

---

Visual Trader

# Manuale TS



# Trading System

---

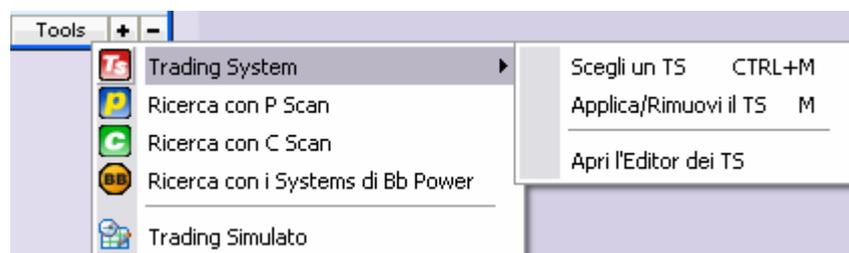
## Introduzione

Il **Trading System** con l'*Editor di formule programmabile*, è applicabile su tutto il periodo: sia su grafici storici storici, sia su quelli Intraday. Questa nuova funzionalità permette, in RealTime, la visualizzazione istantanea dell'Operazione di Trading da effettuare, delle operazioni compiute con percentuale di Resa Totale, attraverso il Report e anche graficamente con l'Equity Line.

Mostreremo nel dettaglio questi aspetti.

L'Editor TS si richiama dal menu Strumenti → Trading System.

Il TS invece si può attivare semplicemente dal pulsante TOOLS presente in basso a destra di ogni grafico:



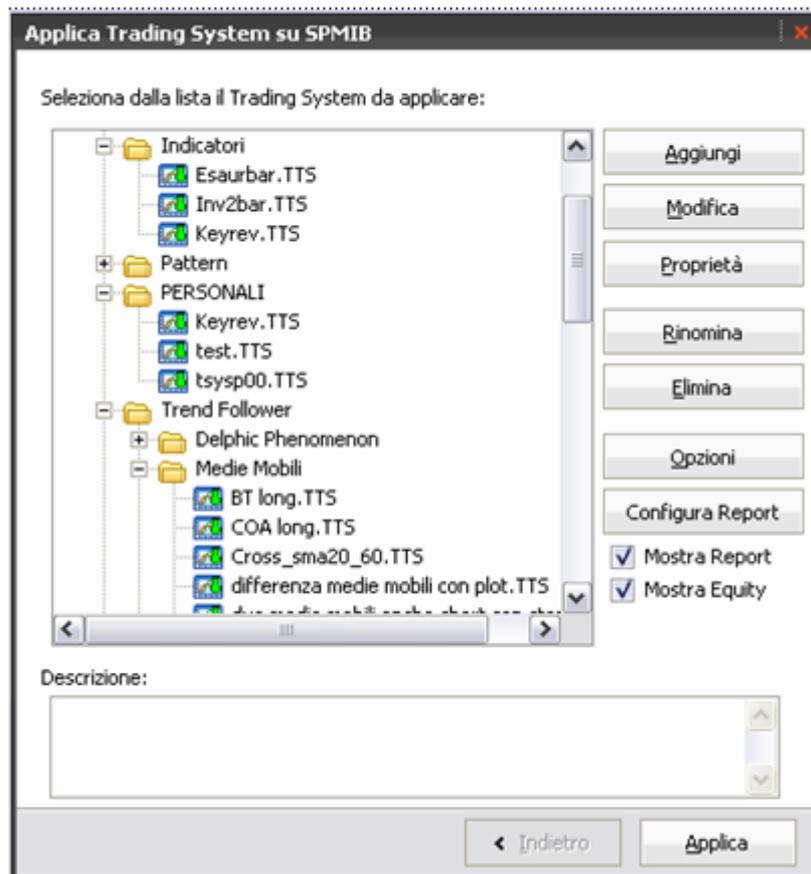
Il sottomenu comprende diverse funzionalità:

***Scegli un TS (Ctrl + M)***

***Applica/Rimuovi il TS scelto (M)***

***Apri l'editor dei TS***

***Scegli un TS***: Permette di aprire la lista di tutte le formule di TS presenti, inizialmente ci saranno solo quelle costruite da Traderlink.



Aggiungi: Apre l'editor TS per creare e quindi aggiungere alla lista un nuovo TS

Modifica: Apre l'editor TS per modificare il TS selezionato

Proprietà: Visualizza la maschera dell proprietà del TS selezionato

Rinomina: Rinomina il TS selezionato

Elimina: Caccia dalla lista il TS selezionato

Opzioni: Visualizza la maschera delle opzioni generali

Configura: Apre la maschera di configurazione del Report TS

	<p>La maschera di configurazione del Report TS, che si visualizza aprendo il menu "Opzioni" "Configura Report"(oppure dall'ultimo pulsante nella barra dei menu), permette di impostare la modalità di operare distinta in:</p> <p><i>Capitale fisso:</i> nel caso in cui venga investito un valore costante. La cifra è il multiplo del valore dell'azione più vicino al valore impostato;</p> <p>oppure nel caso di <u>Derivati</u></p> <p><i>Num. Azioni Fisso:</i> il numero di azioni investito per ogni singola operazione. In questo caso possiamo impostare anche il valore <i>Euro a punto:</i> valore di ogni singolo punto;</p>
--	--

Le *Commissioni*; è possibile stabilire una percentuale sulle commissioni oppure una cifra minima e una massima, in base a cui verrà calcolato il valore percentuale delle commissioni, inoltre possiamo stabilire una cifra fissa che esprime in maniera univoca il prezzo delle commissioni;

*Esportazione Automatica Dati:* selezionando questa opzione possiamo esportare i dati relativi a tutte le operazioni dei Report oppure tutti i dati relativi alla Equity Line in documenti di testo importabile da Excel. I file chiamati rispettivamente verranno salvati direttamente nella cartella Vtrader posizionata solitamente in C: .

Prima possiamo scegliere il *Periodo di Analisi*.

A questo punto abbiamo tre diversi modi per applicare la formula, mostrarne il report (dopo avere selezionato la formula dalle lista).

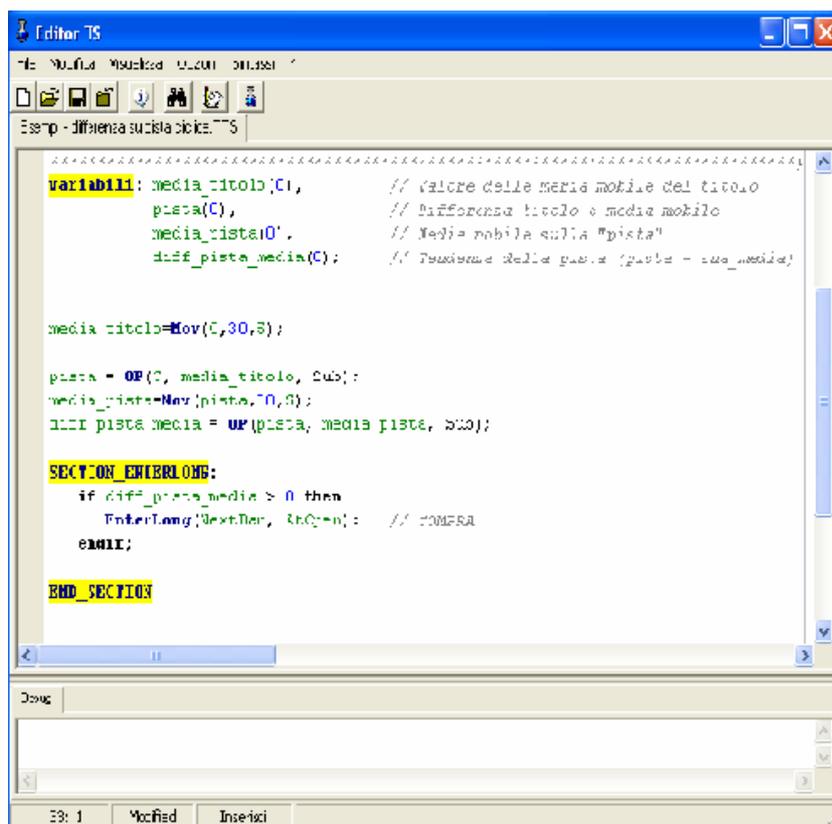
Primo metodo: Menu principale Trading System "Applica/Rimuovi il TS scelto" e "Report Dettaglio Operazioni TS "

Secondo metodo: Pulsanti da tastiera "M" e "Shift + M"

Terzo metodo: Bottoniera sotto la lista principale di tutti i TS.

**Apri L'Editor dei TS:** Apre l'EditorTS tramite cui è possibile creare le tue Strategie di Trading, o modificare quelle già inserite, attraverso un linguaggio di programmazione, molto simile ad altri sistemi di Trading.

La sintassi del linguaggio è spiegata nella guida per l'utente (comando F2 oppure menu info guida per l'utente).



L'editorTS comprende un menu principale per accedere a tutte le funzioni e una barra di strumenti che comprende le funzioni più utilizzate.

Menu File: da cui è possibile creare un nuovo TS, aprire un TS già esistente visualizzare le ultime formule, salvare, visualizzare le informazioni, chiudere tutte le formule;

Menu Modifica: comprende le principali funzioni per modificare il testo della formula, quali, annullamento e la ripetizione dell'ultima digitazione, il "copia taglia incolla", la funzione trova, sostituisci e la selezione;

Menu Visualizza: prevede di personalizzare l'interfaccia dell' editorTS; l'utente può scegliere se visualizzare o meno la barra di stato(in basso all'EditorTS) e il pannello di debug.che mostra durante il check della sintassi se ci sono degli errori;

Menu Sintassi: Effettua il check della sintassi(tasto funzione F5) e riporta il risultato nel pannello di debug.

? : Mostra la versione e le informazioni generali sull'editorTS. E' incluso anche un manuale interno che prevede la definizione di alcune regole sintattiche e la definizione di tutte le funzioni che si possono applicare.

---

## Analisi del Report

La maschera del Report si divide in:

**Report**

**Operazioni**

**Equity Line**

ESENPI - MEDIA MOBILE ANCHE SHORT (18.09.36): STMICROELECTRONICS			
Report   Operazioni   Equity Line			
<b>Resa</b>	Punti		
Assoluta Totale:	0	0,16 %	
Annuale Stimata:	0	ND	
Rapporto Resa/Rintracc:	40		
<b>Analisi Operazioni:</b>			
Effettuate:	253		
Positive:	107	49,43 %	
Negative:	135		
Consecutive Positive:	10		
Consecutive Negative:	8		
Vigliore:		0,27 %	
Peggiora:		-0,15 %	
<b>Analisi Operazioni</b>			
Massimo Run Up:		0,43 %	
Massimo Draw Down:		-0,15 %	
Draw Down Medio:		-0,04 %	
<b>Efficienza:</b>			
Velocità di Entrata:		72,17 %	
Velocità di Uscita:		61,53 %	
Velocità complessiva:		27,53 %	
Tempo "Amercato":		98,82 %	
<b>Profitto</b>			
Netto			47,51 €
%:		0,16 %	
% Annuale Stimata:		ND	
Medio:		0,00 %	
(Ultima Operazione Aperta):			73,70 €
<b>Capitale</b>			
Iniziale:			31.007,00 €
Finale:			31.047,51 €
Reintegr. Necessari:			257,94 €
Commissioni Pagate:			0,00
<b>Rendimento</b>			
Valore Medio di Profitto:			18,79 €
Valore Medio di Perdita:			-15,01 €
Rapporto Medio Prof/Perd:			1,04
Somma dei Profitti:			2.443,02 €
Somma delle Perdite:			-2.395,50 €
Rapporto Somme Prof/Perd:			1,02

**Dati provvisorio, ultima operazione non conclusa**

## Report

### Resa:

Assoluta Totale: netto delle commissioni espressa in punti nel caso di derivati, e in percentuale;

Annuale Stimata: stima del netto medio delle commissioni espressa in punti e in percentuale (è la Resa Assoluta Totale divisa per 12);

Rapporto Resa/Rintracc: calcola il rapporto tra il rendimento percentuale e la media dei ritracciamenti (ovvero variazioni negative della equity line). Il rendimento percentuale è pesato tenendo conto del numero di campioni a disposizione, in pratica più sono i campioni a disposizione maggiore è l'influenza del rendimento (sia positiva che negativa).

Questo indicatore è di grandissima importanza, in quanto permette di valutare rapidamente un TS (anche se in maniera sommaria) prima ancora di prendere in esame tutti gli altri parametri numerici del report. Tenete d'occhio quindi questo numero, scoprirete che c'è una forte correlazione tra il suo valore e la valutazione globale complessiva di un TS.

### Operazioni:

Effettuate: numero totale di operazioni compiute intese come apertura e chiusura

$Operazioni\ Concluse = Num.\ Tot.\ Op / 2$  oppure

$Operazioni\ Concluse = (Num.\ Tot.\ Op / 2) - 1$  (se l'ultima operazione non è stata chiusa);

Positive: numero totale di operazioni effettuate, concluse con % positiva al "lordo" delle commissioni (comprese le commissioni), il valore in percentuale

esprime il rapporto tra le operazioni positive e tutte le operazioni, definisce la qualità del TS.

**Percentuale di Profitto** =  $(op. \text{ positive } / tot) * 100$

Se è maggiore o uguale al 50% è di colore verde. Se è minore o uguale al 50% è di colore rosso;

Negative: numero totale di operazioni effettuate, concluse con % negativa al “lordo” delle commissioni (comprese le commissioni);

Consecutive Positive: numero massimo delle operazioni positive consecutive

Consecutive Negative: numero massimo delle operazioni negative consecutive

Queste quantità sono espresse anche nella lista delle operazioni, in cui sono più visibili;

Migliore: esprime in percentuale l'operazione che ha avuto maggiore guadagno (consultare anche la lista delle operazioni in corrispondenza della colonna “%Tot”);

Peggior: esprime in percentuale l'operazione che ha avuto maggiore perdita (consultare anche la lista delle operazioni in corrispondenza della colonna “%Tot”);

#### **Analisi Operazioni:**

Massimo Run Up: In caso di Long (intervallo compra/vendi) rappresenta l'escursione verso l'alto ovvero il guadagno ipotetico massimo, ottenuto se avessi venduto nel punto di massimo della curva dal punto di acquisto; In caso di Short (intervallo vendi/compra) il massimo Run Up rappresenta l'escursione verso il basso ovvero il massimo guadagno ipotetico ottenuto se avessi comprato (ricopertura del Short) nel punto di minimo della curva;

Massimo Draw Down: è il parametro opposto al Massimo Run Up. In caso di Long (intervallo compra/vendi) rappresenta l'escursione verso il basso ovvero la massima perdita ipotetica, ottenuta se avessi venduto nel punto di minimo della curva dal punto dopo l'acquisto e prima della vendita programmata dal TS; In caso di Short (intervallo vendi/compra) il massimo Draw Down rappresenta l'escursione verso l'alto ovvero la perdita ipotetica massima ottenuta se avessi comprato nel punto di massimo della curva prima della ricopertura programmata dal TS;

Draw Down Medio: esprime la media in percentuale rispetto a tutti i Draw Down;

#### **Analisi Efficienza:**

Media di Entrata: somma dell'efficienza di entrata di ogni singola operazione divisa per il numero di operazioni, per la definizione di efficienza vedi nella descrizione dei campi della lista delle operazioni;

Media di Uscita: somma dell'efficienza di uscita di ogni singola operazione divisa per il numero di operazioni, per la definizione di efficienza vedi nella descrizione dei campi della lista delle operazioni;

Media Complessiva: somma dell'efficienza di entrata e di uscita di ogni singola operazione divisa per il numero di operazioni, per la definizione di efficienza vedi nella descrizione dei campi della lista delle operazioni;

Tempo “A Mercato”: rapporto tra la somma delle durate delle singole operazioni (espresse in periodi) e il numero di totale dei periodi

**Profitto:**

Netto: guadagno o perdita effettiva del TS;

%: guadagno complessivo rispetto al capitale investito, può essere calcolato solo se si stabilisce un investimento a capitale fisso;

% Annuale Stimato: resa complessiva divisa per il numero di anni, può essere calcolata solo se il TS è stato applicato a grafici storici(servizio EndOfDay);

Medio: somma dei guadagni espressa in percentuale di ogni singola operazione divisa per il numero totale delle operazioni.

Ultima Operazione Aperta: guadagno o perdita calcolato sull'ultima operazione che realmente non è stata ancora chiusa;

**Capitale:**

Iniziale: la cifra iniziale, espressa in euro, prima di effettuare operazioni, può essere impostata arbitrariamente nelle configurazioni;

Finale: esprime la somma del Capitale iniziale e del Profitto Netto;

Reintegro Necessario: la cifra, espressa in euro, necessaria per effettuare operazioni, anche questa quantità può essere impostata dall'utente tramite i parametri di configurazione del TS (nella maschera di configurazione è denominato capitale fisso);

Commissioni Pagate: somma totale delle spese per le commissioni della banca, si possono definire nella maschera di configurazione;

**Rendimento:**

Valore Medio di Profitto: somma espressa in euro di tutte i profitti divisa per il numero delle operazioni positive;

Valore Medio di Perdita: somma espressa in euro di tutte le perdite divisa per il numero delle operazioni negative;

Media dei Guadagni: somma espressa in euro di tutti i guadagni divisa per il numero delle operazioni positive;

Rapporto Medio Prof/Perd: rapporto tra la media dei guadagni e la media delle perdite, se è maggiore di 1 allora significa che il TS è vantaggioso(il guadagno è superiore);

Somma dei Profitti: :somma espressa in euro di tutti i guadagni;

Somma delle Perdite: somma espressa in euro di tutte le perdite;

Rapporto Somme Prof/Perd: rapporto tra la somma dei guadagni e la somma delle perdite, se è maggiore di 1 allora significa che il TS è vantaggioso(il guadagno è superiore);

N.B.: il valore della singola operazione viene stabilito usando il sottomultiplo del valore della singola azione più vicino al valore dell'operazione.

Esempio. Azione a 20€ e il capitale investito è 90€ allora l'investimento viene posto uguale a 80€.

Questa maschera permette anche di esportare automaticamente i dati del Report e dell'Equity Line.

## Operazioni

"T" : E' il tipo di operazione, può essere Long (freccia verde) o Short (freccia rosa);

Data1, Ora1, Prezzo1: sono rispettivamente la data l'ora e il prezzo di acquisto se l'operazione è di tipo Long, di vendita se è di tipo Short;

Data2, Ora2, Prezzo2: sono rispettivamente la data l'ora e il prezzo di vendita se l'operazione è di tipo Long, di acquisto se è di tipo Short;

%Tot: Percentuale di guadagno o di perdita dell'operazione;

Portaf: differenza relativa al valore del Capitale iniziale;

EffEnt: Nell'intervallo compra/vendi esprime in percentuale il rapporto tra la differenza di massimo del grafico e valore di acquisto e la differenza tra massimo e minimo del grafico nell'intervallo considerato;

In caso contrario (intervallo vendi/compra) esprime in percentuale il rapporto tra la differenza di minimo del grafico e valore di vendita e la differenza tra massimo e minimo del grafico nell'intervallo considerato;

**EffEnt = |(Max - Val.Acq.) / (Max - Min)| \* 100** intervallo compra/vendi

**EffEnt = |(Min - Val.Ven.) / (Max - Min)| \* 100** intervallo vendi/compra

EffUsc: Nell'intervallo compra/vendi esprime in percentuale il rapporto tra la differenza di massimo del grafico e valore di vendita e la differenza tra massimo e minimo del grafico nell'intervallo considerato;

In caso contrario (intervallo vendi/compra) esprime in percentuale il rapporto tra la differenza di minimo del grafico e valore di vendita e la differenza tra massimo e minimo del grafico nell'intervallo considerato;

**EffUsc = |(Max - Val.Ven.) / (Max - Min)| \* 100** intervallo compra/vendi

**EffUsc = |(Min - Val.Acq.) / (Max - Min)| \* 100** intervallo vendi/compra

EffTot: Nell'intervallo compra/vendi esprime in percentuale il rapporto tra la differenza di massimo del grafico e valore di acquisto e la differenza tra massimo e minimo del grafico nell'intervallo considerato;

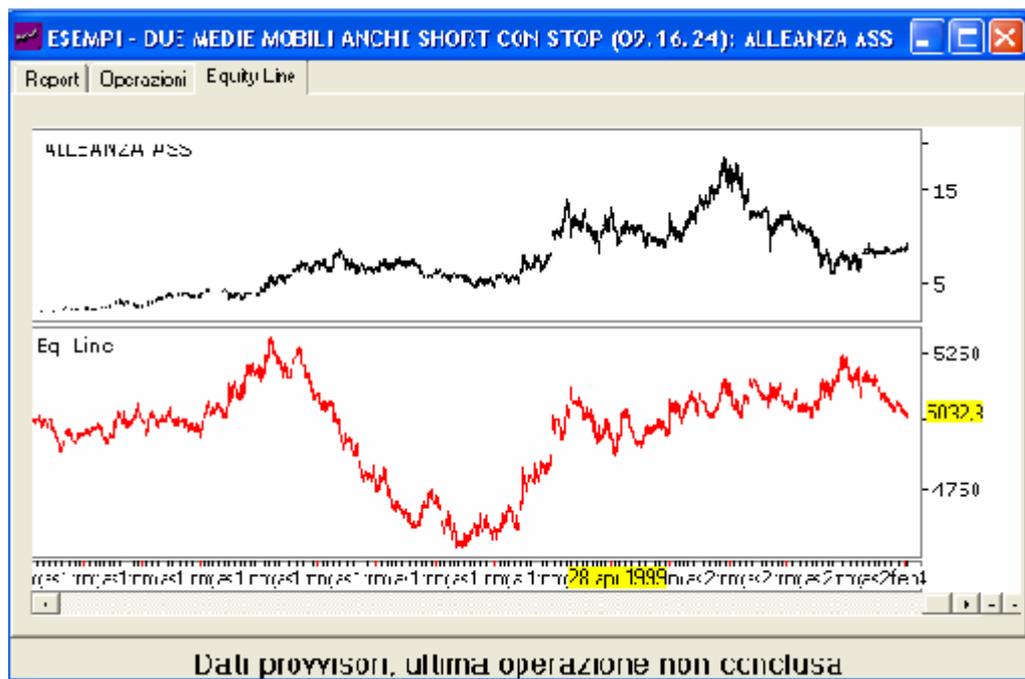
In caso contrario (intervallo vendi/compra) esprime in percentuale il rapporto tra la differenza di minimo del grafico e valore di vendita e la differenza tra massimo e minimo del grafico nell'intervallo considerato;

**Eff. Tot. = |(Val.Acq. - Val.Ven.) / (Max - Min)| \* 100** intervallo compra/vendi

**Eff. Tot. = |(Val.Ven. - Val.Acq.) / (Max - Min)| \* 100** intervallo vendi/compra.

## Equity Line

L'ultima scheda è riservata alla parte grafica.



Nella prima zona è mostrato il grafico e nell'area sottostante c'è l' Equity Line.

La curva Equity Line rappresenta l'andamento del portafoglio calcolato nei punti in cui il TS era a mercato. Per la rappresentazione del grafico si è pensato di farlo partire da un valore iniziale arbitrario posto uguale a 1000.

Posizionandosi con il mouse sul grafico del titolo o sull' Equity Line viene mostrato esattamente il valore sulla destra nella scala valori e la data in basso.

E' possibile effettuare uno zoom dei grafici tramite i pulsanti "+" e "-" posizionati in fondo a sinistra inoltre è possibile effettuare un ridimensionamento delle 2 zone, posizionandosi nella linea di confine delle due e effettuando un trascinamento del mouse.

---

## Guida per l'utente(Sintassi)

### Introduzione

EditorTS ti permette di creare le tue Strategie di Trading, attraverso un linguaggio di programmazione, molto simile ad altri sistemi di Trading.

### Come creare un TS

Per creare un Trading System seguire I seguenti passi:

- 1) Aprire il programma EditorTS, presente nella cartella VTRADER. Cliccare sul bottone "Nuovo" oppure, "File" ->"Nuovo";
- 2) Inserire una formula all'interno dell'editor;

- 3) Salvare la formula nella cartella delle formule;
- 4) Eseguire il check della sintassi, cliccare su “Sintassi” -> “Check Sintassi”, oppure tasto F5;

Questa operazione permette di verificare se la sintassi risulta corretta.

- 5) Per visualizzare il Trading System andare su Visual Trader, aprire un grafico, andare sul menu “Analisi” -> “Trading System”, oppure premere il tasto M, per eseguire il Trading System, con SHIFT+M viene anche visualizzato un Report delle Operazioni.

## Sintassi del Linguaggio

I paragrafi della Sintassi comprendono:

Commenti, Istruzioni e Delimitatori, Tipo di Dati e Variabili, Assegnazione Istruzioni, Operazioni, Istruzioni Condizionali e Sezioni.

### *Commenti*

Puoi usare i **commenti** per mettere delle note sul tuo codice. I commenti non hanno alcun peso sull'esecuzione dello Script. Ci sono 2 forme di commenti.

#### **Commenti a Blocchi**

Usare le parentesi graffe per racchiudere un blocco di commento.

*Esempio*

```
{ Questo è un commento e non sarà eseguito }
```

#### **Commenti a linea singola**

Usare I caratteri "//" per creare singole linee di commenti.

*Esempio*

```
//Esempio di Commento in linea
```

### *Istruzioni e Delimitatori*

Il TS esegue le Istruzioni in ordine, dall'alto verso il basso. Utilizzare le Istruzioni Condizionali per controllare l'esecuzione del flusso.

#### **Punto e Virgola “;”**

Ogni Istruzione deve terminare con il punto e virgola finali (;).

*Esempio*

Vedere anche Istruzioni Condizionali.

### *Tipo di Dati e Variabili*

#### **Tipi di Dati**

Per definire un tipo di dato, basta impostare il valore di default nella dichiarazione delle Variabili.

Nota: Al momento tutti i valori sono impostati come **float**.

### *float*

I valori a virgola mobile vanno da  $1.5 \times 10^{-45}$  a  $3.4 \times 10^{38}$ , e mantengono da 7 ad 8 cifre significative.

### *boolean*

Un valore uguale a -1 (**=true**) altrimenti (**=false**).

### **Come dichiarare una Variabile**

In alto alla formula utilizzare **Var**: seguito dalle variabili, separati da virgole, per dichiarare le variabili utilizzate nello script. La dichiarazione termina con un punto e virgola “;”.

*Esempio*

**Var:** miamedia(0), miocontatore(0), miopar(0), miavar(0);

### **Come dichiarare una Variabile d’Input**

In alto alla formula utilizzare **Input**: seguito dalle variabili, separati da virgole, per dichiarare le variabili utilizzate nello script. La dichiarazione termina con un punto e virgola “;”.

Dichiarando le variabili nella sezione **Input**, ogni volta che si esegue il Trading System, viene richiesto il valore della variabile.

*Esempio*

**Input:** mioparametro(3), altravar(0);

In questo caso *mioparametro* avrà come default “3”, nella finestra di Input.

## **Assegnazione Istruzioni**

Utilizzerai l’assegnazione delle Istruzioni per impostare dei valori nelle tue variabili.

Per assegnare utilizza l’operatore di assegnazione (=).

*Esempio*

**Var:** miocontatore(0); miocontatore = miocontatore + 1;

## **Operazioni**

Ci sono differenti tipi di **operazioni** che è possibile calcolare; matematiche, stringhe, e booleani.

### **Operazioni Matematiche**

Puoi utilizzare le operazioni matematiche classiche: addizione(+), sottrazione(-), moltiplicazione (\*), e divisione (/). Le moltiplicazioni e divisioni sono valutate per prima, altrimenti vengono valutate da sinistra a destra. Consigliamo di utilizzare le parentesi per definire l’ordine delle operazioni da valutare.

*Esempio*

**Var:** x(0); x = 1 / 2; x = x \* 5 + 1; x = ( x - 5 ) / ( x \* 2 );  
x = x - ( 2 / ( 3 \* x ) );

Vedere la lista delle Funzioni per utilizzare le funzioni matematiche più complesse.

### **Operazioni Booleani**

Puoi utilizzare operazioni booleane (true, false) per assegnare valori di Vero (true) o Falso (false).

## ***Istruzioni Condizionali***

Le Istruzioni **condizionali** permettono di controllare l'esecuzione del flusso nelle vostre formule. Utilizzare le Istruzioni **if**, **then**, **else** ed **Endif** per questo scopo.

### **Istruzioni If/Then/Endif**

Usare le Istruzioni **if/then/Endif**; per fare test logici, se una condizione risulta Vera (true) oppure no (false).

*Esempio*

```
Var: x(0);      x = 10;
if x > 10 then
  x = x + 1;     { non verrà eseguito }
endif;
```

```
if x <= 10 then
  x = x * 2;     { verrà eseguito }
endif;
```

```
if ( x = 20 ) or ( x = 10 ) then
  x = x / 3;     { verrà eseguito }
endif;
```

Notare che un'Istruzione termina sempre con **endif** seguito dal punto e virgola “;”.

### **Istruzione Else**

Puoi usare l'Istruzione **else** per eseguire delle Istruzioni se la condizione **if/then** risulta falsa.

*Esempio*

```
Var: x(0);      x = 10;
if x = 5 then
  x = x * 20;
else
  x = x / 20;
endif;
```

### **If/Then If/Then/Else/Endif Nidificati**

Puoi “nidificare” uno o più Istruzioni **if/then/else/endif** dentro un'altro.

*Esempio*

```
Var: x(0); x = 10;
```

```

if x = 10 then
  if x * 2 < 20 then
    x = x / 3;
    x = x + 2;
  endif;
endif;

```

```

if x < 2 then
  x = x * 10;
  x = x - 5;
else
  x := ( x + 1 ) / x;
  x := x * 2;
endif;

```

### **Istruzione ElseIf**

E' infine possibile utilizzare istruzioni **ElseIf** per effettuare ulteriori controlli senza la necessità di specificare istruzioni

**If/Then** annidate, rendendo così il proprio codice più conciso e facile da leggere.

*Esempio*

```

if cond = 0 then
  x = 10;
else
  if cond = 1 then
    x = 20;
  else
    x = 30;
  endif;
endif;

```

potrebbe essere scritto, in modo analogo ma più chiaro, così:

```

if cond = 0 then
  x = 10;
elseif cond = 1 then
  x = 20;
else
  x = 30;
endif;

```

### **Funzione IIF**

E' infine possibile utilizzare la funzione **IIF** per assegnare un valore booleano (vero o falso) in base alla condizione (1° parametro).

```
Var: Miorisultato;  
Miorisultato = iif(miacondizione, risultato_se_vero, risultato_se_falso);  
//Questo equivale a scrivere a:  
if miacondizione then  
    Miorisultato = risultato_se_vero;  
else  
    Miorisultato = risultato_se_falso;  
endif;
```

#### *Esempio*

```
Var es1, es2;  
es1 = iif(C > C[1], true, false);  
es2 = iif(C > MOV(C, 10, S), C > C[1], C > C[2]);  
.....
```

### **Sezioni (Section)**

Le sezioni sono un parte importante per suddividere le condizioni di acquisto/vendita.

Una volta che la condizione di Entrata/Uscita viene verificata, attraverso le funzoni EnterLong, EnterShort, ExitLong ed ExitShort, il ciclo passa alla Barra successiva, pertanto non vengono ammesse, in questa versione, operazioni multiple.

**Var:** { Parte contenente le variabili}

prova(0);

{Dichiarazione degli Indicatori}

**SECTION\_ENTERLONG:**

{ Sezioni contenente le condizioni di **Entrata** posizione **Long**}

**END\_SECTION** //Fine Sezione

**SECTION\_EXITLONG:**

{ Sezioni contenente le condizioni di **Uscita** da una posizione **Long**}

**END\_SECTION**

**SECTION\_ENTERSHORT:**

{ Sezioni contenente le condizioni di **Entrata** posizione **Short**}

END\_SECTION

SECTION\_EXITSHORT:

{ Sezioni contenente le condizioni di **Uscita** da una posizione **Short** }

END\_SECTION

---

## Accesso ai Dati

### **AvgPrice** oppure **A**

**AvgPrice: float**

Descrizione

Ritorna il prezzo medio della barra attuale. Il prezzo medio è  $(\text{Close} + \text{Open} + \text{High} + \text{Low}) / 4$ .

Per ritornare i valori precedenti e' necessario utilizzare le parentesi quadre:

**AvgPrice[N]** : Ritorna il valore del prezzo medio individuale della barra di N periodi fa.

*Esempio*

```
if AvgPrice > AvgPrice[1] then
```

```
    { La barra attuale ha il prezzo medio piu' in alto della barra precedente (1) }
```

```
    ...
```

```
endif;
```

### **Close** oppure **C**

**Close: float**

Descrizione

Ritorna il prezzo di chiusura della barra attuale.

Per ritornare i valori precedenti e' necessario utilizzare le parentesi quadre:

**Close[N]**: Ritorna il valore del prezzo di chiusura individuale della barra di N periodi fa.

*Esempio*

```
if C > C[1] then
```

```
    { La barra attuale ha chiuso piu' in alto della barra precedente (1) }
```

```
    ...
```

```
endif;
```

## Day oppure D

**Day: integer**

### Descrizione

Ritorna il valore della data della barra attuale.

Per ritornare i valori di date precedenti e' necessario utilizzare le parentesi quadre:

**Day[N]**: Ritorna la data della barra di N periodi fa.

## High oppure H

**High: float**

### Descrizione

Ritorna il prezzo massimo della barra attuale.

Per ritornare i valori precedenti e' necessario utilizzare le parentesi quadre:

**High[N]**: Ritorna il valore del prezzo massimo individuale della barra di N periodi fa.

### *Esempio*

```
if H > H[1] and H[1] > H[2] then
```

```
    { Trend in salita: La barra attuale ha il massimo piu' alto della precedente, e la  
    barra precedente ha il massimo piu' alto di quella di 2 periodi fa' .}
```

```
    ...
```

```
endif;
```

## IsBlack oppure B

**IsBlack: boolean**

### Descrizione

Ritorna True se la chiusura è inferiore all'apertura della barra attuale del Titolo, cioè:

= True se  $Close < Open$ .

### *Esempio*

```
if (B = true) and (B[1] = true) then
```

```
    { La barra attuale e quella precedente sono nere }
```

```
    ...
```

```
endif;
```

## IsWhite oppure W

**IsWhite: boolean**

### Descrizione

Ritorna True se la chiusura è superiore o uguale all'apertura della barra attuale del Titolo, cioè:

= True se  $Close \geq Open$ .

### *Esempio*

if (**W** = true) and (**W**[1] = true) then

{ La barra attuale e quella precedente sono bianche }

...

endif;

## **LOW** oppure **L**

**Low: float**

### Descrizione

Ritorna il prezzo minimo della barra attuale.

Per ritornare i valori precedenti e' necessario utilizzare le parentesi quadre:

**Low[N]**: Ritorna il valore del prezzo minimo individuale della barra di N periodi fa.

### *Esempio*

```
if L > L[1] then
```

```
    { La barra attuale ha il minimo piu' alto della barra precedente (1) }
```

```
    ...
```

```
endif;
```

## **Open** oppure **O**

**Open: float**

### Descrizione

Ritorna il prezzo di apertura della barra attuale.

Per ritornare i valori precedenti e' necessario utilizzare le parentesi quadre:

**Open[N]**: Ritorna il valore del prezzo di apertura individuale della barra di N periodi fa.

### *Esempio*

```
if O > O[1] then
```

```
    { La barra attuale ha l'apertura piu' alta della barra precedente (1) }
```

```
    ...
```

```
endif;
```

## **Range** oppure **R**

**Range: float**

### Descrizione

Ritorna il range del giorno, cioe' Range è uguale al massimo meno il minimo della barra attuale (H - L).

Per ritornare i valori precedenti e' necessario utilizzare le parentesi quadre:

**Range[N]**: Ritorna il range di prezzo della barra di N periodi fa.

### *Esempio*

```
if R > R[1] then
    { La barra attuale ha un range piu' alto della barra precedente (1) }
...
endif;
```

## Volume oppure **V**

**Volume: float**

### Descrizione

Ritorna il volume della barra attuale.

Per ritornare i valori precedenti e' necessario utilizzare le parentesi quadre:

**Volume**[N]: Ritorna il valore del volume individuale della barra di N periodi fa.

### *Esempio*

```
if V > V[1] then
    { La barra attuale ha il volume piu' alto della barra precedente (1) }
...
endif;
```

## Parole Chiavi

Tutte le Parole Chiavi usate nel TS, non possono essere dichiarate come Variabili dell'Utente:

Parole chiavi ad 1 carattere: **A, C, O, L, H, R, V, W**.

---

# Funzioni

## Funzioni di Accesso ai Dati

### CurrentBar

**CurrentBar: float**

### Descrizione

Ritorna il numero di Barra corrente, a partire dalla prima barra utile per fare Trading.

Pertanto la Barra 1 è la prima barra in cui è possibile fare operazioni.

### *Esempio*

```
if CurrentBar = 1 then
```

```
        // Inizializza delle variabili
        ....
    endif;
```

## GetTick

**GetTick: float**

### Descrizione

Ritorna il Valore di Tick del Titolo. Il Valore di Tick è il piu' piccolo movimento di prezzo incrementale che un contratto del titolo puo' fare.

## GetDataType

**GetDataType: integer**

### Descrizione

Ritorna il tipo di dati attualmente utilizzato:

### *Esempio*

```
    if GetDataType = Intraday then
        // Sono in Intraday
    endif;
    if GetDataType = EndOfDay then
        // Sono in EndOfDay
    endif;
```

## GetCompression

**GetCompression: integer**

### Descrizione

Ritorna il tipo di compressione dati utilizzata:

### In Intraday:

Ritorna il numero di secondi.

### In EndOfDay:

Ritorna:

Daily -> giornaliera

Week -> settimanale

Month -> mensile

### *Esempio*

```
    if GetDataType = Intraday then
        if GetCompression = 60 then
            // Compressione ad 1 minuto
```

```

        // ...
    endif;
endif;
if GetDataType = EndOfDay then
    // Sono in EndOfDay
    if GetCompression = Daily then
        // Sono in compressione Giornaliera
    // ...
    endif;
    if GetCompression = Week then
// Sono in compressione Settimanale
// ...
    endif;
endif;

```

## GetValues

**GetValues (tipo, numperiodi, open, min, max, close):boolean**

### Descrizione

Ritorna i valori di apertura, minimo, massimo, chiusura, del periodo selezionato. La funzione ritorna True quando i valori vengono aggiornati.

### Tipo:

bars: periodo in base alla compressione scelta

days: calcola i giorni.

hours: calcola le ore.

minutes: calcola i minuti.

### *Esempio*

```
Var: mioopen, miomin, miomax, mioclose, giornocambiato(false);
```

```
{ Ottieni i valori del giorno precedente (Utilizzato solo con grafici Realtime)}
```

```
giornocambiato = GetValues(days, 1, mioopen, miomin, miomax, mioclose);
```

```
if giornocambiato then
```

```
    // calcola qualcosa in base ai valori del gg precedente, utilizzando, in questo
```

```
    //esempio, le variabili (mioopen, miomin, miomax, mioclose)
```

```
endif;
```

```
// Plotto il valore della chiusura di ieri
```

```
PlotChart(mioclose, 0, green, solid, 2);
```

### *Esempio2*

```
{ Plotta i minimi e massimi di 5 giorni fa (con Grafico End-Of-Day)}
```

```
GetValues(weeks, 1, mioopen, miomin, miomax, mioclose);
```

```
// Plotto il minimo a 5 giorni
PlotChart(miomin, 0, green, solid, 2);
// Plotto il massimo a 5 giorni
PlotChart(miomax, 0, blue, solid, 2);
```

### Esempio3

```
{*****}
* Questo è un Esempio dove vengono calcolati i Pivot
*****}
```

```
Var: H_data2, Range_Data2, O_data2, L_data2, C_data2, MP(0), PivPnt(0),
Resistance1(0), Support1(0), NumCont(0),
    ggcambiato(false);
```

```
// Estrai i valori di "1" giorno fa.
```

```
ggcambiato = GetValues (days, 2, O_data2, L_data2, H_data2, C_data2);
```

```
if ggcambiato then
```

```
    Range_Data2 = H_data2 - L_data2;
```

```
    // calcola i Pivot
```

```
    PivPnt = (H_data2 + L_data2 + C_data2)/3;
```

```
    Resistance1 = (PivPnt * 2) - L_data2;
```

```
    Support1 = (PivPnt * 2) - H_data2;
```

```
endif;
```

```
// Disegna i Pivot (Supporto e Resistenza)
```

```
PlotChart (L_DATA2, 0, green, solid,2);
```

```
PlotChart (H_DATA2, 0, red, solid,2);
```

### EOD.x

**EOD.x: float;**

#### Descrizione

Con l'utilizzo della parola chiave EOD, vengono estrapolati i valori Daily, quando applichiamo un TS ad un grafico Intraday.

Dove *x* può essere una lettera chiave:

A, C, O, L, H, R, V.

#### **Note:**

Utilizzato solo con grafici Intraday.

*Esempi:*

EOD.H           -> Ritorna il massimo di oggi  
EOD.C[1]       -> Ritorna l'ultimo valore di chiusura del giorno precedente  
EOD.L[2]       -> Ritorna il minimo di 2 giorni fa.

**Nota:**

Questi valori non possono essere utilizzati come ingresso degli oscillatori/indicatori, ma solo con operazioni matematiche.

*Esempio (con Grafico Intraday)*

```
{*****  
** Disegno dei Pivot Point  
** Esempio dell'utilizzo di EOD, GetValues e DrawHLinePer  
*****}
```

```
Var: previousH,previousL,previousC,  
    Pivot,R1,S1, R1mid,S1mid, prova, cambiatogiorno,  
    lin1,lin2,lin3,lin4,lin5,  
    indperiodo, numper, ggini, primavolta(-1);
```

```
previousH=EOD.H[1];  
previousL=EOD.L[1];  
previousC=EOD.C[1];
```

```
cambiatogiorno = GetValues(days, 1, prova, prova, prova, prova);
```

```
// Calcola Pivot  
R1=(2*Pivot)-previousL;  
Pivot=(previousH+previousC+previousL)/3;  
S1=(2*Pivot)-previousH;
```

```
R1mid=(R1+pivot)/2;  
S1mid=(S1+pivot)/2;
```

```
if cambiatogiorno then  
    // Termina le linee precedenti qui, Crea nuove Linee per il prossimo Giorno  
    lin1 = CREATEOGG;  
    lin2 = CREATEOGG;  
    lin3 = CREATEOGG;  
    lin4 = CREATEOGG;  
    lin5 = CREATEOGG;
```

```

indperiodo = 0;
ggini = GetDate; // da dove parto a tracciare la linea orizzontale

    primavolta = false;
endif;

indperiodo = indperiodo + 1;
numper = indperiodo;

// Disegna Linee orizzontali lunghe solo in quel giorno
lin1 = drawhlineper(lin1,0,ggini, r1, numper, fuchsia,1,0);
lin2 = drawhlineper(lin2,0,ggini, Pivot, numper, blue,2,0);
lin3 = drawhlineper(lin3,0,ggini, s1, numper, aqua,1,0);
lin4 = drawhlineper (lin4,0,ggini, R1mid, numper, fuchsia,1,2);
lin5 = drawhlineper (lin5,0,ggini, S1mid, numper, aqua,1,2);

// Disegna una linea Verticale in corrispondenza delle 15:30
if T=1530 then
    DrawVLine(0,0,GetDate,lime,1,2);
endif;

```

## InRange

**InRange ( Valore): boolean**

### Descrizione

Ritorna True se *Valore* è compreso all'interno della barra attuale del Titolo, cioè:  
 = True se *Valore*  $\geq$  *Low* and *Valore*  $\leq$  *High*.

## Inside

**Inside: boolean**

### Descrizione

Ritorna True se la barra attuale è contenuta nella barra precedente, cioè:  
 = True se *High*  $\leq$  *High[1]* e *Low*  $\geq$  *Low[1]*.

## Outside

**Outside: boolean;**

### Descrizione

Ritorna True se la barra attuale contiene la barra precedente, cioè:  
 = True se *High*  $\geq$  *High[1]* e *Low*  $\leq$  *Low[1]*.

## IsInside

**IsInside (numbarra): boolean**

### Descrizione

Ritorna True se la barra numbarraprec è contenuta nella barra precedente a questa, cioè:

= True se  $High[numbarra] \leq High[numbarra + 1]$  e  $Low \geq Low[numbarra + 1]$ .

### *Esempio 1*

if **IsInside(0) and IsInside(1)** then // sia la candela attuale, che quella precedente è inside.

// fai qualcosa

endif;

## IsOutside

**IsOutside (numbarra): boolean**

### Descrizione

Ritorna True se la barra numbarraprec è outside rispetto alla barra precedente a questa, cioè:

= True se  $High[numbarra] \geq High[numbarra + 1]$  e  $Low \leq Low[numbarra + 1]$ .

### *Esempio 1*

if **IsOutside(0) and IsOutside(1)** then // sia la candela attuale, che quella precedente è outside.

// fai qualcosa

endif;

## SetTickMin

**SetTickMin: boolean;**

### Descrizione

Imposta il minimo Tick del titolo che si sta analizzando.

Nota: Utilizzato solo con Titoli/Future che non hanno un definito, o non corretto, tick minimo.

## Funzioni Date/Time

### Bar

**Bar: boolean**

#### Descrizione

Ritorna la barra attuale (0).

Utilizzato nei parametri in cui occorre scegliere la barra.

#### *Esempio*

```
{ Compra in chiusura della barra }  
if C > C[1] then  
    EnterLong(Bar, AtClose);  
endif;
```

### NextBar

**NextBar: boolean**

#### Descrizione

Ritorna la barra successiva (1).

Utilizzato nei parametri in cui occorre scegliere la barra.

#### *Esempio*

```
{ Compra in apertura della barra successiva }  
if C > C[1] then  
    EnterLong(NextBar, AtOpen);  
endif;
```

### NextOpen

**NextOpen: boolean;**

#### Descrizione

Utilizzato nelle funzioni di Trading, quando si vuole impostare una condizione sull'apertura della barra successiva.

#### *Esempio 1*

```
{ Compra in Apertura della barra successiva SOLO SE l'apertura è INFERIORE  
o UGUALE (limit) a valscattato }  
if condizione_di_acquisto then  
    EnterLong(NextOpen, valscattato, limit);  
endif;
```

#### *Esempio 2*

```
{ Compra in Apertura della barra successiva SOLO SE l'apertura è  
SUPERIORE o UGUALE (stop) a valscattato }
```

```
if condizione_di_acquisto then
    EnterLong(NextOpen, valscattato, stop);
endif;
```

### *Esempio 3*

{ Compra in Apertura della barra successiva SOLO SE l'apertura è UGUALE (exact) a valscattato }

```
if condizione_di_acquisto then
    EnterLong(NextOpen, valscattato, exact);
endif;
```

## LastBar

**LastBar: boolean**

### Descrizione

Ritorna se la barra attuale è l'ultima barra del giorno, disponibile per poter fare trading.

N.B. Utilizzato con grafici Intraday.

### *Esempio*

{ Chiudi la posizione se siamo alla fine della giornata }

```
if LastBar then
    ExitLong(NextBar, AtClose);
endif;
```

## IsFirstBarDay

**IsFirstBarDay: boolean**

### Descrizione

Ritorna se la barra attuale è la prima barra del giorno.

*N.B. Utilizzato con grafici Intraday.*

### *Esempio*

{ Esempio di IsFirstBarDay }

```
if IsFirstBarDay then
    // fai qualcosa
endif;
```

## IIF

**iif: boolean**

### Descrizione

Per ritornare una condizione booleana, in un'unica riga. Equivale ad utilizzare le istruzioni **if-then-else-endif**;

*Esempio*

```
{ Esempio di iif }
```

```
Var: Miacondizione, Miorisultato;
```

```
// Sintassi 1
```

```
if miacondizione then
```

```
    Miorisultato = risultato_se_vero;
```

```
else
```

```
    Miorisultato = risultato_se_falso;
```

```
endif;
```

```
//Questo equivale a scrivere a:
```

```
// Sintassi 2
```

```
Miorisultato = iif(miacondizione, risultato_se_vero, risultato_se_falso);
```

*Esempio*

```
Var es1, es2;
```

```
es1 = iif(C > C[1], true, false);
```

```
es2 = iif(C > MOV(C, 10, S), C > C[1], C > C[2]);
```

## IsFirstBarBegin

**IsFirstBarBegin:** boolean

Descrizione

Ritorna se la barra attuale è la prima barra, in assoluto, del ciclo.

*Esempio*

```
{ Esempio di IsFirstBarBegin }
```

```
if IsFirstBarBegin then
```

```
    // fai qualcosa
```

```
endif;
```

## GetDate

**GetDate:** integer

Descrizione

Ritorna la data della barra attuale.

Per l'EndOfDay = YYYYMMGG

Per l'Intraday ritorna comunque un valore che identifica una **data+ora**.

*N.B.: Utilizzato sia con grafici EndOfDay che con Intraday.*

*N.B. Si utilizza per il disegno degli Oggetti (vedi: DrawLine, DrawHLine, DrawCircle...).*

*Esempio*

```
Var: miogiorno(0);  
{ Compra solo fino alle 17}  
If Condizione_Primo_Punto then  
miogiorno = GetDate;  
    // plotto una linea Verticale  
    DrawVLine (0, 0, miogiorno, C, red, 1, 0);  
Endif;
```

## GetDateDaily

**GetDateDaily: integer**

Descrizione

Ritorna la data della barra attuale.

Per l'EndOfDay = YYYYMMGG

*N.B.: Utilizzato sia con grafici EndOfDay che con Intraday.*

A differenza del GetDate per l'Intraday ritorna comunque un valore che identifica **solo la data**.

*Esempio*

```
Var: miogiorno(0);  
{ Compra solo fino alle 17}  
If Condizione_Primo_Punto then  
miogiorno = GetDateDaily;  
    // plotto una linea Verticale  
    DrawVLine (0, 0, miogiorno, C, red, 1, 0);  
Endif;
```

## GetDay

**GetDay: integer**

Descrizione

Ritorna il giorno della barra attuale

Utilizzato sia per grafici intraday che daily.

*Esempio:*

```
if (GetDay = 10) and (GetMonth = 05) then  
    // compra ogni 10 maggio
```

```
EnterLong(NextBar, AtOpen);  
endif;
```

## GetMonth

**GetMonth: integer**

### Descrizione

Ritorna il mese della barra attuale

Utilizzato sia per grafici intraday che daily.

*Esempio:*

```
if (GetMonth <> 08) then  
    // ..... altre istruzioni .....  
  
    // Compra tutti i mesi, tranne ad Agosto  
    EnterLong(NextBar, AtOpen);  
endif;
```

## GetYear

**GetYear: integer**

### Descrizione

Ritorna l'anno della barra attuale

Utilizzato sia per grafici intraday che daily.

*Esempio:*

```
if (GetYear = 2006) then  
    // ..... altre istruzioni .....  
  
    // Compra solo nell'anno 2006.  
    EnterLong(NextBar, AtOpen);  
endif;
```

## GetHour

**GetHour: integer**

### Descrizione

Ritorna l'ora della barra attuale

Utilizzato per grafici intraday.

N.B. Utilizzato con grafici Intraday.

*Esempio*

```
{ Compra solo fino alle 17 }  
if GetHour < 17 then  
    EnterLong(NextBar, AtOpen);  
endif;
```

## GetMinute

**GetMinute: integer**

### Descrizione

Ritorna i minuti della barra attuale. Utilizzato per grafici intraday.

N.B. Utilizzato con grafici Intraday.

## GetSecond

**GetSecond: integer**

### Descrizione

Ritorna i secondi della barra attuale. Utilizzato per grafici intraday.

N.B. Utilizzato con grafici Intraday.

## DayOfWeek

**DayOfWeek: integer**

### Descrizione

Ritorna il giorno della settimana della barra attuale.

N.B.: Per le costanti vedere il capitolo in fondo al manuale.

### *Esempio*

```
if (DayOfWeek = friday) then  
    ....  
endif;
```

## CompareTime

**CompareTime (ora, min, sec) : integer**

### Descrizione

Ritorna la differenza, espressa in secondi, tra il Tempo attuale ed il tempo passato con *ora, min, sec*.

N.B. Utilizzato con grafici Intraday.

### *Esempio*

```
{ Compra solo nella parte centrale del giorno, tra le 10 e le 17 }  
if (CompareTime(10, 0, 0) > 0) and (CompareTime(17, 0, 0) < 0) then  
    EnterLong(NextBar, AtOpen);  
endif;
```

## RangeDate

**RangeDate (miadatainizio, miadatafine) : boolean**

### Descrizione

Ritorna true se la barra attuale è compresa nell'intervallo tra le due date.

Il formato delle date è YYYYMMDD.

N.B. Utilizzato con grafici Daily.

*Esempio*

```
{ Compra solo tra il 01/05/2001 ed il 01/05/2002 }
```

```
{ Questo serve per scansionare la formula su periodi ben definiti }
```

```
if (RangeDate(20010501, 20020501)) and miacondizione) then
```

```
EnterLong(NextBar, AtOpen);
```

```
endif;
```

## REF

**REF (DataArray, Periodi)**

### Descrizione

Ritorna l'elemento precedente dell'Array.

Un valore di "Periodi" positivo si riferisce al valore passato.

*Esempio:*

**REF** (C, 5) -> ritorna il valore di chiusura di 5 barre fa. Questo equivale a C[5].

**REF** (MOV(C, 10, S), 1) -> ritorna il valore della media mobile, di 1 barra precedente. Nel caso daily, sia la barra di ieri.

**REF** (RSI(C, 14), 2) -> ritorna il valore dell'RSI, di 2 barre precedenti. Nel caso daily, sia la barra di 2 giorni fa.

## Time oppure T

**T: integer**

### Descrizione

Ritorna le ore ed i minuti della barra attuale, nel formato HHMM.

N.B. Utilizzato con grafici Intraday.

*Esempio*

```
{ Compra solo fino alle 17:10 }
```

```
if T < 1710 then
```

```
    EnterLong(NextBar, AtOpen);
```

```
endif;
```

## Funzioni Indicatori

### Accesso al Valore dell'Indicatore

Ci sono 2 forme utilizzabili per ogni funzione dell'Indicatore.

Nella prima forma gli indicatori ritornano il valore dell'indicatore della barra attualmente analizzata.

*Esempio*

```
Var: x(0); x = MOV(Close, 24, S);
```

### SECTION\_ENTERLONG:

```
if x > C then
```

```
    { la media e' al di sopra del valore di chiusura della candela analizzata }
```

```
endif;
```

### END\_SECTION

*x* Rappresenta il valore della Media Mobile Semplice (S) a 24 periodi (*intraday*: 24 barre; *end-of-day* = 24 giorni) della barra attualmente analizzata.

### Accesso al DataArray Completo

La seconda forma ritorna un puntatore all'Array Dati completo. Questo puntatore e' un valore intero che puoi passare ad ogni funzione che accetta come parametro "DataArray".

E' pertanto possibile calcolare indicatori che, come DataArray, contengono un altro indicatore.

*Esempio1*

```
Var: media(0), m2m (0);
```

```
media = MOV(Close, 24, S);
```

```
m2m = MOV(media, 14, E);
```

### SECTION\_ENTERLONG:

```
if (media > C) and (m2m > C) then
```

```
    { la media e' al di sopra del valore di chiusura della candela analizzata e la  
    media della media e' maggiore di 20 }
```

```
    EnterLong(NextBar, AtOpen); // Compra in apertura della barra successiva
```

```
endif;
```

### END\_SECTION

*Esempio2*

```
Var: x(0); x = MOV(RSI(x, 14, S), 24, S);
```

### **SECTION\_ENTERLONG:**

if (x > C) and (miorsi > 20) then

{ la media e' al di sopra del valore di chiusura della candela analizzata e l'rsi e' maggiore di 20}

**EnterLong**(NextBar, AtOpen); // Compra in apertura della barra successiva  
endif;

**END\_SECTION**

### **Parametri presenti negli Indicatori**

#### DataArray

Impostare l'array d'ingresso per il calcolo della media.

*C/L/H/O/A/array:*

C , Close -> Chiusura barra

L , Low -> Minimo barra

H , High -> Chiusura barra

O , Open -> Apertura barra

A , AvgPrice -> Prezzo medio barra.

*DataArray* -> Altro Indicatore dichiarato in precedenza.

#### Periods

Indica il numero di barre utilizzati per il calcolo.

In Intraday:

Ogni barra rappresenta N minuti, a seconda della Compressione Dati impostata.

In End-Of-Day (Daily):

Ogni barra/periodo rappresenta 1 giorno.

### **Lista degli Indicatori:**

Qui viene indicata la Lista degli Indicatori e Oscillatori che è possibile utilizzare all'interno della formula, seguito dalla sintassi con cui dichiararli.

### **AccumDist**

**AccumDist (DataArray, Type): float**

#### Descrizione

AccumDist calcola l'Accumulazione e Distribuzione dello specificato DataArray. Il tipo e' specificato dal parametro *Type*.

*Type:*

CUM = Cumulativo; STD = Standard.

*Esempio*

Var: mioosc (0); mioosc = **AccumDist** (C, CUM);

## ATR

**ATR(DataArray, Periods): float**

Descrizione

ATR calcola l'Average True Range del periodo passato (*Periods*).

*Esempio*

Var: mioatr(0); mioatr = **ATR**(C, 10);

## BBandCenter, BBandUpper, BBandLower

**BBandCenter (DataArray, Periods, DevStandard, PerShift): float**

Descrizione

BBandCenter calcola la Bollinger Band Centrale/Superiore/Inferiore dello specificato DataArray, con la deviazione standard e lo shift orizzontale (PerShift).

*Esempio*

Var: mioosc(0); mioosc = **BBandCenter** (C, 20, 4, 0);

## CandleCount

**CandleCount(DataArray, periodo): float**

Descrizione

CandleCount calcola la differenza tra il numero di candele bianche ed il numero di candele nere, per un certo periodo di tempo, rapportato in percentuale.

## CCI

**CCI (DataArray, Periods): float**

Descrizione

CCI calcola il Commodity Channel Index dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **CCI** (C, 14);

## ChaOsc

**ChaOsc (DataArray): float**

Descrizione

ChaOsc calcola il *Chaikin Oscillator* dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **ChaOsc** (C);

## ChTrendS

**ChTrendS (DataArray): float**

### Descrizione

ChTrendS calcola il Chande Trendscore dello specificato DataArray.

### *Esempio*

Var: mioosc (0); mioosc = **ChTrendS** (C);

## ConstVal

**ConstVal (valore): float**

### Descrizione

ConstVal crea un DataArray con un valore costante.

Viene utilizzato per fare calcoli tra i DataArray (cioè tra altri oscillatori).

### **Esempio**

Var: mioval5, miamedia;

miamedia = Mov(C, 10, S);

mioval5 = **ConstVal** (5);

if **CrossOver**(miamedia, mioval5) then

    // La media ha superato il valore 5 fai qualcosa

endif;

## DIFF

**DIFF(DataArray1, DataArray2, Type): float**

### Descrizione

DIFF calcola la Differenza tra due Array di Dati (Indicatori). Il tipo e' specificato dal parametro *Type*.

### *Type:*

Sub = Differenza in valore.

Perc = Differenza in percentuale.

Perc2 = Differenza in percentuale rispetto al secondo valore.

### *Esempio*

Var: miadiff (0), miamedia(0); miamedia = **MOV**(C, 10, S);

    miadiff = **DIFF** (C, miamedia, Sub);

    // Differenza tra il titolo e la sua media

## DMADX

**DMADX (DataArray, Periods): float**

### Descrizione

DMADX calcola il *Directional Movement ADX* dello specificato DataArray.

### *Esempio*

mioosc = **DMADX (C, 14)**;

## DMIndex

**DMIndex (DataArray, Periods): float**

### Descrizione

DMIndex calcola il *Directional Movement Index* dello specificato DataArray.

### *Esempio*

mioosc = **DMIndex (C, 14)**;

## DMNDX

**DMNDX (DataArray, Periods): float**

### Descrizione

DMNDX calcola il *Directional Movement -DX* dello specificato DataArray.

### *Esempio*

mioosc = **DMNDX (C, 14)**;

## DMPDX

**DMPDX (DataArray, Periods): float**

### Descrizione

DMPDX calcola il *Directional Movement +DX* dello specificato DataArray.

### *Esempio*

mioosc = **DMPDX (C, 14)**;

## EXPSMOOTH

**EXPSMOOTH (DataArray, Periods): float**

### Descrizione

EXPSMOOTH calcola l'Exponential Smoothing dello specificato DataArray.

### *Esempio*

Var: mioosc (0);mioosc = **ExpSmooth (C, 10)**;

## EXPSMOOTH di WILDER

**WILDER(DataArray, Periods): float**

### Descrizione

EXPSMOOTH di Wilder calcola l'Exponential Smoothing di Wilder dello specificato DataArray.

### *Esempio*

Var: mioWilder, Periodi;  
mioWilder = **WILDER(C, Periodi)**;

## ForceIndex

**ForceIndex (DataArray, Periods): float**

### Descrizione

ForceIndex calcola il Force Index dello specificato DataArray.

### *Esempio*

Var: mioosc (0);mioosc = **ForceIndex (C, 13)**;

## HHV

**HHV (DataArray, Periods): float**

### Descrizione

HHV calcola il valore massimo del periodo passato (*Periods*).

### *Esempio*

Var: miomax10g (0); miomax10g = **HHV (C, 10)**;  
if C = miomax10g then  
    ... // Ho trovato un massimo (a 10 giorni) del valore di Chiusura  
endif;

## HLEnvelope

**HLEnvelope (DataArray, Periods): float**

### Descrizione

HLEnvelope calcola l'High-Low Envelope dello specificato DataArray.

### *Esempio*

Var: mioosc (0);mioosc = **HLEnvelope (C, 12)**;

## IMI

**IMI (DataArray, Periods): float**

### Descrizione

IMI calcola l'Intraday Momentum Indicator dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **IMI (C, 9)**;

## Keltner

**Keltner (DataArray): float**

Descrizione

Keltner calcola il Keltner Channel principale dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **Keltner (C)**;

## KeltnerLower

**KeltnerLower (DataArray): float**

Descrizione

KeltnerLower calcola il Keltner Channel Inferiore dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **KeltnerLower (C)**;

## KeltnerUpper

**KeltnerUpper (DataArray): float**

Descrizione

KeltnerUpper calcola il Keltner Channel Superiore dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **KeltnerUpper (C)**;

## KST

**KST (DataArray, PerMM1, PerRoc1, PerMM2, PerRoc2, PerMM3, PerRoc3, PerMM4, PerRoc4, TypeMov, PerMMTrig): float**

Descrizione

KST calcola il KST utilizzando 3 Medie Mobili e 3 Rate of Change, con la media mobile di Trig di tipo *TypeMov* ed il periodo *PerMMTrig*.

*TypeMov:*

S = Media Mobile Semplice.

A = Media Mobile Aritmetica.

E = Media Mobile Esponenziale.

*Esempio*

Var: mioosc (0); mioosc = **KST (C, 10, 10, 10, 15, 10, 20, 15, 30, E, 8)**;

## LLV

**LLV (DataArray, Periods): float**

### Descrizione

LLV calcola il valore minimo del periodo passato (*Periods*).

### *Esempio*

```
Var: miomin10g (0);  
miomin10g = LLV (C, 10); // il minimo dei close a 10 periodi  
if C = miomin10 then  
    ... // Ho trovato un minimo (a 10 giorni) del valore di Chiusura  
endif;
```

### *Esempio2*

```
Var: miomin10g (0);  
miomin10g = LLV (L, 10); // il minimo dei Minimi a 10 periodi  
if L = miomin10 then  
    ... // Ho trovato un minimo (a 10 giorni) del minimo  
endif;
```

## MFI

**MFI (DataArray, Periods): float**

### Descrizione

MFI calcola il Money Flow Index dello specificato DataArray.

### *Esempio*

```
Var: mioosc (0); mioosc = MFI (C, 14);
```

## MOV

**MOV(DataArray, Periods, Type): float**

### Descrizione

MOV calcola la Media Mobile del periodo passato (*Periods*). Il tipo di media e' specificato dal parametro *Type*.

### *Type:*

S = Media Mobile Semplice.

E = Media Mobile Esponenziale.

A = Media Mobile Aritmetica.

### *Esempio*

```
// Calcola la media mobile (S) sul titolo  
media=Mov(C,30,S);
```

### **SECTION\_ENTERLONG:**

```
// Acquistiamo se il titolo sale sopra la media mobile
```

```
if C > media then
    EnterLong(NextBar, AtOpen); // COMPRA
endif;
END_SECTION
```

## MACD

**MACD(DataArray, periodiMM1, periodiMM2): float**

### Descrizione

MACD calcola il MACD utilizzando il parametro della 1° media a *periodiMM1*, la 2° media a *periodiMM2*, senza il signal.

### *Esempio*

Var: miomacd(0); miomacd = **MACD**(C, 5, 5);

## MACDSIGN

**MACDSIGN(DataArray, periodiMM1, periodiMM2, periodi\_segna): float**

### Descrizione

MACDSIGN calcola il Signal dell'MACD utilizzando il parametro della 1° media a *periodiMM1*, la 2° media a *periodiMM2* ed il periodo per il segnale.

### *Esempio*

Var: miomacd(0); miomacd = **MACDSIGN**(C, 5, 5, 3);

## Momentum

**Momentum(DataArray, Periods, Type): float**

### Descrizione

MOMENTUM calcola il Momentum del periodo passato (*Periods*). Il tipo e' specificato dal parametro *Type*.

### *Type:*

D = Differential; R = Ratio; P = Percent.

### *Esempio*

Var: miomom(0); miomom = **MOMENTUM** (C, 14, D);

## NVI

**NVI (DataArray): float**

### Descrizione

NVI calcola il Negative Volume Index dello specificato DataArray.

### *Esempio*

Var: mioosc (0); mioosc = **NVI** (C, 9);

## OP

**OP(DataArray1, DataArray2, Type): float**

### Descrizione

OP esegue **operazioni aritmetiche** sugli Array di Dati (Indicatori). Il tipo di operazione e' specificata dal parametro *Type*.

### Type:

Add = Addizione tra i 2 DataArray.  
Sub = Sottrazione tra i 2 DataArray.  
Divis = Divisione tra i 2 DataArray.  
Mul = Moltiplicazione tra i 2 DataArray.  
Pow = Elevamento a Potenza tra i 2 DataArray.

### *Esempio*

Variabili: operaz (0), miamedia(0);

```
miamedia = MOV(C, 5, E);  
operaz = OP (C, miamedia, Sub);
```

## OSC\_LN

**OSC\_LN (DataArray): float**

### Descrizione

OSC\_LN calcola il Logaritmo Naturale del DataArray passato.

### *Esempio*

Var: mioLn, indvp;

```
mioLn = OSC_LN (C);
```

```
indvp = createviewport(200, true, true);  
PlotChart(mioLn, indvp, black, solid,2); // disegna il Logaritmo Naturale
```

## OscV

**OscV (DataArray): float**

### Descrizione

OscV calcola il *Volume % +/- average* dello specificato DataArray.

### *Esempio*

Var: mioosc (0); mioosc = **OscV** (C);

## PistaCiclica

**PistaCiclica (DataArray, Periods): float**

### Descrizione

PistaCiclica calcola la Pista Ciclica dello specificato DataArray.

### *Esempio*

Var: mioosc (0);mioosc = **PistaCiclica (C, 25)**;

## PivotR

**PivotR (DataArray, Livello): float**

### Descrizione

PivotR calcola il Pivot Point (Superiore) dello specificato DataArray, del *Livello* specificato.

### *Livello:*

1 = Resistenza di Primo Livello.

2 = Resistenza di Secondo Livello.

3 = Resistenza di Terzo Livello.

### *Esempio*

Var: mioosc (0); mioosc = **PivotR (C, 1)**;

## PivotS

**PivotS (DataArray, Livello): float**

### Descrizione

PivotS calcola il Pivot Point (Inferiore) dello specificato DataArray, del *Livello* specificato.

### *Livello:*

1 = Supporto di Primo Livello.

2 = Supporto di Secondo Livello.

3 = Supporto di Terzo Livello.

### *Esempio*

Var: mioosc (0); mioosc = **PivotS (C, 1)**;

## PriceOsc

**PriceOsc (DataArray, PerMovBreve, PerMovLunga, TypeMov): float**

### Descrizione

PriceOsc calcola il Price Oscillator dello specificato DataArray.

*PerMovBreve* e *PerMovLunga* sono i periodi della media mobile breve e lunga.

*TypeMov* e' il tipo di Media Mobile:

### *TypeMov:*

S = Media Mobile Semplice.

E = Media Mobile Esponenziale.

A = Media Mobile Aritmetica.

*Esempio*

Var: mioosc (0); mioosc = **PriceOsc (C, 9)**;

## PVI

**PVI (DataArray): float**

Descrizione

PVI calcola il Positive Volume Index dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **PVI (C, 9)**;

## PVT

**PVT (DataArray, Periods): float**

Descrizione

PVT calcola il Price Volume Trend dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **PVT (C, 9)**;

## QStick

**QStick (DataArray, Periods, TypeMov): float**

Descrizione

QStick calcola il QStick dello specificato DataArray.

*TypeMov:*

S = Media Mobile Semplice.

E = Media Mobile Esponenziale.

A = Media Mobile Aritmetica.

*Esempio*

Var: mioosc (0); mioosc = **QStick (C, 14)**;

## RegrLin

**RegrLin(DataArray, Periodi): float**

Descrizione

RegrLin calcola la regressione Lineare del periodo passato (Periodi).

*Esempio*

Var: miaregr(0);

miaregr = **RegrLin (C, 14)**;

## ROC

**ROC(DataArray, Periods): float**

### Descrizione

ROC calcola il Rate of Change del periodo passato (*Periods*).

### *Esempio*

Var: valroc(0);valroc = **ROC(C, 10)**;

## RS

**RS (DataArray): float**

### Descrizione

RS calcola la Forza Relativa dello specificato DataArray.

### *Esempio*

Var: mioosc (0); mioosc = **RS (C)**;

## RSI

**RSI(DataArray, Periods, Type): float**

### Descrizione

RSI calcola l'Rsi del periodo passato (*Periods*). Il tipo e' specificato dal parametro *Type*.

### *Type:*

S = Simple.

B = Blended.

T = Straight.

### *Esempio*

Var: mioatr(0); mioatr = **RSI(C, 14, S)**;

## SAR

**SAR (DataArray, Acc, MaxAcc ): float**

### Descrizione

SAR calcola il *Parabolic Stop and Reverse Point* dello specificato DataArray.

Indicare il *Fattore di Accelerazione* (da 0.001 a 0.1) e il *Massimo valore del Fattore di Accelerazione* (da 0.1 a 1).

### *Esempio*

Var: mioosc (0); mioosc = **SAR (C, 0.02, 0.2)**;

## STDDEV

**StdDev(DataArray, Periodi): float**

### Descrizione

RegrLin calcola la Deviazione Standard del periodo passato (Periodi).

Esempio

Var: miadev(0);

miadev = **StdDev**(C, 14);

## STOCHD

**STOCHD(DataArray, Periods%K, Periods%KS, Periodo%DL): float**

Descrizione

STOCHD calcola lo Stocastico utilizzando il parametro Periods%DL.

*Esempio*

Var: miostochd(0); miostochd = **STOCHD** (C, 14, 10, 14);

## STOCHK

**STOCHK (DataArray, Periods%K, Periods%KS): float**

Descrizione

STOCHK calcola lo Stocastico %K utilizzando il parametro Periods%K, Periods%KS e %DL.

*Esempio*

Var: miostochD (0), miostochK (0), par1(14), par2(14), pardl(14);

// Aggiungere qui le variabili che vi servono

miostochD = **STOCHD** (C, par1, par2, pardl);

miostochK = **STOCHK** (C, par1, par2);

**SECTION\_ENTERLONG:**

if miostochK > miostochD and miostochK [1] <= miostochD [1] then

**EnterLong**(NextBar, AtOpen);

endif;

**END\_SECTION**

## SumIncDaily

**SumIncDaily (DataArray): float**

Descrizione

SumIncDaily calcola la Somma degli Incrementi Giornalieri dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **SumIncDaily** (C);

## SumVal

**SumVal (DataArray, Periods): float**

### Descrizione

SumVal calcola la Sommatoria Valori del periodo passato, da 1 a 100 (*Periods*).

### *Esempio*

Var: mioosc (0); mioosc = **SumVal** (C, 1);

## SwingChart

**SwingChart(DataArray): float**

### Descrizione

SwingChart calcola lo SwingChart progettato da W.D.Gann.

## TillsonT3

**TILLSONT3 (DataArray, periods, Amplificazione, ShiftVerticale): float**

### Descrizione

TILLSONT3 utilizza la formula di Tillson.

### *Esempio*

Var: OscTillson, indvp;

OscTillson = **TILLSONT3** (C, 5, 0.7, 0);

indvp = createviewport(200, true, true);

PlotChart(OscTillson, indvp, black, solid,2); // *disegna l'oscillatore*

## Tradingweek-M1

### Descrizione

**TRADINGWEEK\_M1(DataArray, ParMedia1, ParTipoMedia1, ParMediaMacd1, ParMediaMacd2, ParMedia2, ParTipoMedia2): float**

### Descrizione

Tradingweek-M1 utilizza la seguente formula:

MM1=mov (c,6,a);

MAC=macd (c,5,6);

OPM=op (MM1,MAC,add);

MM2=mov (OPM,5,e);

TRADINGWEEK-M1=op (MM2,MAC,add);

### *Esempio*

Var: MioTW;

MioTW = **TRADINGWEEK\_M1**(C, 6, 1, 5, 6, 5, 2);

## **True Strength Index**

**TSI(DataArray; PerSmoothing1, PerSmoothing2, PerMomentum): float**

*Esempio*

Var: **mioTSI**, PerSmoothing1; PerSmoothing2, PerMomentum;  
mioTSI = **TSI**(C, **PerSmoothing1**, **PerSmoothing2**, **PerMomentum**);

**UltimateOsc**

**UltimateOsc (DataArray, Periods1, Periods2, Periods3): float**

Descrizione

UltimateOsc calcola l'Ultimate Oscillator dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **UltimateOsc** (C, 7, 14, 28);

**Variaz**

**Variaz (DataArray, Tipologia, ValConfr): float**

Descrizione

Variaz calcola le Variazioni dello specificato DataArray. Il tipo e' specificato dal parametro *Type*.

*Tipologia:*

P = Percentuale.

V = Valore.

*ValConfr:*

PREV = Confronto con il Periodo precedente .

FIRST = Primo Periodo.

*Esempio*

Var: mioosc (0); mioosc = **Variaz** (C, P, PREV);

**Vel**

**Vel (DataArray, Periods, TypeMov, PerShift): float**

Descrizione

Vel calcola la Velocità del periodo passato (*Periods*), con la media mobile di tipo *TypeMov* e lo shift orizzontale (*PerShift*, da -50 a +50).

*TypeMov:*

S = Media Mobile Semplice.

E = Media Mobile Esponenziale.

A = Media Mobile Aritmetica.

*Esempio*

Var: mioosc (0);mioosc = **Vel** (C, 30, S, 10);

**Volatility**

**Volatility (DataArray): float**

Descrizione

Volatility calcola la Volatilità dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **Volatility (C,5)**;

## WilliamsR

**WilliamsR (DataArray, Periods): float**

Descrizione

WilliamsR calcola il William's %R dello specificato DataArray.

*Esempio*

Var: mioosc (0); mioosc = **WilliamsR (C, 14)**;

## ZigZag

**ZigZag (DataArray, ValoreAmpiezza, ValConfr): float**

Descrizione

ZigZag calcola lo Zig Zag dello specificato DataArray. Il tipo e' specificato dal parametro *Type*.

*ValoreAmpiezza*: Valore compreso tra 0.01 e 50000.

*ValConfr*: PERC = Confronto in Percentuale ; POINT= Confronto in Punti.

*Esempio*

Var: mioosc (0); mioosc = **ZigZag (C, 5, PERC)**;

## Funzioni Matematiche

### AddTick

**AddTick( Valore, N:float ): float**

#### Descrizione

Ritorna il Valore selezionato aggiunto di N Tick. Il Valore di Tick è il piu' piccolo movimento di prezzo incrementale che un contratto del titolo puo' fare.

### AddPerc

**AddPerc( Valore, Perc:float ): float**

#### Descrizione

Ritorna il Valore selezionato aggiunto della percentuale impostata (*Perc*).

### Abs

**Abs( Valore: float ): float**

#### Descrizione

Ritorna il valore assoluto del valore specificato.

### ArcTan

**ArcTan( Valore: float ): float**

#### Descrizione

Calcola l'arcotangente del numero specificato, in gradi.

### BarSince

**BarSince(condizione): integer**

#### Descrizione

Conta il numero di barre passate, da quando la condizione si "verificata e continua ad essere valida.

Nota Bene: Questa funzione deve essere scritta al di fuori delle sezioni (Enter ed Exit), in quanto deve essere sempre controllata.

*Esempi:*

```
if BarSince(C > C[1]) = 3 then
```

```
{Se sono in un trend in salita, dove per almeno 3 barre consecutive la chiusura è stata superiore di quella precedente}
```

```
...
```

```
endif;
```

```

if BarSince(macd(C, 5, 5) < 0) = 2 then
    { Se sono passate 2 barre dopo che il MACD è diventato negativo... }
    ...
endif;

```

### **BarSinceInverted(condizione): integer**

#### Descrizione

Conta il numero di barre passate fino ad ora, quando si verificò la condizione, ma adesso non lo è più.

Nota Bene: Questa funzione deve essere scritta al di fuori delle sezioni (Enter ed Exit), in quanto deve essere sempre controllata.

Esempi:

{ Se è da 3 barre fa che non sono più long, quindi ora sono flat ... }

```

if BarSince(PositionDir=1) = 3 then

```

```

    // faccio altri controlli ... oppure compro

```

```

endif;

```

{ Se il MACD aveva invertito il suo segnale al ribasso di 2 barre fa, ed adesso la chiusura è maggiore di quella di ieri ... allora compro. }

```

VAR: val_macd, val_signal;

```

```

val_macd = MACD(C, 12, 26);

```

```

val_signal = Mov(val_macd, 9, S); // Media mobile sul MACD ("Signal")

```

```

if (BarSince(val_macd < val_signal) > 2) and (C > C[1]) then

```

```

    EnterLong(nextbar, atopen); // Entro Long

```

```

endif;

```

## Cos

### **Cos( Valore: float ): float**

#### Descrizione

Ritorna il coseno dell'angolo specificato. L'angolo dovrebbe essere specificato in gradi.

## Cosh

### **Cosh( Valore: float ): float**

### Descrizione

Ritorna il coseno iperbolico dell'angolo specificato.

## CrossOver

**CrossOver(DataArray1, DataArray 2): boolean**

### Descrizione

Ritorna *True* nel momento in cui si verifica il Cross della prima curva *DataArray1* al di sopra della curva *DataArray2*.

## CrossUnder

**CrossUnder(DataArray1, DataArray 2): boolean**

### Descrizione

Ritorna *True* nel momento in cui si verifica il Cross della prima curva *DataArray1* al di sotto della curva *DataArray2*.

## CrossOverLine

**CrossOverLine( IdOggetto:integer, DataArray:Array): float;**

### Descrizione

Ritorna *True* nel momento in cui si verifica il Cross dell'oggetto (IdOggetto) al di sopra della curva *DataArray*

### *Esempio*

```
idogg = DrawLine(NEWOGG, 0, giornoini1, valIni, giornofin, valFine, blue, 2, 0, EXTENDED);
```

```
// Disegno una linea impostando (data inizio, valore inizio, data fine, valore fine) che viene proiettata all'infinito (EXTENDED).
```

```
if CrossOverLine(idogg1, C) then // Se l'oggetto attraversa sopra il valore di chiusura del Titolo
```

```
    colorbar(green); // colora di verde
```

```
else
```

```
    colorbar(red); // altrimenti colora di rosso
```

```
endif;
```

## CrossUnderLine

**CrossUnderLine( IdOggetto:integer, DataArray:Array): float;**

### Descrizione

Ritorna *True* nel momento in cui si verifica il Cross dell'oggetto (IdOggetto) al di sotto della curva *DataArray*

*Esempio*

```
idogg = DrawLine(NEWOGG, 0, giornoini1, vallni, giornofin, valFine, blue, 2, 0, EXTENDED);
```

// Disegno una linea impostando (data inizio, valore inizio, data fine, valore fine) che viene proiettata all'infinito (EXTENDED).

```
if CrossUnderLine(idogg1, C) then
```

```
// Se l'oggetto attraversa sotto il valore di chiusura del Titolo
```

```
    colorbar(yellow); // colora di giallo
```

```
else
```

```
    colorbar(fuchsia); // altrimenti colora di fucsia
```

```
endif;
```

## Dec

**Dec( Valore: integer )**

Descrizione

Decrementa la variabile passata di 1.

## Exp

**Exp( Valore: float ): float**

Descrizione

Ritorna il valore di "e" elevato al valore specificato *Valore*, dove "e" è la base del logaritmo naturale.

## HighestSince

**HighestSince (N, Espressione, DataArray)**

Descrizione

Ritorna il valore massimo del DataArray dall' N-esima occorrenza quando l' "Espressione" risulta vera.

*Esempio:*

```
HighestSince (2, crossover(C, MOV(C, 10, S)), C)
```

In questo esempio, ritorna il valore più alto di chiusura dell'occorrenza più recente quando la Chiusura attraversa sopra la media a 10 giorni.

## Inc

**Inc( Valore: integer )**

Descrizione

Incrementa la variabile passata di 1.

## Int

**Int( Valore: float ): float**

### Descrizione

Ritorna la parte intera di un valore con la virgola.

## LN

**LN( Valore: float ): float**

### Descrizione

Ritorna il Logaritmo naturale del valore specificato.

## LowestSince

**LowestSince (N, Espressione, DataArray)**

### *Descrizione*

Ritorna il valore minimo del DataArray dall' N-esima occorrenza quando l' "Espressione" risulta vera.

### Esempio:

**LowestSince** (2, crossunder(C, MOV(C, 10, S)), C)

In questo esempio, ritorna il valore più basso di chiusura dell'occorrenza più recente quando la Chiusura attraversa sotto la media a 10 giorni.

## Max

**Max( n1, n2: float ): float**

### Descrizione

Ritorna il più grande tra i due valori specificati.

## Min

**Min( n1, n2: float ): float**

### Descrizione

Ritorna il più piccolo tra i due valori specificati.

## Pi

**Pi: float**

### Descrizione

Ritorna Pi-greco, circa 3.1415927.

## Rnd

**Rnd( Valore: float ): float**

### Descrizione

Ritorna un numero casuale compreso tra 0 ed il *Valore*.

## Random

**Random: float**

### Descrizione

Ritorna un numero casuale compreso tra 0 e 1.

## Randomize

**Randomize**

### Descrizione

Inizializza il generatore di numeri random con un valore random. Chiamare questa funzione all'inizio della formula per assicurarsi che si ottengano dei valori random differenti ogni volta che lo script viene eseguito.

## Round

**Round( Valore: float ): integer**

### Descrizione

Arrotonda il valore a virgola mobile con il valore intero più vicino.

## Sin

**Sin( Valore: float ): float**

### Descrizione

Ritorna il seno dell'angolo specificato. L'angolo dovrebbe essere specificato in gradi.

## Sinh

**Sinh( Valore: float ): float**

### Descrizione

Ritorna il seno iperbolico dell'angolo specificato.

## Sqrt

**Sqrt( Valore: float ): float**

### Descrizione

Ritorna la radice quadrata del valore specificato.

## Sum

**Sum(DataArray, numperiodi:integer): float**

### Descrizione

Ritorna la Sommatoria di N valori dell'Array impostato in ingresso.

*Esempio:*

Var: media1(0), somma;

```
media1 = Mov(C, 10, S); // Calcola Media del titolo a 10 barre
```

```
somma = Sum(media1, 3);
```

```
// ricevi la sommatoria delle ultime 3 barre della media.
```

```
// è come "somma = (media[0] + media[1] + media[2])"
```

## Tan

**Tan( Valore: float ): float**

### Descrizione

Ritorna la tangente dell'angolo specificato. L'angolo dovrebbe essere specificato in gradi.

## Tanh

**Tanh( Valore: float ): float**

### Descrizione

Ritorna la tangente iperbolica del Valore specificato.

## TerminateTs

### **TerminateTs**

#### Descrizione

Permette di terminare, prematuramente, il ciclo del Trading System.

Può servire nei casi in cui non si voglia continuare con l'esecuzione del sistema.

#### Esempio

```
{faccio qualcosa}
```

```
.....
```

```
if condizione then
```

```
    TerminateTs; // Esce dal ciclo e non esegue più nient'altro
```

```
endif;
```

## WithIn

### **WithIn(Valore1, valore2, valore3 ): float**

#### Descrizione

Ritorna *True* se valore1 è compreso tra i valori valore2 e valore3:

= True se *Valore1 > Valore2 and Valore1 < Valore3*.

## Funzioni di Output

### FileWriteString

#### **FileWriteString (nomefile:string, miastringa:string; acapo:boolean)**

#### Descrizione

Permette di scrivere una stringa all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

#### *Esempio*

```
if miacondizione then
```

```
    FileWriteString("c:\prova.txt", "condizione verificata", true);
```

```
endif;
```

### FileWriteVal

#### **FileWriteVal (nomefile:string; miovalore:single; acapo:boolean)**

#### Descrizione

Permette di scrivere un valore all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

*Esempio*

```
if miacondizione then
```

```
    FileWriteVal("c:\prova.txt", C, true);
```

```
// scrivo sul file il valore di chiusura di ogni candela (e vado a capo)
```

```
endif;
```

## FileWriteTicker

**FileWriteTicker(nomefile:string; acapo:boolean)**

Descrizione

Permette di scrivere il codice VT del Titolo all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

*Esempio*

```
if miacondizione then
```

```
    FileWriteTicker("c:\prova.txt", false);
```

```
endif;
```

## FileWriteDescriz

**FileWriteTicker(nomefile:string; acapo:boolean)**

Descrizione

Permette di scrivere la descrizione del Titolo all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

*Esempio*

```
if miacondizione then
```

```
    FileWriteDescriz("c:\prova.txt", false);
```

```
endif;
```

## FileWriteStringVal2

**FileWriteStringVal2 (nomefile:string; miastringa1:string; miovalore1:single; miastringa2:string; miovalore2:single; acapo:boolean)**

Descrizione

Permette di scrivere una coppia di stringa, valore, all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

*Esempio*

if miacondizione then

```
FileWriteStringVal2("c:\prova.txt", "compra se supera ", C, " oppure vendi se  
va sotto ", L, true);
```

endif;

## FileWriteDateHour

**FileWriteDateHour (nomefile:string; acapo:boolean)**

Descrizione

Permette di scrivere la data e ora attuale, all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

*Esempio*

if miacondizione then

```
FileWriteDateHour("c:\prova.txt", true);
```

```
// scrivo sul file la data e ora di ogni candela (e vado a capo)
```

endif;

## FileWriteDescr

**FileWriteDescr (nomefile:string; acapo:boolean)**

Descrizione

Permette di scrivere il nome del titolo (descrizione) all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

*Esempio*

if miacondizione then

```
FileWriteDescr("c:\prova.txt", true);
```

```
// Se sto analizzando Fiat mi ritorna "Fiat"
```

endif;

## FileWriteTicker

**FileWriteTicker (nomefile:string; acapo:boolean)**

Descrizione

Permette di scrivere il codice del titolo (Ticker) all'interno di un file.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

*Esempio*

if miacondizione then

```

FileWriteTicker("c:\prova.txt", true);
// Se sto analizzando Fiat mi ritorna "F" (codice interno di VT, ticker).
endif;

```

## PlaySound

### **PlaySound (nomefile:stringa)**

#### Descrizione

Permette di eseguire un file sonoro (wav), in Realtime, al verificarsi della condizione che viene verificata sull'ultima barra Completata.

Nel parametro "nomefile" va specificato anche il percorso, come mostrato in esempio.

#### *Esempio*

```

if mia_condizione then
  PlaySound ("c:\windows\media\chimes.wav");
endif;

```

## MsgLista

### **MsgLista(tipoOp:integer, ValoreOp:Float, Messaggio:string);**

#### Descrizione

Mostra l'ultimo messaggio, indicato in "Messaggio", sulla finestra "Lista Messaggi Trading System".

#### *Esempio*

```

if PositionDir <> 1 then // se non sono Long
  if condizione_di_acquisto then
    EnterLong(NextOpen, C, stop); // COMPRA Domani se supera la
    chiusura di oggi
    MsgLista(tsEnterlong, C, "Domani se supera :");
  endif;
endif;

```

#### Risultato in Visual Trader:



**Nota Bene:** Viene mostrato l'ultimo messaggio, per ogni TS applicato. Pertanto comparirà una riga per ogni grafico con applicato un TS.



## Funzioni Trading System

### EnterLong

**EnterLong(Barra, Valore, {LIMIT | STOP| EXACT}, NumBarreAttivo);**

#### Descrizione

Apri una Posizione Long nella barra impostata.

Vediamo i valori che possiamo assegnare ai parametri:

*Barra:* NextBar

Bar

*Valore:* AtClose

AtOpen

*Valore numerico* (es: O, C, Miovalore)

*3° parametro (facoltativo) :*

**LIMIT** : Esegue l'ordine quando il prezzo viene raggiunto o è inferiore (<= prezzo).

**STOP** : Esegue l'ordine quando il prezzo viene raggiunto o superato (>= prezzo)

**EXACT** : Esegue l'ordine quando il prezzo viene raggiunto (= prezzo)

Pertanto:

per impostare un Target, l'ordine sarà di tipo LIMIT.

per impostare uno Stop loss, l'ordine sarà di tipo STOP.

inseriamo EXACT se vogliamo specificare il quarto parametro(*Numbarreattivo*) senza LIMIT e STOP.

*Numbarreattivo (facoltativo)* : rappresenta il numero di barre successive in cui verrà tentato l'ingresso.

N.B.: Se la posizione attuale è Short e si vuole entrare Long, il TS chiude in Automatico la posizione Short e poi apre la Posizione Long.

*Esempio*

**SECTION\_ENTERLONG:**

If *condizione\_di\_entrata* then

**EnterLong**(NextBar, AtOpen);

Endif;

**END\_SECTION**

### ExitLong

**ExitLong(Barra, Valore, {LIMIT | STOP| EXACT}, NumBarreAttivo);**

#### Descrizione

Chiude una Posizione Long nella barra impostata.

Vediamo i valori che possiamo assegnare ai parametri:

*Barra:* NextBar

Bar

*Valore:* AtClose

AtOpen

*Valore numerico* (es: O, C, Miovalore)

*3° parametro (facoltativo) :*

**LIMIT** : Esegue l'ordine quando il prezzo viene raggiunto o è inferiore (<= prezzo).

**STOP** : Esegue l'ordine quando il prezzo viene raggiunto o superato (>= prezzo).

**EXACT** : Esegue l'ordine quando il prezzo viene raggiunto (= prezzo).

Pertanto:

per impostare un Target, l'ordine sarà di tipo **LIMIT**,  
per impostare uno Stop loss, l'ordine sarà di tipo **STOP**  
inseriamo **EXACT** se vogliamo specificare il quarto  
parametro(*Numbarreattivo*) senza LIMIT e STOP.

*Numbarreattivo (facoltativo) :* rappresenta il numero di barre successive in cui verrà tentato l'uscita.

*Esempio*

**SECTION\_EXITLONG:**

If *condizione\_di\_uscita* then

**ExitLong**(NextBar, AtOpen);

Endif;

**END\_SECTION**

## EnterShort

**EnterShort(Barra, Valore, {LIMIT | STOP| EXACT}, NumBarreAttivo);**

Descrizione

Apri una Posizione Short nella barra impostata.

Vediamo i valori che possiamo assegnare ai parametri:

*Barra:* NextBar

Bar

*Valore:* AtClose

AtOpen

*Valore numerico* (es: O, C, Miovalore)

*3° parametro (facoltativo) :*

**LIMIT** : Esegue l'ordine quando il prezzo viene raggiunto o superato (>= prezzo)

**STOP** : Esegue l'ordine quando il prezzo viene raggiunto o è inferiore (<= prezzo).

**EXACT** : Esegue l'ordine quando il prezzo viene raggiunto (= prezzo).

Pertanto:

per impostare un Target, l'ordine sarà di tipo **LIMIT**,

per impostare uno Stop loss, l'ordine sarà di tipo **STOP**

inseriamo **EXACT** se vogliamo specificare il quarto

parametro(*Numbarreattivo*) senza LIMIT e STOP.

*Numbarreattivo (facoltativo)* : rappresenta il numero di barre successive in cui verrà tentato l'ingresso.

N.B.: Se la posizione attuale è Long e si vuole entrare Short, il TS chiude in Automatico la posizione Long e poi apre la Posizione Short.

*Esempio*

**SECTION\_ENTERSHORT:**

If *condizione\_di\_entrata* then

**EnterShort**(NextBar, AtOpen);

Endif;

**END\_SECTION**

**ExitShort**

**ExitShort(Barra, Valore, {LIMIT | STOP| EXACT}, NumBarreAttivo);**

Descrizione

Chiude una Posizione Short nella barra impostata.

Vediamo i valori che possiamo assegnare ai parametri:

*Barra*: NextBar

Bar

*Valore*: AtClose

AtOpen

*Valore numerico* (es: O, C, Miovalore)

*3° parametro (facoltativo)* :

**LIMIT** : Esegue l'ordine quando il prezzo viene raggiunto o superato (>= prezzo)

**STOP** : Esegue l'ordine quando il prezzo viene raggiunto o è inferiore (<= prezzo).

**EXACT** : Esegue l'ordine quando il prezzo viene raggiunto (= prezzo).

Pertanto:

per impostare un Target, l'ordine sarà di tipo **LIMIT**.

per impostare uno Stop loss, l'ordine sarà di tipo **STOP**  
inseriamo **EXACT** se vogliamo specificare il quarto  
parametro(*Numbarreattivo*) senza LIMIT e STOP.

*Numbarreattivo (facoltativo)* : rappresenta il numero di barre successive in cui verrà tentato l'uscita.

*Esempio*

**SECTION\_EXITSHORT:**

If *condizione\_di\_uscita* then

**ExitShort** (NextBar, AtOpen);

Endif;

**END\_SECTION**

## InstallStopLoss

**InstallStopLoss (tipoval:integer, valore: single);**

Descrizione

Imposta uno StopLoss, limitando le perdite.

Ogni volta che viene effettuata un'operazione di entrata (Long o Short), viene attivato lo StopLoss impostato. Pertanto, quando la condizione di stop si verifica (*es: il titolo scende sotto del 3% del livello di acquisto*), viene chiusa la posizione attuale.

*tipoval:*

**INPERC** = in *valore*: viene impostata una percentuale.

**INVAL** = in *valore*: viene impostato un valore.

**INTICK** = in *valore*: viene impostato un numero di tick.

**INDIFF** = in *valore*: viene impostato una differenza (in valore).

Note:

1. Con l'utilizzo degli StopLoss è possibile uscire dalle posizioni, anche durante l'aggiornamento dei dati in Tempo Reale, cioè prima che la barra sia conclusa.
2. La stessa funzione vale anche per posizioni Long e per quelle Short.

*Esempio*

Variabili: media1, media2;

// Media mobili

media1=Mov(C,10,E); // Media piu' lenta

media2=Mov(C,5,E); // Media piu' veloce

// gli stoploss e take profit, se non devono essere modificati, devono essere aggiunti all'inizio di questo codice. In questo modo, ad ogni entrata di posizione, viene attivato lo stoploss (o take profit) attivato.

**InstallStopLoss**(INPERC, 3); // Chiudi posizione se sto perdendo il 3%.

if not **PositionLong** then // Se non sono Long

if **CrossOver**(media1, media2) then // quando la media1 attraversa Sopra la media2

**EnterLong**(NextBar, AtOpen); // Entra Long ed attiva lo StopLoss impostato prima.

endif;

endif;

Se, durante l'aggiornamento delle quotazioni, il titolo scende al di sotto del 3%, chiude la posizione.

Stessa cosa se fossimo entrati Short: al di sopra del 3% avrebbe chiuso la posizione.

## InstallTakeProfit

**InstallTakeProfit** (tipoval:integer, valore: single)

### Descrizione

Imposta un TakeProfit, prendendo profitto.

Ogni volta che viene effettuata un'operazione di entrata (Long o Short), viene attivato il TakeProfit impostato. Pertanto, quando la condizione di take si verifica (*es: il titolo sale sopra il 3% del livello di acquisto*), viene chiusa la posizione attuale.

*tipoval:*

**INPERC** = in *valore*: viene impostata una percentuale.

**INVAL** = in *valore*: viene impostato un valore.

**INTICK** = in *valore*: viene impostato un numero di tick.

**INDIFF** = in *valore*: viene impostato una differenza (in valore).

Note:

1. Con l'utilizzo dei TakeProfit è possibile uscire dalle posizioni, anche durante l'aggiornamento dei dati in Tempo Reale, cioè prima che la barra sia conclusa.
2. La stessa funzione vale anche per posizioni Long e per quelle Short.

*Esempio*

Variabili: media1, media2;

// Media mobili

```
media1=Mov(C,10,E); // Media piu' lenta
media2=Mov(C,5,E); // Media piu' veloce
```

// gli stoploss e take profit, se non devono essere modificati, devono essere aggiunti all'inizio di questo codice. In questo modo, ad ogni entrata di posizione, viene attivato lo stoploss (o take profit) attivato.

```
InstalltakeProfit(INPERC, 3); // Chiudi posizione se sto guadagnando il 3%.
```

```
if not PositionLong then // Se non sono Long
```

```
  if CrossOver(media1, media2) then // quando la media1 attraversa Sopra la media2
```

```
    EnterLong(NextBar, AtOpen); // Entra Long ed attiva lo StopLoss impostato prima.
```

```
  endif;
```

```
endif;
```

Se, durante l'aggiornamento delle quotazioni, il titolo sale sopra il 3%, chiude la posizione.

Stessa cosa se fossimo entrati Short: al di sotto del 3% avrebbe chiuso la posizione.

## InstallTrailingProfit

**InstallTrailingProfit (tipoval:integer, valore: single, valoreritracc: single);**

### Descrizione

Imposta uno stop ad inseguimento, prendendo profitto.

Ogni volta che viene effettuata un'operazione di entrata (Long o Short), viene attivato il TrailingProfit impostato.

### Esempio di Acquisto:

```
InstallTrailingProfit(INPERC, 3, 1);
```

**Fase 1** – Il titolo sale sopra il 3%: Viene attivato uno stoploss ad inseguimento del 1%.

**Fase 2** – Quando il titolo sale, viene aggiornato lo stop (1%) sul massimo raggiunto.

**Fase 3** – Quando il titolo scende al di sotto del 1%, rispetto al massimo, chiude la posizione.

*tipoval:*

**INPERC** = in *valore e valoreritracc* viene impostata una percentuale.

**INVAL** = in *valore e valoreritracc* viene impostato un valore.

**INTICK** = in *valore e valoreritracc* viene impostato un numero di tick.

**INDIFF** = in *valore e valoreritracc* viene impostato una differenza (in valore).

Note:

1. Con l'utilizzo dei TakeProfit è possibile uscire dalle posizioni, anche durante l'aggiornamento dei dati in Tempo Reale, cioè prima che la barra sia conclusa.
2. La stessa funzione vale anche per posizioni Long e per quelle Short.

*Esempio*

Variabili: media1, media2;

```
// Media mobili
```

```
media1=Mov(C,10,E); // Media piu' lenta
```

```
media2=Mov(C,5,E); // Media piu' veloce
```

```
// gli stoploss e take profit, se non devono essere modificati, devono essere  
aggiunti all'inizio di questo codice. In questo modo, ad ogni entrata di posizione,  
viene attivato lo stoploss (o take profit) attivato.
```

```
InstallTrailingProfit(INPERC, 3, 1); // Chiudi posizione se, dopo aver  
raggiunto il 3%, scende dell'1%.
```

```
if not PositionLong then // Se non sono Long
```

```
  if CrossOver(media1, media2) then // quando la media1 attraversa Sopra la  
  media2
```

```
    EnterLong(NextBar, AtOpen); // Entra Long ed attiva il TrailingProfit  
    impostato.
```

```
  endif;
```

```
endif;
```

## RemoveStopLoss

**RemoveStopLoss;**

Descrizione

Rimuove lo StopLoss impostato in precedenza.

Utilizzato per disattivare gli stop, quando non sono più necessari.

## RemoveTakeProfit

**RemoveTakeProfit;**

Descrizione

Rimuove il TakeProfit e il TrailingProfit impostati in precedenza.

Utilizzato per disattivare gli stop, quando non sono più necessari.

## PositionDir

**PositionDir: float**

### Descrizione

Ritorna la posizione attuale di trading.

Risultati:

= 1 : Posizione Long Aperta.

= -1 : Posizione Short Aperta.

= 0 : Posizione Flat, nessuna posizione aperta.

## PositionValue

**PositionValue: float**

### Descrizione

Ritorna il valore di entrata della posizione attuale di trading.

## PositionQta

**PositionQta: float**

### Descrizione

Ritorna la quantità attuale in carico.

se qta = 0 allora posizione FLAT

se qta > 0 allora posizione LONG

se qta < 0 allora posizione SHORT

## PositionLong

**PositionLong: boolean**

Ritorna true se la posizione attuale di trading e' Long.

## PositionShort

**PositionShort: boolean**

Ritorna true se la posizione attuale di trading e' Short.

## Funzioni Plotting delle Curve

### CreateViewport

**CreateViewport ( Dimzona, Mostragriglia, Crea ) : integer**

### Descrizione

Crea una Zona vuota in basso nel grafico, per poterci disegnare (plottare) gli oscillatori, indicatori o nuove curve.

*Dimzona* : è la dimensione (in millesimi) occupata dalla nuova zona del grafico, con 500 utilizzerà metà del grafico.

*Mostragriglia*: visualizza la griglia sul grafico.

Ritorna l'indice della nuova zona creata, (da utilizzare nella funzione PlotChart).

### *Esempio*

```
Var: Indzona1(0);
```

```
// Creo una zona, sotto al grafico, alta il 10% del grafico stesso.
```

```
Indzona1 = CreateViewport(100, 0, true);
```

```
.....
```

(vedere esempio sotto)

## PlotChart

**PlotChart (DataArray, IndZona, Colore, Stile, Spessore) : integer**

### Descrizione

Plotta sul grafico la Curva impostata (DataArray) nella Zona indicata con "IndZona".

*DataArray* : è l'array, l'oscillatore da plottare.

*IndZona*: è l'indice della zona su cui plottare. Con 0 viene plottato sopra il Titolo principale.

*Colore*: indica il colore utilizzato per disegnare la curva, si veda la sezione Costanti.

*Stile*: indica lo stile utilizzato per disegnare la curva, si veda la sezione Costanti.

*Spessore*: indica lo spessore utilizzato per disegnare la curva.

N.B.:Questa funzione PlotChart deve essere **SEMPRE** scritta in fondo alla formula, in quanto potrebbe utilizzare delle variabili che vengono assegnate nella formula.

### *Esempio1*

```
Var: Indzona1(0), media1(0), media2(0), mioosc(0);
```

```
media1 = Mov(C, 10, S); // Media
```

```
media2 = Mov(C, 20, S); // Media
```

```
mioosc = media2 – media1;
```

```
SECTION_ENTERLONG:
```

```
.....
```

END\_SECTION

SECTION\_EXITLONG:

.....

END\_SECTION

// Alla fine della Formula

// Creo una zona, sotto al grafico, alta il 10% del grafico stesso.

Indzona1 = CreateViewport(100, true, true);

// Plotto l'oscillatore "mioosc" nella Nuova zona "Indzona1", curva rossa, linea, spessore 2.

**PlotChart(mioosc, Indzona1, red, solid, 2);**

// Plotto l'oscillatore "media1" nella zona del Titolo principale, curva verde, linea, spessore 1.

**PlotChart(media1, 0, green, solid, 1);**

Note:

- In Realtime: Attivare la voce "Attiva in Realtime" (botone TS verde), per permettere il ricalcolo Automatico delle Curve.

- PlotChart puo' essere utilizzato anche per tracciare delle Linee Orizzontali.

*Esempio2*

.....

// Alla fine della Formula

// Tracciare una Linea orizzontale al livello 30.

**PlotChart(30, Indzona1, red, solid, 1);**

// Tracciare una Linea orizzontale al livello 0.

**PlotChart(0, Indzona1, red, solid, 1);**

## ColorBar

**ColorBar ( Colore )**

Descrizione

Colora la barra attuale sul grafico attivo.

*Colore* : colore d'applicare alla barra attuale. Per i codici dei colori vedere la Sezione "Costanti".

*Esempio*

If Condizione1 then

**ColorBar (green);**

Endif;

## ColorNextBar

### **ColorNextBar ( Colore )**

#### Descrizione

Colora la barra successiva a quella attuale sul grafico attivo.

Se il grafico è daily allora indica la barra del giorno successivo.

*Colore* : colore d'applicare alla barra successiva. Per i codici dei colori vedere la Sezione "*Costanti*".

#### *Esempio*

If Condizione1 then

**ColorNextBar (Moneygreen);**

Endif;

## ColorPrevBar

### **ColorPrevBar (Colore , numbarre)**

#### Descrizione

Colora la barra N-esima precedente a quella attuale sul grafico attivo.

Se il grafico è daily allora indica la barra del giorno N precedente.

*Colore* : colore d'applicare alla barra successiva. Per i codici dei colori vedere la Sezione "*Costanti*".

*numbarre* : numero di barre precedenti a quella attuale, in cui colorare la barra.

#### *Esempio*

If Condizione1 then

**ColorPrevBar (red, 2) // 2 barre prima**

Endif;

## DrawLine

### **DrawLine ( codogg , indzona, data1, valore1, data2, valore2, colore, spessore, [stile], [opzioni])**

#### Descrizione

Disegna una Linea sul Grafico (indzona).

*CodOgg*: è il codice dell'oggetto da utilizzare (serve per poter riaggiornare lo stesso oggetto su diversi punti della formula); con *NEWOGG* .

Con *NEWOGG*: viene creato tutte le volte un nuovo oggetto(consigliato).

Con *CREATEOGG*: Per ogni ciclo, crea tutte le volte un nuovo oggetto.

*IndZona*: è l'indice della zona su cui plottare. Con 0 viene plottato sopra il Titolo principale.

*Data1*: data corrispondente al Primo Punto della linea (la funzione *GetDate* restituisce la data corrente).

*Valore1*: valore corrispondente al Primo Punto della linea.  
*Data2*: data corrispondente all' ultimo Punto della linea.  
*Valore2*: valore corrispondente all'Ultimo Punto della linea.  
*Colore*: colore d'applicare alla linea. Si veda la sezione Costanti.  
*Spessore*: indica lo spessore utilizzato per disegnare la linea.  
*[Stile:]* indica lo stile utilizzato per disegnare la linea.  
*[Opzioni] (flag)*: Con EXTENDED: Prolunga la linea all'Infinito.

*Esempio*

```
Var: lin1, indzona, miadata1, data1, valore1, data2, valore2;
```

```
indzona = 0; // zona del grafico principale
```

```
if Condizione_Primo_Punto then  
    miadata1 = GetDate; //ottengo la data corrente  
    valore1 = C;  
endif;
```

```
If Condizione_Secondo_Punto then  
    // plotto la linea e la estendo all'infinito  
    DrawLine (0, indzona, miadata1, valore1, GetDate, C, green, 2, 0,  
EXTENDED);  
    data2 = GetDate;  
    valore2 = C;  
    lin1 = DrawLine(lin1, indzona, data1, valore1, data2, valore2, red, 2, 0);  
Endif;
```

```
If Condizione_Secondo_Punto_Aggiorno then  
    // In questo modo modifico sempre la linea (lin1).  
    lin1 = DrawLine(lin1, indzona, data1, valore1, GetDate, C, green, 2, 0);  
Endif;
```

Nota:

Al posto di “*Condizione\_Primo\_Punto, Condizione\_Secondo\_Punto e Condizione\_Secondo\_Punto\_Aggiorno*”, occorre impostare delle condizioni.

## DrawVLine

**DrawVLine ( codogg , indzona, data, colore, spessore, [stile], [opzioni])**

### Descrizione

Disegna una Linea Verticale sul Grafico (indzona).

*CodOgg*: è il codice dell'oggetto da utilizzare (serve per poter riaggiornare lo stesso oggetto su diversi punti della formula), con NEWOGG viene creato tutte le volte un nuovo oggetto.

*IndZona*: è l'indice della zona su cui plottare. Con 0 viene plottato sopra il Titolo principale.

*Data*: data in cui plottare la linea Verticale (la funzione `GetDate` restituisce la data corrente).

*Colore*: colore d'applicare alla linea. Si veda la sezione Costanti.

*Spessore*: indica lo spessore utilizzato per disegnare la linea.

*[Stile:]* indica lo stile utilizzato per disegnare la linea.

*[Opzioni] (flag)*: nessuno

### *Esempio*

```
Var: lin1(0), // codice dell'oggetto  
    indzona(0); // zona del grafico principale
```

```
If Condizione_Primo_Punto then  
    // plotto la linea Verticale  
    DrawVLine (NEWOGG, indzona, GetDate, red, 1, 0);  
Endif;
```

Nota:

Al posto di “*Condizione\_Primo\_Punto*” occorre impostare una o più Condizioni.

## DrawVLineIf

**DrawVLineIf (condizione, colore, spessore, [stile], [opzioni])**

### Descrizione

Disegna una nuova Linea Verticale sul Grafico Principale, se la condizione risulta vera.

*Condizione*: E' una qualsiasi condizione che, se verificata, crea questa linea verticale sul grafico.

*Colore*: colore d'applicare alla linea. Si veda la sezione Costanti.

*Spessore*: indica lo spessore utilizzato per disegnare la linea.

*[Stile:]* indica lo stile utilizzato per disegnare la linea.

*[Opzioni] (flag)*: nessuno

### *Esempio*

Var: **Miacondizione**; // zona del grafico principale

*MiaCondizione* = (*H* > *H*[1]) and (*C* > *MOV*(*C*, 10, *S*); // mettere la condizione

// plotto la linea Verticale se la condizione risulta verificata (true)

**DrawVLineIf** (*MiaCondizione*, green, 1, 0);

## DrawHLine

**DrawHLine ( codogg , indzona, valore, colore, spessore, [stile], [opzioni])**

### Descrizione

Disegna una Linea Orizzontale sul Grafico (indzona).

*CodOgg*: è il codice dell'oggetto da utilizzare (serve per poter riaggiornare lo stesso oggetto su diversi punti della formula), con NEWOGG viene creato tutte le volte un nuovo oggetto.

*IndZona*: è l'indice della zona su cui plottare. Con 0 viene plottato sopra il Titolo principale.

*Valore*: valore dove plottare la linea orizzontale.

*Colore*: colore d'applicare alla linea. Si veda la sezione Costanti.

*Spessore*: indica lo spessore utilizzato per disegnare la linea.

[*Stile*:] indica lo stile utilizzato per disegnare la linea.

[*Opzioni*] (*flag*): nessuno

### *Esempio*

Var: **lin1(0)**, // codice dell'oggetto

**indzona(0)**; // zona del grafico principale

If *Condizione\_Primo\_Punto* then

// plotto la linea Orizzontale

**DrawHLine** (NEWOGG, **indzona**, *C*, red, 1, 0);

Endif;

//Esempio2: plotto la linea Orizzontale a 25000

**DrawHLine** (NEWOGG, 0, 25000, red, 1, 0);

Nota:

Al posto di “*Condizione\_Primo\_Punto*” occorre impostare una o più Condizioni.

## DrawCircle

**DrawCircle ( codogg , indzona, data, valore, colore, colorefondo, spessore, [stile], [opzioni])**

### Descrizione

Disegna un Cerchio sulla barra determinata dalla data del Grafico.

*CodOgg*: è il codice dell'oggetto da utilizzare (serve per poter riaggiornare lo stesso oggetto su diversi punti della formula), con NEWOGG viene creato tutte le volte un nuovo oggetto.

*IndZona*: è l'indice della zona su cui plottare. Con 0 viene plottato sopra il Titolo principale.

*Data*: data in cui plottare il cerchio (la funzione [GetData](#) restituisce la data corrente).

*Valore*: valore dove plottare il cerchio: (UP: DOWN: MIDDLE:

*Colore*: colore d'applicare al cerchio. Si veda la sezione Costanti.

*Colorefondo*: colore di sfondo d'applicare al cerchio. (→ non attivo per ora).

*Spessore*: indica lo spessore utilizzato per disegnare il cerchio.

*[Stile:]* indica lo stile utilizzato per disegnare il cerchio.

*[Opzioni]*: nessuno

### *Esempi*

// plot di un cerchio su di un valore

**DrawCircle** (0, indzona, miovalore, middle, green, 0, 3);

// plot di un cerchio sul Massimo della candela del titolo

**DrawCircle** (0, indzona, GetData, up, green, 0, 3);

// plot di un cerchio sul Minimo della candela del titolo

**DrawCircle** (0, indzona, GetData, down, green, 0, 3);

// plot di un cerchio sull'intera candela del titolo

**DrawCircle** (0, indzona, GetData, middle, green, 0, 3);

## DrawHLinePer

**DrawHLinePer ( codogg , indzona, data1, valore1, data2, colore, spessore, [stile], [opzioni])**

### Descrizione

Disegna una Linea Orizzontale sul Grafico, a partire da una data, fino ad un certo valore.

*CodOgg*: è il codice dell'oggetto da utilizzare (serve per poter riaggiornare lo stesso oggetto su diversi punti della formula), con NEWOGG viene creato tutte le volte un nuovo oggetto.

*IndZona*: è l'indice della zona su cui plottare. Con 0 viene plottato sopra il Titolo principale.

*Data1*: data in cui iniziare a plottare la linea Orizzontale (la funzione [GetData](#) restituisce la data

*Valore1*: valore dove iniziare a plottare la linea orizzontale.

*Data2*: data in cui termina la linea. (con **LINEEND** : plotta fino alla fine del grafico).

*Colore*: colore d'applicare alla linea. Si veda la sezione Costanti.

*Spessore*: indica lo spessore utilizzato per disegnare la linea.

*[Stile:]* indica lo stile utilizzato per disegnare la linea.

*[Opzioni] (flag)*: nessuno

#### *Esempio1*

```
Var: lin1(0), // codice dell'oggetto
    indzona(0), data1(0); // zona del grafico principale
```

```
data1 = 20000105; // 5 gennaio 2000
```

```
If Condizione_A then
```

```
//Esempio1: plotto la linea Orizzontale a 25000, fino ad adesso (GetDate)
```

```
    DrawHLinePer (0, indzona, data1, 25000, GetDate, green, 1, 0);
```

```
Endif;
```

```
If Condizione_B then
```

```
//Esempio2:
```

```
// plotto la linea Orizzontale da GetDate, fino in fondo
```

```
DrawHLinePer (0, indzona, GetDate , C, LINEEND, red, 1, 0);
```

```
endif;
```

Nota:

Al posto di “*Condizione\_Primo\_Punto*” occorre impostare una o più Condizioni.

#### *Esempio2 (con Grafico Intraday)*

```
{*****}
** Disegno dei Pivot Point
** Esempio dell'utilizzo di EOD, GetValues e DrawHLinePer
*****}
```

```
Var: previousH,previousL,previousC,
    Pivot,R1,S1, R1mid,S1mid, prova, cambiatogiorno,
    lin1,lin2,lin3,lin4,lin5,
    indperiodo, numper, ggini, primavolta(-1);
```

```
previousH=EOD.H[1];
```

```

previousL=EOD.L[1];
previousC=EOD.C[1];

cambiatogiorno = GetValues(days, 1, prova, prova, prova, prova);

// Calcola Pivot
R1=(2*Pivot)-previousL;
Pivot=(previousH+previousC+previousL)/3;
S1=(2*Pivot)-previousH;

R1mid=(R1+pivot)/2;
S1mid=(S1+pivot)/2;

if cambiatogiorno then
    // Termina le linee precedenti qui, Crea nuove Linee per il prossimo Giorno
    lin1 = CREATEOGG;
    lin2 = CREATEOGG;
    lin3 = CREATEOGG;
    lin4 = CREATEOGG;
    lin5 = CREATEOGG;

    indperiodo = 0;
    ggini = GetDate; // da dove parto a tracciare la linea orizzontale

    primavolta = false;
endif;

indperiodo = indperiodo + 1;
numper = indperiodo;

// Disegna Linee orizzontali lunghe solo in quel giorno
lin1 = drawhlineper(lin1,0,ggini, r1, numper, fuchsia,1,0);
lin2 = drawhlineper(lin2,0,ggini, Pivot, numper, blue,2,0);
lin3 = drawhlineper(lin3,0,ggini, s1, numper, aqua,1,0);
lin4 = drawhlineper (lin4,0,ggini, R1mid, numper, fuchsia,1,2);
lin5 = drawhlineper (lin5,0,ggini, S1mid, numper, aqua,1,2);

// Disegna una linea Verticale in corrispondenza delle 15:30
if T=1530 then
    DrawVLine(0,0,GetDate,lime,1,2);

```

endif;

## Costanti

### Colori

	Aqua
	Black
	Blue
	Cream
	DkGray
	Fuchsia
	Gray
	Green
	Lime
	LtGray
	Maroon
	MedGray
	MoneyGreen
	Navy
	Olive
	Purple
	Red
	Silver
	SkyBlue
	Teal
	White
	Yellow

### Stile Curva

Solid: linea continua

Dash: trattini

Dot: punti

DashDot: trattino punto trattino punto

DashDotDot: trattino punto punto trattino punto punto

### Giorni della Settimana

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

### Compressione Dati

Daily

Week

Month

**Data Type (tipo dati)**

Intraday

EndOfDay

---

## Creazione Automatica di un TS

Visual Trader, permette di creare automaticamente formule di TS; utilizzando questo meccanismo non è necessario conoscere il linguaggio di programmazione e di conseguenza utilizzare l'Editor.

Cliccando con il tasto destro del mouse su un qualunque oscillatore già inserito sul grafico compare un sotto-menu in cui c'è anche la voce:

**Crea Trading System.**

Si apre una maschera in cui impostare le condizioni di acquisto e vendita, cliccando sul pulsante APPLICA, avremo il TS impostato sul grafico e anche visibile nella lista di tutte le formule nella cartella *Personali*.





