",Onclose,def,2);
-----	---------	--------------------------

※ EL모드에서만 사용가능한 예약어

All	모든 진입 수량	Exitlong all shares next bar at market;
Bar	이전 값 참조(봉기준)	Close of 1bar Ago
Day	이전 값 참조(일수기준)	Close of 1day Ago
Days	이전 값 참조(일수기준)	Close of 1days Ago
Default	default	Plot1(var1,"plot1",default);
Limit	지정가	Sell next bar at high limit;
Point	포인트/원	Buy at low-1 point limit
Points	포인트/원	Buy at low-2 points limit
Share	수량	Exitlong 1 share next bar at market;
Shares	수량	Exitlong 2 shares next bar at market;

Stop	역지정가	Buy next bar at high stop;
Today	현재 바	Close of Today
Tomorrow	다음 바	Buy At Open Of Tomorrow+1 stop;

18) Skip Word

Skip Word는 EL모드에서만 사용가능

A	an	At	based	by
does	from	is	of	on
place	the	was		