



# EASYSCRIPT

## Indice generale

Introduzione a EasyScript®.....	8
Premessa.....	8
Prerequisiti.....	8
Organizzazione della guida.....	8
Linguaggio di programmazione EasyScript®.....	9
Introduzione: concetti fondamentali.....	9
Logica Booleana.....	10
Struttura del programma.....	10
Funzioni.....	11
Parametri / Variabili.....	11
Programmazione Vettoriale.....	12
Gap di prezzi e volatilità.....	14
Analisi Tecnica.....	15
Variabili Primitive, Costanti & Parametri.....	18
Vettori di Prezzo.....	18
Vettori Temporalis.....	18
Costanti di Colore.....	19
Costanti Cromatiche.....	19
Costanti di Base.....	20
Costanti Operatori (da usare tramite le funzioni LOOP e BARLOOP).....	20
Costanti di Moving Averages.....	20
Costanti di Trend (da usare tramite la funzione Trend).....	21
Costanti Punti o Percentuali (da usare con indicatore Volume Oscillator)....	21
Costanti di Patterns Candlestick.....	21
Variabili per Function Script.....	24
Parametri per Signal o Indicator Script.....	24
Parametri di Money Management.....	25
Variabili definite dall'utente.....	27
Operatori Matematici e Logici.....	28
Moltiplicazione (*).....	28
Divisione (/ o \).....	28
Elevamento a potenza (^).....	28
Sottrazione (-).....	28
Somma (+).....	28
Uguale (=).....	28
Maggiore di (>).....	29
Minore di (<).....	29
Maggiore o uguale a (>=).....	29
Minore o uguale a (<=).....	30
Non uguale a (<> o !=).....	30
AND (&&).....	30
OR (  ).....	31
XOR ( ).....	31
NOT (!).....	32
EQV (&).....	32
Funzioni Primitive e Matematiche.....	33

Funzioni Primitive.....	34
AVG.....	34
AVGOF.....	34
BARLOOP.....	34
CHANGEIF.....	35
COUNTIF.....	35
CROSSOVER.....	36
CROSSUNDER.....	36
FOLLOWME.....	37
IF.....	37
LASTIF.....	38
LOOP.....	38
MAX.....	39
MAXOF.....	39
MIN.....	39
MINOF.....	39
NTHMAXOF.....	40
NTHMINOF.....	40
PRINT.....	40
REF.....	41
SUM.....	41
SUMIF.....	42
SUMOF.....	42
TREND.....	42
TRUEHIGH.....	43
TRUELOW.....	43
Funzioni Matematiche.....	44
ABS - POS.....	44
ARCCOS.....	44
ARCSIN.....	44
ARCTAN.....	45
CEIL.....	45
COS.....	45
DEGTORAD.....	45
EXP.....	46
FLOOR.....	46
FRACPORTION.....	46
INTPORTION.....	47
ISLASTBAR.....	47
LOG.....	47
LOG10.....	47
MOD.....	48
NEG.....	48
POW.....	48
RADTODEG.....	49
RND - RANDOM.....	49
ROUND.....	49
ROUNDTICK.....	50

SIGN.....	50
SIN.....	50
SQRT - SQUAREROOT.....	51
SQUARE.....	51
TAN.....	51
TODAYHIGH.....	52
TODAYLOW.....	52
TODAYOPEN.....	52
DIV.....	52
YESTERDAYCLOSE.....	53
YESTERDAYHIGH.....	53
YESTERDAYLOW.....	53
YESTERDAYOPEN.....	54
HeikinAshiCLOSE.....	54
HeikinAshiHIGH.....	54
HeikinAshiLOW.....	55
HeikinAshiOPEN.....	55
Funzioni di Analisi Tecnica ed Indicatori.....	56
Moving Averages.....	57
Simple Moving Average.....	57
Exponential Moving Average.....	57
Time Series Moving Average.....	58
Variable Moving Average.....	58
Triangular Moving Average.....	59
Weighted Moving Average.....	59
Welles Wilder Smoothing Moving Average.....	60
VIDYA Moving Average.....	61
Double Exponential Moving Average.....	61
Triple Exponential Moving Average.....	62
Linear Regression.....	63
Forecast.....	63
Intercept.....	63
R <sup>2</sup> (R-Squared).....	64
Slope.....	64
Bands Functions.....	65
Bollinger Bands.....	65
Fractal Chaos Bands.....	66
High Low Bands.....	66
Keltner Channel.....	67
Moving Average Envelope.....	68
Prime Number Bands.....	68
Starc.....	69
Oscillatori.....	70
Aroon.....	70
Aroon Oscillator.....	70
Average True Range.....	71
Chaikin Volatility.....	72
Chande Forecast Oscillator.....	72

Chande Momentum Oscillator.....	73
Detrended Price Oscillator.....	74
Directional Movement Index.....	74
Ease of Movement Oscillator.....	75
Elder Force Index.....	76
Fractal Chaos Oscillator.....	76
High Minus Low.....	77
Moving Average Convergence / Divergence (MACD).....	77
Momentum Oscillator.....	78
Price Oscillator.....	79
Prime Number Oscillator.....	79
Rainbow Oscillator.....	80
Sine Wave.....	81
Stochastic Oscillator.....	81
Trix Oscillator.....	82
True Range.....	83
Vertical Horizontal Filter.....	83
Volume Oscillator.....	84
Williams %R.....	85
Williams Accumulation Distribution.....	85
Funzioni Cromatiche.....	87
COLORIZE.....	87
Imposta il colore del primo parametro di uscita dell'indicatore con una scala di colore dal rosso al verde, a seconda del valore stesso dell'indicatore.....	87
RGB (solo funzione).....	88
Imposta il colore del primo parametro di uscita dell'indicatore con la tonalità del colore di sfondo dell'esempio di cui sopra (giallo chiaro).....	88
Altre Funzioni.....	89
Accumulative Swing Index.....	89
Center of Gravity.....	89
Chande Forecast Oscillator.....	90
Chaikin Money Flow.....	91
Commodity Channel Index.....	92
Comparative RSI.....	93
Coppock Curve.....	93
Correlation Analysis.....	94
Ehler Fisher Transform.....	94
Elder Force Index.....	95
Elder Ray Bull Power.....	96
Elder Ray Bear Power.....	96
Elder Thermometer Index.....	97
Gann Swing.....	98
Gaussian Filter.....	98
Gopalakrishnan Range Index.....	99
Highest High Value.....	99
Historical Volatility Index.....	100
Intraday Momentum Index.....	100
Klinger Volume Oscillator.....	101

Lowest Low Value.....	102
Market Facilitation Index.....	103
Mass Index.....	103
Median Price.....	104
Money Flow Index.....	105
Negative Volume Index.....	105
On Balance Volume.....	106
Parabolic Stop & Reversal (Parabolic SAR).....	107
Performance Index.....	107
Positive Volume Index.....	108
Pretty Good Oscillator.....	108
Price Rate of Change (ROC).....	109
Price CCI Divergence.....	110
Price Volume Trend.....	111
Qstick.....	111
Random Walk Index.....	112
Ravi.....	113
Relative Strenght Index (RSI).....	113
RGB (solo funzione).....	114
Schaff Trend Cycle.....	115
Standard Error.....	115
Standard Deviation.....	116
Stochastic Momentum Index.....	117
Simple Super Trend.....	118
Super Trend.....	118
Swing Index.....	119
Trade Volume Index.....	120
Typical Price.....	120
Twiggs Money Flow.....	121
Ultimate Oscillator.....	122
Variance.....	122
Volume Rate of Chance (Volume ROC).....	123
Weighted Close.....	123
Wilder Volatility.....	124
Zero Line (solo funzione).....	124
Zig Zag.....	125
Strategy Position.....	126
Bars Since Entry.....	126
Bars Since Exit.....	127
Total Net Profit.....	127
Open Position.....	127
Average Entry Price.....	128
Last Entry Price.....	128
Current Contracts.....	129
RunUp.....	130
DrawDown.....	130
GetMultiplier.....	131
ConsecutiveEntries.....	131

PlayOptions.it Functions.....	132
Trend Slope.....	132
Fiuto Canale.....	132
Fiuto Facile.....	133
Global Function.....	134
SetGlobalVar.....	135
GetGlobalVar.....	135
I Pattern di Candele Giapponesi.....	136
Identificazione dei Patterns.....	136
Costruire le proprie Funzioni ed Indicatori.....	138
Premessa.....	138
Creazione di proprie Function.....	138
Creazione di propri Indicator.....	139
Creazione di proprie Condition.....	140
Creazione di propri Expert Advisor.....	140
Creazione di propri Signal.....	141
Esempi di Trading System & Tecniche.....	142
Trading Systems.....	142
bee Moving Average Crossover.....	142
bee Bollinger Bands.....	144
Bobao Forecast System.....	145
bee Commodity Channel Index.....	147
bee Swing Trailing.....	148
Il sistemista Consiglia: Money Management.....	148

## Introduzione a EasyScript®

### **Premessa**

beeTrader® è stato concepito per inviare gli ordini a mercato nel momento in cui si verifica la condizione TRUE/FALSE e quindi nessun ordine verrà inviato in giacenza al Server del Broker con parametri Limit (così o meglio) o Stop (così o peggio). **Al momento dell'invio l'ordine è inviato Market se non specificato diversamente!**

Evitiamo in questo modo di avere degli ordini messi in anticipo ma che magari non vengono eseguiti perchè le quantità in attesa non vengono soddisfatte. Forse pagheremo un tick in più ma certamente non perderemo il TREND sul quale abbiamo scelto di entrare.

**Altra importante caratteristica è la possibilità di scegliere da beeTrader® se eseguire la Strategy in modalità On Close oppure Tick by Tick.** Di default beeTrader® esegue in modalità Tick by Tick, è possibile selezionare la modalità On Close dal menù Advanced Settings della Sidebar nella sezione Strategy.

Esempio:

#### **CLOSE > SimpleMovingAverage**

è una condizione che invierà l'ordine come se fosse scritta

#### **LAST > SimpleMovingAverage**

### **Prerequisiti**

Una conoscenza di base dell'analisi tecnica è l'unico prerequisito per l'utilizzo di questa guida di programmazione.

### **Organizzazione della guida**

La prima sezione di questa guida contiene brevi esempi che illustrano come eseguire comuni attività di base, come ad esempio l'individuazione di titoli all'interno di una fascia di prezzo specifica, che incrociano il valore di uno o più indicatori, che hanno una volatilità in aumento, e così via.

È possibile copiare e incollare molti di questi esempi nella sezione per la programmazione di beeTrader®.

L'ultima sezione di questa guida contiene un riferimento di funzioni, proprietà e costanti supportate dal linguaggio EasyScript® nonché esempi pratici del



sistema di trading.

Questo metodo di organizzazione permette al programmatore di ottenere subito dei risultati e progredire con l'apprendimento in funzione della presentazione di nuovi concetti.

### ***Linguaggio di programmazione EasyScript®***

EasyScript® è un linguaggio di programmazione vettoriale integrato in beeTrader® concepito appositamente per lo sviluppo di trading system tramite script. EasyScript® non riconosce le lettere maiuscole o minuscole (è case insensitive), perciò l'utente deve fare attenzione alla digitazione in quanto linguaggio riporta tutte le lettere come fossero maiuscole, si potrebbero quindi creare dei dopponi (MovingAverage = MOVINGAVERAGE).

Uno script è semplicemente una sequenza di istruzioni che istruiscono beeTrader® sulle operazioni da svolgere, come ad esempio fornire un avviso quando il prezzo di un titolo raggiunge un nuovo massimo, attraversa una Moving Average, o scende di una certa percentuale. Le possibilità sono pressoché infinite.

### ***Introduzione: concetti fondamentali***

EasyScript® è un linguaggio di programmazione potente e versatile per i traders. Il linguaggio fornisce tutto il necessario per costruire sofisticati trading system pezzo per pezzo senza un'approfondita formazione o esperienza di programmazione.

Lo script che segue è un esempio molto semplice per identificare che l'ultimo prezzo di negoziazione sia superiore al prezzo di apertura:

#### **LAST > OPEN**

E' quasi superfluo dire che lo scopo di questo script è quello di individuare quando l'ultimo prezzo di negoziazione è superiore al prezzo di apertura ... è quasi chiaro come l'inglese.

Proprio come una lingua parlata EasyScript® dà molti modi per esprimere ogni idea e offre una vasta gamma di modi per programmare un trading system.

Gli script possono essere molto semplici, come mostrato, o estremamente complessi, costituiti da centinaia di righe di istruzioni. Comunque, nella maggior parte dei trading system, gli script di solito consistono in poche righe di codice in quanto determinare tante condizioni significa anche restringere il numero delle volte che queste possano avvenire nel futuro.

## **Logica Booleana**

Gli script mostrati in questa prima sezione possono essere collegati tra loro utilizzando la logica booleana tramite l'utilizzo delle parole chiave AND o OR, per esempio:

Script 1 restituisce TRUE quando il LAST è maggiore del prezzo OPEN:

**LAST > OPEN**

Script 2 restituisce TRUE quando il VOLUME è due volte il VOLUME della barra del giorno precedente:

**VOLUME > REF (VOLUME, 1) \* 2**

È possibile aggregare gli script in modo che lo script restituisca risultati per i titoli che sono superiori all'apertura e con il VOLUME due volte il VOLUME della barra del giorno precedente:

**LAST > OPEN AND VOLUME > REF (VOLUME, 1) \* 2**

Allo stesso modo, è possibile modificare la AND in un OR per trovare titoli con prezzo superiore all'apertura oppure con un VOLUME due volte il VOLUME della barra del giorno precedente:

**LAST > OPEN OR VOLUME > REF (VOLUME, 1) \* 2**

Ancora una volta, le istruzioni sono semplici quasi come la lingua inglese. L'uso della logica booleana con le parole chiave AND e OR è un concetto molto importante che è ampiamente utilizzato dal linguaggio di programmazione EasyScript®.

## **Struttura del programma**

Non importa se il codice è tutto su una sola riga o su più righe. Spesso è più facile leggere uno script in cui il codice è suddiviso su più righe. Il seguente script funziona esattamente come l'esempio precedente, ma è un po' più facile da leggere:

**LAST > OPEN OR  
VOLUME > REF (VOLUME, 1) \* 2**

E' consigliabile costruire i vostri script in modo da renderli il più intuitivo possibile per riferimenti futuri. In alcuni casi può essere utile aggiungere commenti a uno script molto complesso. Un commento viene utilizzato per includere note esplicative in uno script.

Ogni volta che il cancelletto (#) è posizionato all'inizio di una riga, lo script ignora le parole che seguono. Le parole serviranno solo come un commento o una nota per rendere lo script più comprensibile:

```
# Valuta true quando il last
# è superiore al valore di apertura
# oppure il volume è doppio rispetto al precedente
LAST > OPEN OR
VOLUME > REF (VOLUME, 1) * 2
```

Lo script viene eseguito proprio come aveva fatto prima con l'unica differenza che è più facile comprendere il progetto e lo scopo dello script. Di default ogni script viene calcolato su massimo 200 barre, questa impostazione può essere modificata con la funzione SET REQUIRED\_BARS = numero di barre desiderate.

## **Funzioni**

Il linguaggio EasyScript® fornisce molte funzioni incorporate che rendono più facile la programmazione. Le funzioni che sono integrate nel nucleo di un linguaggio di programmazione vengono chiamate primitive. La funzione TREND ne è un esempio:

```
TREND (CLOSE, 30) = UP
```

In questo esempio, la funzione TREND dice a EasyScript® di identificare i TREND in cui il prezzo di chiusura è in un TREND rialzista da 30 periodi.

I valori che sono contenuti all'interno di una funzione (ad esempio la funzione REF o la funzione TREND) sono chiamati argomenti. Nell'esempio ci sono due argomenti della funzione TREND. Il primo argomento è il prezzo di chiusura, ed il secondo argomento è il numero 30 (periodi).

Se nello script non si scrivono tutti gli argomenti necessari per una funzione, EasyScript® segnalerà un errore, a meno che la funzione non preveda dei parametri di default, peculiarità che velocizza la creazione di script. Nel primo caso quindi verrà segnalato cosa c'è di sbagliato nello script, e quindi consentirà di risolvere il problema e riprovare.

In altre parole, EasyScript® non fornirà mai risultati errati a causa di un errore nella scrittura degli argomenti di ingresso di una funzione senza prima un avviso di un problema potenziale.

## **Parametri / Variabili**

EasyScript® integra al suo interno anche il concetto di variabile utente. L'utente può, utilizzando una variabile, trasmettere a beeTrader® che il valore di una particolare funzione non è fisso, ma è modificabile e va ricercato nella riga INPUTS della strategia. Riprendiamo l'esempio di prima:

```
TREND (CLOSE, 30) = UP
```

```
#utilizzando variabili utente
```

**INPUTS: @PRICE (CLOSE), @LENGHT (30)  
TREND (@PRICE, @LENGTH) = UP**

ecco quindi che invece di imputare il valore della lunghezza del periodo (30) si può utilizzare una variabile, cioè un termine a scelta dell'utente preceduto da @. Leggendo il carattere @, beeTrader® capisce che il valore va ricercato nella riga INPUTS delle variabili proprie della strategia.

*Nota: questa funzione è indispensabile per l'ottimizzazione del proprio trading system, in quanto l'ottimizzazione avverrà tra la scala di valori che l'utente imposta nella stringa INPUTS delle variabili proprie del Signal. E' comunque possibile impostare i valori da ottimizzare successivamente durante il processo di ottimizzazione.*

### **Programmazione Vettoriale**

I linguaggi di programmazione vettoriale (conosciuti anche come linguaggi multidimensionali) generalizzano le operazioni su operatori scalari per applicarli trasparentemente a vettori, matrici, ed array a più dimensioni.

L'idea fondamentale alla base della programmazione vettoriale è che le operazioni si applicano immediatamente a tutta una serie di valori (un vettore o campo). Questo permette di pensare e operare su interi insiemi di dati, senza dover ricorrere ad esplicite operazioni scalari singole.

A titolo di esempio, per calcolare la somma del prezzo medio di un titolo a 14 periodi, in un linguaggio di programmazione tradizionale come BASIC sarebbe necessario scrivere un programma simile a questo:

```
For bar = 14 to max  
Sum = 0  
For n = bar - 14 to bar  
Sum = Sum + (CLOSE + OPEN) / 2  
Next  
Next bar
```

Con EasyScript® si può ottenere lo stesso risultato scrivendo:

**SET Summation = Sum ((CLOSE + OPEN) / 2, 14)**

Ed ora Summation è un nuovo vettore che contiene la somma a 14 periodi del prezzo medio dello strumento in ogni punto.

Non è raro trovare esempi di programmazione vettoriale costituiti da una singola linea di codice che richiedono più di un paio di pagine di BASIC, Java o C++.

A questo punto ci si potrebbe chiedere che cosa sono "REF" e "TREND". Queste sono due delle funzioni primitive più utili costruite nel linguaggio EasyScript®. La funzione REF viene utilizzata ogni volta che si vuole fare riferimento a un valore in qualsiasi punto specifico in un vettore. Si supponga che il vettore MedianAverage contenga il prezzo medio di uno stock. Per accedere a un particolare elemento del vettore usando un linguaggio di programmazione tradizionale, si può scrivere:

**SET A = MedianAverage[n]**

Usando EasyScript® si può scrivere:

**SET A = REF(MedianAverage, n)**

La differenza principale è data da una variazione nella sintassi, ovvero i linguaggi tradizionali usano come riferimento i punti in un vettore a partire dall'inizio, oppure da 0 se i vettori sono a base zero. EasyScript® usa come riferimento i valori a ritroso, a partire dalla fine. Questo è più conveniente in quanto lo scopo di EasyScript® è, naturalmente, di sviluppare trading system. E' sempre l'ultimo, il valore più recente che è di maggior importanza.

Per ottenere il valore più recente del vettore MedianAverage potremmo scrivere:

**SET A = REF (MedianAverage, 0)**

Che è lo stesso che non utilizzare affatto la funzione REF. Quindi il modo migliore per ottenere l'ultimo valore (il valore più recente) in un vettore è quello di scrivere semplicemente:

**SET A = MedianAverage**

E' sempre preso in considerazione l'ultimo valore di un vettore se la funzione REF è assente.

Per ottenere il valore di una barra fa, scriveremo:

**SET A = REF (MedianAverage, 1)**

O di due barre fa:

**SET A = REF (MedianAverage, 2)**

Gli operatori di borsa fanno riferimento ad un "TREND" come uno stato in cui il prezzo di un titolo è in aumento (UP-TREND) o in diminuzione (DOWN-TREND) per diversi giorni, settimane, mesi o anni. L'investitore tipico o trader eviterebbe di aprire una nuova posizione lunga di un titolo che è in un TREND al ribasso per molti mesi.

EasyScript® offre una funzione primitiva chiamata TREND per rilevare appunto le tendenze di prezzo dello strumento, di volume, o di indicatori:

**TREND (CLOSE, 30) = UP**

Questo dice a EasyScript® di identificare i trades in cui il prezzo di chiusura è in un TREND rialzista da 30 giorni. Allo stesso modo, si potrebbe anche utilizzare la funzione TREND per individuare le tendenze dei volumi o di indicatori tecnici:

# il volume è stato  
# in un trend al ribasso da almeno 10 periodi  
**TREND (VOLUME, 10) = DOWN**

# l'indicatore CMO a 14 giorni  
# è in trend al rialzo da almeno 20 periodi  
**TREND (CMO (CLOSE, 14), 20) = UP**

La funzione TREND può essere utilizzata come conferma di un segnale del trading system. Supponiamo di avere un trading system che acquista quando il prezzo di chiusura passa sopra la Simple Moving Average a 20 periodi. Lo script potrebbe essere simile a questo:

# Restituisce un segnale di acquisto quando il prezzo di chiusura attraversa SMA a 20 periodi  
**CROSSOVER (CLOSE, SimpleMovingAverage (CLOSE, 20)) = TRUE**

Potrebbe essere utile ridurre i segnali generati dallo script ai soli casi in cui sia in atto un TREND rialzista da un certo numero di periodi. Aggiungendo la seguente riga di codice si ottiene il risultato voluto:

**AND TREND (CLOSE, 40) = UP**

TREND identifica se un vettore ha registrato un andamento verso l'alto, verso il basso o lateralmente, ma non ci dice l'ampiezza che quel TREND ha avuto. Possiamo utilizzare la funzione REF per determinare l'intervallo in cui i dati sono stati in TREND. Per trovare la variazione tra il prezzo più recente e il prezzo di 40 periodi fa, potremmo scrivere:

**SET A = LAST - REF (CLOSE, 40)**

### **Gap di prezzi e volatilità**

Anche se la funzione TREND può essere utilizzata per individuare i TREND e la funzione REF può essere utilizzata per determinare la durata che quel TREND ha avuto, è spesso molto utile identificare i gaps dei prezzi e le variazioni di volume estreme, che possono essere le prime indicazioni di un'inversione di TREND. Possiamo quindi scrivere:

# Restituisce vero se il prezzo è gap up  
**LOW > REF (HIGH, 1)**

# Restituisce vero se il prezzo è gap down  
**HIGH < REF (LOW, 1)**

È possibile inoltre specificare una percentuale minima per il gap di prezzo:

# Restituisce vero se il prezzo è distanziato almeno dell'1%  
**LOW > REF (HIGH, 1) \* 1,01**

E con una leggera variazione possiamo anche imputare che il volume abbia avuto uno scostamento con un ampio margine:

# Il volume è aumentato di 10 volte  
**VOLUME > REF (VOLUME, 1) \* 10**

O sulla media dei volumi:

# Il volume attuale è maggiore di 10 volte rispetto al volume medio  
**VOLUME > SimpleMovingAverage (VOLUME, 30) \* 10**

Possiamo anche misurare la volatilità di prezzo o volume utilizzando uno degli indicatori tecnici, come Volume Oscillator, Chaikin Volatility Index, Coefficient of Determination, Price Rate of Change, Historical Volatility Index, ecc. Questi indicatori tecnici sono descritti nel capitolo 3.

## ***Analisi Tecnica***

EasyScript® offre molte funzioni integrate di analisi tecnica. Utilizzando solo una singola riga di codice è possibile calcolare funzioni come Moving Average, Bollinger Bands, Candlestick, e così via.

Un elenco completo delle funzioni di analisi tecnica è previsto nel capitolo 3.

Il seguente è un semplice esempio di come utilizzare una delle funzioni più comuni di analisi tecnica, la Simple Moving Average:

**LAST > SimpleMovingAverage (CLOSE, 20)**

Questo script restituirà vero quando il LAST è sopra la Simple Moving Average a 20 periodi calcolata sul prezzo di chiusura.

La variabile CLOSE è in realtà un vettore di prezzi di chiusura, non solo il prezzo più recente. È anche possibile utilizzare i vettori OPEN, HIGH, LOW, CLOSE, VOLUME per creare i propri vettori utilizzando la parola chiave SET.

**SET Median = (CLOSE + OPEN) / 2**

Questo codice crea un vettore contenente il prezzo medio per ciascuna barra di negoziazione.

Una volta creato, possiamo usare il vettore Median all'interno dello stesso script:

### **LAST > SimpleMovingAverage (Median, 20)**

Questa funzione restituisce vero quando l'ultimo prezzo è maggiore della Simple Moving Average a 20 periodi sul prezzo medio di ciascuna barra di negoziazione.

Poiché le funzioni restituiscono vettori, le funzioni possono anche essere utilizzate come argomenti validi all'interno di altre funzioni:

### **LAST > SimpleMovingAverage (SimpleMovingAverage (CLOSE, 30), 20)**

Questa funzione restituisce vero quando l'ultimo prezzo è maggiore della Simple Moving Average a 20 periodi della Simple Moving Average a 30 periodi del prezzo CLOSE.

Molti indicatori tecnici, come il MACD ad esempio, hanno una "linea di segnale". Un segnale di acquisto o di vendita viene generato quando la linea di segnale attraversa verso l'alto o verso il basso l'indicatore.

La funzione CROSSOVER individua quando una serie ne ha attraversata un'altra.

Per esempio, possiamo trovare il punto esatto nel tempo in cui la Simple Moving Average ne ha attraversato un'altra utilizzando appunto la funzione CROSSOVER:

**SET MA1 = SimpleMovingAverage (CLOSE, 28)**

**SET MA2 = SimpleMovingAverage (CLOSE, 14)**

### **CROSSOVER (MA1, MA2)**

Lo script precedente restituisce vero quando il vettore MA1 ha attraversato dal basso verso l'alto il vettore MA2.

*Invertendo gli argomenti della funzione CROSSOVER, possiamo individuare quando il vettore MA2 ha attraversato dal basso verso l'alto il vettore MA1:*

### **CROSSOVER (MA2, MA1)**

Cerchiamo ora di creare uno script che trova una figura di "Key Reversal", in modo da poter vedere in prima persona come EasyScript® possa essere utilizzato per creare trading system basati su regole complesse.

La figura del "Key Reversal" è identificata da una barra che, dopo un TREND rialzista, presenta un'apertura maggiore della chiusura precedente, segna un nuovo massimo, ed ha un prezzo di chiusura inferiore al prezzo minimo della barra precedente.

Traduciamolo in forma di script con EasyScript®:



```
# Prima di tutto assicurarsi che lo stock sia in trend rialzista  
TREND (CLOSE, 30) = UP
```

```
# OPEN deve essere superiore al CLOSE della barra precedente  
AND OPEN > REF (CLOSE, 1)
```

```
# La barra attuale deve segnare un nuovo massimo  
AND HIGH >= HighestHighValue (30)
```

```
# E l'ultimo prezzo deve essere inferiore LOW della barra precedente  
AND LAST < REF (LOW, 1)
```

Questo script può anche essere invertito:

```
# Prima di tutto assicurarsi che lo stock sia in trend ribassista  
TREND (CLOSE, 30) = DOWN
```

```
# OPEN deve essere inferiore al CLOSE della barra precedente  
AND OPEN < REF (CLOSE, 1)
```

```
# La barra attuale deve segnare un nuovo minimo  
AND LOW <= LowestLowValue
```

```
# E l'ultimo prezzo deve essere superiore HIGH della barra precedente  
AND LAST > REF (HIGH, 1)
```

La figura di Key Reversal è molto rara, ma ha un'alta percentuale di affidabilità quando si verifica.

## Variabili Primitive, Costanti & Parametri

Questo capitolo è un riepilogo di riferimento per le variabili primitive, le costanti, i Candlestick Patterns, ed i parametri utilizzabili negli script.

### Vettori di Prezzo

I dati relativi ai prezzi come OPEN, HIGH, LOW, CLOSE, VOLUME, DATE e TIME possono essere utilizzati come vettori per qualsiasi funzione o espressione.

<b>OPEN</b>	Prezzo di apertura del periodo
<b>HIGH</b>	Prezzo massimo del periodo
<b>LOW</b>	Prezzo minimo del periodo
<b>CLOSE</b>	Prezzo di chiusura del periodo
<b>HAOPEN</b>	Heikin Ashi - Prezzo di apertura del periodo
<b>HAHIGH</b>	Heikin Ashi - Prezzo massimo del periodo
<b>HALOW</b>	Heikin Ashi - Prezzo minimo del periodo
<b>HACLOSE</b>	Heikin Ashi - Prezzo di chiusura del periodo
<b>LAST</b>	Ultimo prezzo battuto (uguale al CLOSE dopo la chiusura del mercato)
<b>VOLUME</b>	Volume degli scambi del periodo
<b>BARNUMBER</b>	Numero progressivo della barra

### Vettori Temporalì

I dati relativi ai fattori temporali, quali per esempio DATE, TIME, DAY, WEEK possono essere usati come vettori per qualsiasi funzione o espressione.

<b>DATE</b>	Data di chiusura del periodo, nel formato YYYYMMDD
<b>TIME</b>	Orario di chiusura del periodo, nel formato HHMM, per orari inferiori a 10:00, la forma numerica diventa HMM, per orari inferiori a 01:00, la forma numerica diventa MM, per orari inferiori a 00:10, la forma numerica diventa M.
<b>DAY</b>	Giorno in formato numerico (1, 2, 3, 4, ..., 31)
<b>MONTH</b>	Mese in formato numerico (1, 2, 3, 4, ..., 12)
<b>YEAR</b>	Anno in formato numerico YYYY (2014, 2015, 2016)
<b>DAYOFWEEK</b>	Giorno della settimana in formato numerico SUNDAY: 0

MONDAY:  
TUESDAY: 2  
WEDNESDAY: 3  
THURSDAY: 4  
FRIDAY: 5  
SATURDAY: 6

### **Costanti di Colore**

Le Costanti di Colore rappresentano i colori predefiniti che è possibile impostare durante la creazione di Indicator personalizzati. Sono presenti 22 colori predefiniti

<b>COLOR_BLACK</b>	Nero
<b>COLOR_WHITE</b>	Bianco
<b>COLOR_SILVER</b>	Grigio Chiaro
<b>COLOR_GRAY</b>	Grigio Scuro
<b>COLOR_LIGHT_RED</b>	Rosso Chiaro
<b>COLOR_RED</b>	Rosso
<b>COLOR_DARK_RED</b>	Rosso Scuro
<b>COLOR_LIGHT_GREEN</b>	Verde Chiaro
<b>COLOR_GREEN</b>	Verde
<b>COLOR_DARK_GREEN</b>	Verde Scuro
<b>COLOR_LIGHT_BLUE</b>	Blu Chiaro
<b>COLOR_BLUE</b>	Blu
<b>COLOR_DARK_BLUE</b>	Blu Scuro
<b>COLOR_LIGHT_YELLOW</b>	Giallo Chiaro
<b>COLOR_YELLOW</b>	Giallo
<b>COLOR_DARK_YELLOW</b>	Giallo Scuro
<b>COLOR_LIGHT_CYAN</b>	Azzurro Chiaro
<b>COLOR_CYAN</b>	Azzurro
<b>COLOR_DARK_CYAN</b>	Azzurro Scuro
<b>COLOR_LIGHT_MAGENTA</b>	Viola Chiaro
<b>COLOR_MAGENTA</b>	Viola
<b>COLOR_DARK_MAGENTA</b>	Viola Scuro

### **Costanti Cromatiche**

Le cromatiche possono essere utilizzate con la funzione COLORIZE, e servono per orientare la gradazione cromatica

<b>COLORIZE_ASCENDING</b>	Orientamento ascendente
<b>COLORIZE_DESCENDING</b>	Orientamento discendente

### **Costanti di Base**

Le Costanti di Base rappresentano valori numerici matematici utilizzati più di frequente.

<b>TRUE</b>	1
<b>FALSE</b>	0
<b>PI</b>	
<b>PI_M</b>	3.1415926535897932384626433832795
<b>PI_MATH</b>	
<b>NULL</b>	Vettore di dati vuoto
<b>NAN</b>	Not a Number
<b>TODAY</b>	Data attuale nel formato YYYYMMDD

### **Costanti Operatori (da usare tramite le funzioni LOOP e BARLOOP)**

Le Costanti Operatori vengono utilizzate per identificare le operazioni matematiche da eseguire tramite le funzioni LOOP e BARLOOP.

<b>ADD</b>	Addizione
<b>SUBTRACT</b>	Sottrazione
<b>MULTIPLY</b>	Moltiplicazione
<b>DIVIDE</b>	Divisione
<b>MAXIMUM</b>	Valore Massimo
<b>MINIMUM</b>	Valore Minimo

### **Costanti di Moving Averages**

Le costanti di Moving Averages possono essere utilizzate solo come parametro MATYPE di altri indicatori o come parametri della funzione MovingAverage, in quanto non sono funzioni, ma costanti.

<b>SIMPLE</b>	1 (identifica la Simple Moving Average)
<b>EXPONENTIAL</b>	2 (identifica la Exponential Moving Average)
<b>TIME_SERIES</b>	3 (identifica la Time Series Moving Average)
<b>VARIABLE</b>	4 (identifica la Variable Moving Average)

<b>TRIANGULAR</b>	5 (identifica la Triangular Moving Average)
<b>WEIGHTED</b>	6 (identifica la Weighted Moving Average)
<b>VOLATILITY</b>	7 (identifica la VIDYA Moving Average)
<b>WILDER</b>	8 (identifica la Welles Wilder Smoothing)
<b>DOUBLE</b>	9 (identifica la Double Exponential Moving Average)
<b>TRIPLE</b>	10 (identifica la Triple Exponential Moving Average)

### ***Costanti di Trend (da usare tramite la funzione Trend)***

<b>UP</b>	1 (identifica il trend in salita)
<b>DOWN</b>	2 (identifica il trend in discesa)
<b>SIDeways</b>	3 (identifica il trend laterale)

### ***Costanti Punti o Percentuali (da usare con indicatore Volume Oscillator)***

<b>POINTS</b>	1 (specifica che l'output è da misurarsi in punti)
<b>PERCENT</b>	2 (specifica che l'output è da misurarsi in percentuale)

### ***Costanti di Patterns Candlestick***

**LONG\_BODY**



**DOJI**



**HAMMER**



**HARAMI**



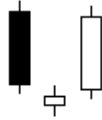
**STAR**



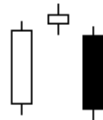
**DOJI\_STAR**



**MORNING\_STAR**



**EVENING\_STAR**



**PIERCING\_LINE**



**BULLISH\_ENGULFING\_LINE**



**BEARISH\_ENGULFING\_LINE**



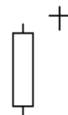
**DARK\_CLOUD\_COVER**



**HANGING\_MAN**



**BEARISH\_DOJI\_STAR**



**BEARISH\_SHOOTING\_STAR**



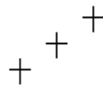
**SPINNING\_TOPS**



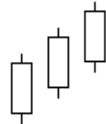
**HARAMI\_CROSS**



**BULLISH\_TRISTAR**



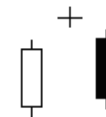
**THREE\_WHITE\_SOLDIERS**



**THREE\_BLACK\_CROWS**



**ABANDONED\_BABY**



**BULLISH\_UPSIDE\_GAP**



**BULLISH\_HAMMER**



**BULLISH\_KICKING**



**BEARISH\_KICKING**



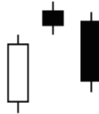
**BEARISH\_BELT\_HOLD**



**BULLISH\_BELT\_HOLD**



**BEARISH\_TWO\_CROWS**



**BULLISH\_MATCHING\_LOW**



## ***Variabili per Function Script***

### **RETURN**

La variabile RETURN può essere utilizzata per ritornare il risultato di una Function.  
In alternativa alla variabile RETURN è possibile utilizzare una variabile il cui nome sia identico al nome della Function.

Esempio:

**SET RETURN = SMA (CLOSE, 10) - SMA (CLOSE, 21)**

La Function restituisce la differenza tra la SMA a 10 periodi e quella a 21 periodi.

## ***Parametri per Signal o Indicator Script***

### **REQUIRED\_BARS**

La parola chiave REQUIRED\_BARS è utilizzata per definire il numero di barre di studio utilizzate per l'inizializzazione dei calcoli. Di default è impostato a 50 per Signals e 200 per Indicators.

Esempio:

**SET REQUIRED\_BARS = 250**

Definisce che il numero di barre utilizzate per l'inizializzazione dei



calcoli sia 250.

## Parametri di Money Management

Le funzioni di Money Management devono essere inserite all'inizio dei Signal, prima delle condizioni:

# codice corretto

**SET STOP\_LOSS = 100**

**CLOSE < MIN (CLOSE, 30)**

# codice errato

**CLOSE < MIN (CLOSE, 30)**

**SET STOP\_LOSS = 100**

<b>MAX_POSITION_OPEN</b>	<p>Quando questo valore è impostato tramite la parola chiave SET, tutte le posizioni saranno chiuse dopo il numero specificato di periodi, indipendentemente se le posizioni sono di acquisto o vendita, o se uno script di uscita è stato valutato TRUE.</p> <p>Esempio: <b>SET MAX_POSITION_OPEN = 20</b> Al raggiungimento della barra n°20 dall'apertura della posizione, il sistema chiude la posizione stessa indipendentemente se in profitto o perdita.</p>
<b>STOP_LOSS STOP_LOSS_PERCENT</b>	<p>Quando questo valore è impostato tramite la parola chiave SET, le posizioni saranno chiuse al raggiungimento della massima perdita specificata da STOP_LOSS.</p> <p>Esempio: <b>SET STOP_LOSS = 100</b> Al raggiungimento di una perdita di 100, il sistema chiude la posizione.</p>
<b>TAKE_PROFIT TAKE_PROFIT_PERCENT</b>	<p>Quando questo valore è impostato tramite la parola chiave SET, le posizioni saranno chiuse al raggiungimento del profitto specificato da TAKE_PROFIT.</p> <p>Esempio: <b>SET TAKE_PROFIT = 100</b> Al raggiungimento di un guadagno di 100, il sistema chiude la posizione.</p>
<b>TRAILING_STOP &amp; TRAILING_PERCENT</b>  <b>NB: SI USANO INSIEME</b>	<p>Quando questi valori sono impostati tramite la parola chiave SET, al raggiungimento del profitto impostato in TRAILING_STOP il sistema mette a rischio la % del profitto specificata in TRAILING_PERCENT.</p> <p>Esempio: <b>SET TRAILING_STOP = 400</b></p>

### **SET TRAILING\_PERCENT = 50**

Al raggiungimento di un guadagno di 400, il sistema calcola il prezzo di uscita per avere un profitto del 50%, ovvero 200. Se il profitto scende, appena raggiunge il valore di 200 il sistema chiude la posizione. Se il profitto sale, al raggiungimento di 800, il sistema ricalcola il prezzo di uscita in modo da ottenere un profitto del 50%, ovvero 400.

Quando questo valore è impostato tramite la parola chiave SET, i Signal utilizzano il numero di contratti indicato sullo script e non quello sulla Sidebar. Questo parametro può essere utile utilizzando la modalità pyramiding.

Esempio:

Buy Script

# Utilizza 5 contratti per le operazioni di Buy

**SET CONTRACTS = 5**

ExitLong Script

# Utilizza i contratti indicati nella sidebar per le operazioni

**SET CONTRACTS = 0**

# oppure nessuna indicazione specifica

In questo esempio supponendo che nella sidebar sia impostato 1 contratto, le operazioni di Buy avvengono con 5 contratti, mentre le operazioni di ExitLong con 1 contratto alla volta.

## **CONTRACTS**

## Variabili definite dall'utente

#assegnazioni variabili utente

**INPUTS: @USERVAR1 (VALUE1), @USERVAR2 (VALUE2)**

#assegnazioni variabili utente e range di parametrizzazione

**INPUTS: @USERVAR1 (VALUE1 [,FROM [,TO [,STEP]]]), @USERVAR2 (VALUE2)**

La parola chiave INPUTS è utilizzata per assegnare i valori alle variabili utente utilizzate come parametri dalle funzioni di EasyScript®, ma anche per assegnare il range e l'intervallo che viene utilizzato per l'ottimizzazione dei Signal. L'ottimizzazione può avvenire SOLO per valori numerici.

Attenzione: se la lista delle variabili è lunga e si desidera andare a capo, la nuova riga deve iniziare nuovamente con INPUTS:

Esempio:

#funzione TREND con parametri fissi

**TREND (CLOSE, 30) = UP**

#funzione TREND con parametri variabili

**INPUTS: @PRICE (CLOSE), @LENGTH (30)**

**TREND (@PRICE, @LENGTH) = UP**

Ogni qualvolta viene utilizzata la variabile utente @PRICE il sistema utilizza CLOSE, e ogni qualvolta viene utilizzata la variabile utente @LENGTH il sistema utilizza 30 periodi.

*Nota: questa funzione è ideata per l'ottimizzazione del proprio trading system, in quanto l'ottimizzazione avverrà tra la scala di valori che l'utente imposta nella stringa INPUTS delle variabili proprie del Signal. E' comunque possibile impostare i valori da ottimizzare successivamente durante il processo di ottimizzazione.*

#parametri variabili e @length impostata a 30 ed ottimizzabile da 3 a 60 con step di 1

**INPUTS: @PRICE (CLOSE), @LENGTH (30, 3, 60, 1)**

**TREND (@PRICE, @LENGTH) = UP**

## Operatori Matematici e Logici

In questo capitolo vengono presentati gli operatori matematici e logici necessari per eseguire operazioni nel linguaggio EasyScript®.

### **Moltiplicazione (\*)**

L'operatore di moltiplicazione viene utilizzato per moltiplicare tra loro due vettori o valori numerici.

### **Divisione (/ o \)**

L'operatore di divisione viene utilizzato per dividere tra loro due vettori o valori numerici.

### **Elevamento a potenza (^)**

L'operatore di elevamento a potenza viene utilizzato per elevare a potenza un vettore o valore numerico.

### **Sottrazione (-)**

L'operatore di sottrazione viene utilizzato per sottrarre tra loro due vettori o valori numerici.

### **Somma (+)**

L'operatore di somma viene utilizzato per sommare tra loro due vettori o valori numerici.

### **Uguale (=)**

L'operatore di uguaglianza viene utilizzato per assegnare un valore a una variabile o un vettore, o per confrontare i valori.

Quando viene utilizzato per l'assegnazione, il valore determinato da espressioni sul lato destro viene assegnato ad una variabile o un vettore sul lato sinistro. Inoltre, la parola chiave SET deve precedere il nome della variabile quando

l'operatore = viene utilizzato per un'assegnazione:

```
SET A = 123  
SET B = 123  
A = B = TRUE
```

### ***Maggiore di (>)***

L'operatore > determina se la prima espressione è maggiore della seconda espressione.

Esempio:

```
SET A = 124  
SET B = 123  
A > B = TRUE
```

### ***Minore di (<)***

L'operatore < determina se la prima espressione è minore della seconda espressione.

Esempio:

```
SET A = 123  
SET B = 124  
A < B = TRUE
```

### ***Maggiore o uguale a (>=)***

L'operatore >= determina se la prima espressione è maggiore o uguale alla seconda espressione.

Esempio:

```
SET A = 123  
SET B = 123  
A >= B = TRUE
```

E:

```
SET A = 124  
SET B = 123  
A >= B = TRUE
```

### **Minore o uguale a (<=)**

L'operatore <= determina se la prima espressione è minore o uguale alla seconda espressione.

Esempio:

```
SET A = 123  
SET B = 123  
A <= B = TRUE
```

E:

```
SET A = 123  
SET B = 124  
A <= B = TRUE
```

### **Non uguale a (<> o !=)**

Entrambi gli operatori != e <> determinano se la prima espressione non è uguale alla seconda espressione.

Esempio:

```
SET A = 123  
SET B = 124  
A != B = TRUE
```

E:

```
SET A = 123  
SET B = 124  
A <> B = TRUE
```

### **AND (&&)**

L'operatore AND viene utilizzato per eseguire una congiunzione logica tra due espressioni, dove le espressioni sono nulle, o sono del sottotipo booleano e hanno un valore True o False.

L'operatore AND può essere utilizzato anche un "operatore bit a bit" per rendere un confronto bit-by-bit di due numeri interi. Se entrambi i bit del confronto sono 1, viene restituito il valore 1. In caso contrario, viene restituito uno 0.

Quando si utilizza AND per confrontare espressioni booleane, l'ordine delle espressioni non è importante.

Esempio:

**(TRUE = TRUE AND FALSE = FALSE) = TRUE**

E:

**(TRUE = TRUE AND FALSE = TRUE) = FALSE**

## **OR (||)**

L'operatore OR viene utilizzato per eseguire una disgiunzione logica tra due espressioni, dove le espressioni sono nulle, o sono del sottotipo booleano e hanno un valore TRUE o FALSE.

L'operatore OR può essere utilizzato anche un "operatore bit a bit" per rendere un confronto bit-by-bit di due numeri interi. Se uno o entrambi i bit nel confronto sono 1, viene restituito il valore 1. In caso contrario, viene restituito uno 0.

Quando si utilizza OR per confrontare espressioni booleane, l'ordine delle espressioni non è importante.

Esempio:

**(TRUE = TRUE OR TRUE = FALSE) = TRUE**

E:

**(FALSE = TRUE OR TRUE = FALSE) = FALSE**

## **XOR (!)**

L'operatore XOR viene utilizzato per eseguire l'esclusione logica tra due espressioni, dove le espressioni sono nulle, o sono del sottotipo booleano e hanno un valore TRUE o FALSE.

L'operatore XOR può essere utilizzato anche un "operatore bit a bit" per rendere un confronto bit-by-bit di due numeri interi. Se entrambi i bit sono uguali nel confronto (entrambi sono 0 o 1), allora viene restituito uno 0. Altrimenti, viene restituito un 1.

Esempio:

**(TRUE XOR FALSE) = TRUE**

E:

**(FALSE XOR FALSE) = FALSE**

### **NOT (!)**

L'operatore NOT viene utilizzato per eseguire una negazione logica su un'espressione. L'espressione deve essere di sottotipo booleano e ha un valore TRUE o FALSE. Questo operatore causa ad un'espressione TRUE di diventare FALSE e ad un'espressione FALSE per diventare TRUE.

Esempio:

**NOT (TRUE = FALSE) = TRUE**

E:

**NOT (TRUE = TRUE) = FALSE**

### **EQV (&)**

L'operatore EQV viene utilizzato per eseguire un confronto logico tra due espressioni, in cui le espressioni sono nulle, o sono del sottotipo booleano e hanno un valore TRUE o FALSE.

L'operatore EQV può essere utilizzato anche un "operatore bit a bit" per rendere un confronto bit-by-bit di due numeri interi. Se entrambi i bit nel confronto sono uguali (entrambi sono 0 o 1), viene restituito un 1. In caso contrario, viene restituito uno 0.

L'ordine delle espressioni nel confronto non è importante.

Esempio:

**TRUE EQV TRUE = TRUE**

E:

**TRUE EQV FALSE = FALSE**



## Funzioni Primitive e Matematiche

Questo capitolo descrive le funzioni di base incorporate di EasyScript®, note anche come primitive. Queste importanti funzioni definiscono il linguaggio di programmazione EasyScript® e forniscono il quadro di base necessario per costruire trading system complessi da zero.

Ogni tipo di trading system può essere sviluppato utilizzando il linguaggio di programmazione EasyScript® con il minimo sforzo. Se un trading system può essere espresso in termini matematici o programmato in qualsiasi linguaggio strutturato o procedurale come il C++, VB, o Java per esempio, si può essere certi che le stesse formule possono anche essere scritte utilizzando il linguaggio di programmazione EasyScript®.

A volte le formule di analisi tecnica possono essere molto complesse, come per esempio, le funzioni di analisi tecnica che richiedono calcoli ricorsivi e strutture complicate (IF-THEN-ELSE) come parte della loro formula. Questi trading system complessi sono tradizionalmente sviluppati in un linguaggio di programmazione a basso livello.

Questo capitolo descrive come EasyScript® può essere utilizzato per eseguire questi stessi calcoli in un modo molto più semplice per mezzo di operazioni vettoriali e di strutture di controllo simulato.

## Funzioni Primitive

### AVG

**AVG (@vector, @periods)**

Restituisce un vettore contenente una media corrente, come specificato dall'argomento periods.

Esempio:

**AVG (CLOSE, 10)**

Lo script restituiscono un vettore di medie dei prezzi CLOSE sulla base di un periodo di 10 barre.

### AVGOF

**AVGOF (@vector1, @vector2, [@vector3], ..., [@vector8])**

Restituisce un vettore contenente una media tra i vettori indicati, fino ad un massimo di otto.

Esempio:

**AVGOF (CLOSE, REF(CLOSE, 1) , REF(CLOSE, 2), REF(CLOSE, 3), REF(CLOSE, 4))**

Lo script restituiscono un vettore di medie dei prezzi CLOSE di oggi, di un periodo fa, di due periodi fa, di tre periodi fa e di quattro periodi fa.

### BARLOOP

**BARLOOP (@initialValue, @offset, @operator, @operand, @minimumValue, @maximumValue)**

Restituisce un valore numerico calcolato progressivamente, barra per barra. Il parametro initialValue rappresenta il valore iniziale da utilizzare nel calcolo: può essere un qualsiasi valore numerico. Il parametro offset specifica il numero di barre trascorse a cui voglio riferirmi, deve essere un valore positivo. Il parametro operator identifica l'operatore matematico da utilizzare, tra ADD, SUBSTRACT, MULTIPLY e DIVIDE (vedi funzione LOOP). Il parametro operand specifica il secondo operando dell'operazione matematica da eseguire, e può essere una qualsiasi espressione (valore numerico, risultato di un'altra funzione, ecc.). I parametri minimumValue e maximumValue rappresentano l'eventuale

range minimo e massimo dei valori in uscita dalla funzione. Se non è necessario limitare il valore minimo o massimo del risultato, è possibile utilizzare NAN.

Esempio:

**SET EntryLB = BARLOOP (20, 1, MULTIPLY, (1 + DeltaHistVol), MinLB, MaxLB)**

Il risultato sarà un valore numerico calcolato progressivamente, barra per barra. Il calcolo utilizza 20 come valore iniziale. Il numero 1 rappresenta il valore della barra precedente, che viene moltiplicato per (1 + DeltaHistVol), e limitato entro il range MinLB e MaxLB.

## **CHANGEIF**

**CHANGEIF (@condition, @vector)**

Restituisce un vettore contenente il vettore imputato se la condizione viene verificata, in caso contrario restituisce il valore calcolato sulla barra precedente.

Esempio:

**CHANGEIF (CLOSE > OPEN, HIGH)**

Il risultato sarà un vettore contenente il valore HIGH se la condizione CLOSE > OPEN viene verificata, altrimenti sarà il valore calcolato sulla barra precedente.

Esempio di utilizzo:

**SET PREV = REF(HIGH, 1)  
CHANGEIF (HIGH > PREV, HIGH)**

EasyScript utilizza quindi il valore HIGH della barra attuale se è maggiore del valore HIGH della barra precedente. In caso contrario viene utilizzato il valore calcolato dalla funzione CHANGEIF sulla barra precedente. Può essere utile nel caso di costruzione di canali di prezzo.

## **COUNTIF**

**COUNTIF (@condition)**

Restituisce un vettore che rappresenta il numero totale di volte che la condizione specificata è risultata TRUE.

Esempio:

## **COUNTIF (CROSSOVER (SimpleMovingAverage (CLOSE, 14), CLOSE))**

Lo script restituisce un vettore con valori crescenti che esprimono il numero di volte in cui la Simple Moving Average a 14 periodi ha attraversato verso l'alto il CLOSE.

## **CROSSOVER**

### **CROSSOVER (@vector1, @vector2)**

Molti indicatori tecnici, come il MACD ad esempio, hanno una "linea di segnale". Un segnale BUY o SELL viene generato quando la linea segnale attraversa sopra o sotto l'indicatore.

La funzione CROSSOVER individua quando una serie ne ha attraversata un'altra al rialzo. Per esempio, possiamo trovare il punto esatto nel tempo in cui la Simple Moving Average ne ha attraversata un'altra utilizzando appunto la funzione CROSSOVER:

```
SET MA1 = SimpleMovingAverage (CLOSE, 28)
```

```
SET MA2 = SimpleMovingAverage (CLOSE, 14)
```

### **CROSSOVER (MA1, MA2)**

Lo script precedente restituisce true quando il vettore MA1 ha attraversato il vettore MA2 al rialzo.

E siamo in grado di invertire lo script utilizzando la funzione CROSSUNDER

### **CROSSUNDER (MA1, MA2)**

## **CROSSUNDER**

### **CROSSUNDER (@vector1, @vector2)**

Molti indicatori tecnici, come il MACD ad esempio, hanno una "linea di segnale". Un segnale BUY o SELL viene generato quando la linea segnale attraversa sopra o sotto l'indicatore.

La funzione CROSSUNDER individua quando una serie ne ha attraversata un'altra al ribasso. Per esempio, possiamo trovare il punto esatto nel tempo in cui la Simple Moving Average ne ha attraversata un'altra utilizzando appunto la funzione CROSSUNDER:

```
SET MA1 = SimpleMovingAverage (CLOSE, 28)
```

```
SET MA2 = SimpleMovingAverage (CLOSE, 14)
```

### **CROSSUNDER (MA1, MA2)**

Lo script precedente restituisce true quando il vettore MA1 ha attraversato il vettore MA2 al ribasso.

E siamo in grado di invertire lo script utilizzando la funzione CROSSOVER:

### **CROSSOVER (MA1, MA2)**

## **FOLLOWME**

### **FOLLOWME ()**

Restituisce TRUE se la funzione FOLLOWME è in territorio positivo, quindi trend rialzista.

Esempio:

### **FOLLOWME () > 0**

La funzione FOLLOWME può essere utilizzata per identificare il trend, attraverso un algoritmo proprietario, che può essere rialzista o ribassista.

Il valore di ritorno della funzione FOLLOWME rappresenta la forza e la direzione del trend, ed è sempre compreso all'interno di due intervalli:

da +50 a +100 per un trend rialzista oppure da -50 a -100 per un trend ribassista.

## **IF**

### **IF (@condition, @truePart, @falsePart)**

La funzione condizionale "IF" consente di progettare complessi filtri di logica booleana. Se si incolla il seguente script nell'area script nell'applicazione beeTrader®, il vettore A conterrà una serie di numeri che oscillano tra 1 e -1, a seconda di quando il prezzo di chiusura è superiore al prezzo di apertura:

### **SET A = IF (CLOSE > OPEN, 1, -1)**

Il primo argomento della funzione "IF" è un test logico. Il secondo argomento è il valore che verrà utilizzato se la condizione restituisce TRUE. Al contrario, il terzo argomento è il valore che verrà utilizzato se la condizione è FALSE. Il test logico può essere qualsiasi valore o espressione che può dare come risultato VERO o FALSO. Ad esempio, CLOSE = OPEN è un'espressione logica; se il prezzo di chiusura è uguale al prezzo di apertura, l'espressione restituisce TRUE. In caso contrario, l'espressione restituisce FALSE.

**NOTA:** la funzione IF può contenere una singola condizione logica, pertanto nell'espressione della condizione non possono apparire gli operatori AND, OR, NOT. La condizione logica non supporta funzioni nidificate.

## LASTIF

### LASTIF (@condition)

Simile a COUNTIF, tranne che LASTIF restituisce un vettore contenente il numero di barre dall'ultima volta che la condizione specificata è risultata TRUE. Il conteggio viene azzerato ogni volta che la condizione restituisce TRUE.

Esempio:

### LASTIF (CLOSE < OPEN)

Lo script restituisce un vettore che aumenta di valore per ciascuna barra in cui il CLOSE non era inferiore all'OPEN. Quando la condizione restituisce TRUE, significa che il CLOSE è inferiore all'OPEN, e il numero di riferimento viene azzerato.

**NOTA:** la funzione LASTIF può contenere una singola condizione logica, pertanto nell'espressione della condizione non possono apparire gli operatori AND, OR, NOT. La condizione logica non supporta funzioni nidificate.

## LOOP

### LOOP (@vector1, @vector2, @offset1, @offset2, @operator)

La funzione "LOOP" fornisce una struttura di controllo di flusso simulata con una singola chiamata a funzione. Consente di eseguire calcoli matematici su un intero vettore di dati in un singolo passaggio.

Vector1 è il vettore che contiene il primo operando dell'operazione. Offset1 è la posizione di Vector1 a partire dalla quale iniziare il calcolo incrementale. Vector2 è il vettore che contiene il secondo operando dell'operazione. Offset2 è la posizione di Vector2 a partire dalla quale iniziare il calcolo incrementale.

Gli operatori validi sono: ADD, SUBTRACT, MULTIPLY e DIVIDE

Esempio:

### LOOP (X, 2, 1, 0, MULTIPLY)

X (Vector1) è una serie di valori che iniziano da 5.25 fino a 9.25.

il risultato ottenuto conterrà valori a partire da X con offset 1 moltiplicati per 2

X	LOOP
5.25	5.25
6.25	10.5
7.25	21
8.25	42
9.25	84

## **MAX**

### **MAX (@vector, @periods)**

Restituisce un vettore contenente il valore massimo presente nelle ultime barre, come specificato dall'argomento Periods.

Esempio:

### **MAX (CLOSE, 10)**

Lo script restituisce un vettore contenente il valore massimo dei prezzi CLOSE sulla base di 10 periodi.

## **MAXOF**

### **MAXOF (@vector1, @vector2, [@vector3], ..., [@vector8])**

Restituisce un vettore contenente il valore massimo tra tutti i vettori specificati, per un massimo di otto vettori. Vector1 e Vector2 sono necessari mentre i vettori da 3 a 8 sono opzionali.

Esempio:

### **MAXOF (CLOSE, OPEN)**

Restituisce un vettore contenente il valore massimo tra OPEN e CLOSE di ciascuna barra.

## **MIN**

### **MIN (@vector, @periods)**

Restituisce un vettore contenente il valore minimo presente nelle ultime barre, come specificato dall'argomento Periods.

Esempio:

### **MIN (CLOSE, 10)**

Lo script restituisce un vettore contenente il valore minimo dei prezzi CLOSE sulla base di 10 periodi.

## **MINOF**

### **MINOF (@vector1, @vector2, [@vector3], ..., [@vector8])**

Restituisce un vettore contenente il valore minimo tra tutti i vettori specificati, per un massimo di otto vettori. Vector1 e Vector2 sono necessari mentre i vettori da 3 a 8 sono opzionali.

Esempio:

#### **MINOF (CLOSE, OPEN)**

Restituisce un vettore contenente il valore minimo tra OPEN e CLOSE di ciascuna barra.

#### **NTHMAXOF**

##### **NTHMAXOF (@position, @vector1, [@vector2], ..., [@vector7])**

Restituisce un vettore contenente l'ennesimo valore massimo tra tutti i vettori specificati, per un massimo di sette vettori. Il primo parametro identifica il valore N desiderato.

Esempio:

##### **NTHMAXOF (2, CLOSE, REF(CLOSE, 1) , REF(CLOSE, 2))**

Restituisce un vettore contenente il valore secondo massimo, quindi tra il CLOSE attuale, di 1 periodo fa e di 2 periodi fa.

#### **NTHMINOF**

##### **NTHMINOF (@position, @vector1, [@vector2], ..., [@vector7])**

Restituisce un vettore contenente l'ennesimo valore minimo tra tutti i vettori specificati, per un massimo di sette vettori. Il primo parametro identifica il valore N desiderato.

Esempio:

##### **NTHMINOF (2, CLOSE, REF(CLOSE, 1) , REF(CLOSE, 2))**

Restituisce un vettore contenente il valore secondo minimo, quindi tra il CLOSE attuale, di 1 periodo fa e di 2 periodi fa.

#### **PRINT**

##### **PRINT (@vector1, @vector2, [@vector3], ..., [@vector8])**

Nella finestra Debug Window di EasyScript Editor verrà visualizzato il valore



della variabile A, ovvero il CLOSE.

Esempio:

```
SET A = CLOSE  
PRINT(A)
```

La funzione PRINT è utilizzata per stampare i valori di una variabile dello script nell'apposita finestra Debug Window di EasyScript Editor.

NOTA: nel caso di Expert Advisor o Signal è consigliato utilizzare la funzione PRINT in una sola scheda per volta (Buy Script oppure Sell Script oppure ExitLong Script oppure ExitShort Script). In questo modo nella finestra Debug Window di EasyScript Editor verranno visualizzati valori relativi esclusivamente alla scheda dove è presente la funzione PRINT.

## **REF**

```
REF (@vector, @periods)
```

Per default tutti i calcoli vengono eseguiti sul valore più recente di un vettore, il LAST. Lo script seguente restituisce TRUE quando l'ultimo prezzo di apertura (OPEN) è inferiore a 30:

```
OPEN < 30
```

OPEN di default viene considerato sulla barra corrente. È possibile fare riferimento a un valore precedente di un vettore utilizzando la funzione REF:

```
REF (OPEN, 1) < 30
```

Ora lo script considera l'OPEN della barra precedente e restituisce TRUE se è inferiore a 30.

Il numero 1 (il secondo argomento) indica alla funzione REF di fare riferimento ad 1 barra fa. Per fare riferimento a valori di due barre fa, è sufficiente utilizzare 2 invece di 1. L'intervallo valido per l'argomento periods è 1-250 se non diversamente specificato.

## **SUM**

```
SUM (@vector, @periods)
```

La funzione SUM (da non confondere con la funzione SUMIF) restituisce un vettore contenente una somma parziale, come specificato dall'argomento periods.

Esempio:

## **SUM (CLOSE, 10)**

Lo script restituisce un vettore contenente, barra per barra, le somme dei prezzi di chiusura delle 10 barre precedenti.

## **SUMIF**

### **SUMIF (@condition, @vector)**

SUMIF è un'ulteriore funzione condizionale, questa funzione esegue una somma parziale di tutti i valori del vettore fornito laddove la condizione è risultata TRUE.

Per esempio, se vogliamo un vettore contenente la somma dei volumi per tutte le barre in cui il CLOSE è aumentato del 5%, si potrebbe scrivere:

```
SET VALORE = REF(CLOSE,1) * 1.05  
SUMIF (CLOSE > VALORE, VOLUME)
```

Il risultato sarà un vettore contenente una somma parziale dei volumi per ogni barra in cui il CLOSE è aumentato del 5% rispetto al CLOSE precedente.

## **SUMOF**

### **SUMOF (@vector1, @vector2, [@vector3], ..., [@vector8])**

Restituisce un vettore contenente la somma tra i vettori indicati, fino ad un massimo di otto.

Esempio:

```
SUMOF (CLOSE, REF(CLOSE, 1) , REF(CLOSE, 2), REF(CLOSE, 3), REF(CLOSE, 4))
```

Lo script restituiscono un vettore della somma dei prezzi CLOSE di oggi, di un periodo fa, di due periodi fa, di tre periodi fa e di quattro periodi fa.

## **TREND**

### **TREND (@vector, @periods, @percent)**

La funzione TREND può essere utilizzata per identificare il TREND che può essere rialzista, ribassista o laterale. Questa funzione può essere utilizzata su un prezzo (OPEN, HIGH, LOW, CLOSE), sul VOLUME, o su qualsiasi altro vettore, e richiede l'impostazione di un periodo di tempo su cui effettuare il calcolo. Il parametro @percent è opzionale, rappresenta lo scostamento minimo rispetto al valore precedente, se non viene impostato nulla viene

utilizzato il valore 0.5.

Il risultato della funzione TREND può assumere soltanto uno tra i valori UP, DOWN o SIDEWAYS.

**TREND (CLOSE, 30, 0.1) = UP AND TREND (VOLUME, 30) = DOWN**

Restituisce TRUE se il TREND dei prezzi CLOSE a 30 periodi è in salita di almeno lo 0.1% e contemporaneamente il TREND del VOLUME a 30 periodi è in discesa.

TREND è spesso la prima funzione utilizzata per filtrare i titoli che non sono nella direzione desiderata.

**SET T = TREND (CLOSE, 30)  
SET PLOT1 = IF (T = UP, 1, 0)  
SET PLOT1 = IF (T = DOWN, -1, PLOT1)**

In questo esempio TREND viene utilizzato per costruire un indicatore, che fornisce il valore 1 se il TREND è UP, il valore -1 se il TREND è DOWN, 0 se il TREND è SIDEWAYS.

## **TRUEHIGH**

**TRUEHIGH ()**

Restituisce un vettore contenente il valore maggiore tra il prezzo CLOSE della barra precedente ed il valore HIGH della barra attuale.

Esempio:

**TRUEHIGH () > SMA (CLOSE, 14)**

Restituisce TRUE se il valore TRUEHIGH è maggiore della Simple Moving Average calcolata su 14 periodi.

## **TRUELOW**

**TRUELOW ()**

Restituisce un vettore contenente il valore minore tra il prezzo CLOSE della barra precedente ed il valore LOW della barra attuale.

Esempio:

**TRUELOW () < SMA (CLOSE, 14)**

Restituisce TRUE se il valore TRUELOW è minore della Simple Moving Average calcolata su 14 periodi.

## Funzioni Matematiche

Tutte le funzioni matematiche restituiscono un vettore. Per esempio **ABS (CLOSE - OPEN)** restituisce un vettore del valore assoluto **CLOSE - OPEN** (un dato per barra), mentre la funzione **RND** restituisce un vettore di valori casuali, uno per ogni barra, e così via.

### **ABS - POS**

**ABS (@vector)**  
**POS (@vector)**

La funzione ABS restituisce il valore assoluto di un numero. I numeri negativi diventano numeri positivi, mentre i numeri positivi rimangono positivi.

Esempio:

**ABS (CLOSE - OPEN)**  
**POS (CLOSE - OPEN)**

Lo script restituisce sempre un numero positivo, anche se il prezzo di apertura è maggiore del prezzo di chiusura.

### **ARCCOS**

**ARCCOS (@vector)**

La funzione ARCCOS restituisce l'arcocoseno di un numero (angolo espresso in radianti).

Esempio:

**ARCCOS (0,707)**

Lo script emette il valore equivalente a  $\pi/4$  (0.785).

### **ARCSIN**

**ARCSIN (@vector)**

La funzione ARCSIN restituisce l'arcoseno di un numero (angolo espresso in radianti).

Esempio:

## **ARCSIN (0,707)**

Lo script emette il valore equivalente a  $\text{PI}/4$  (0.785).

## **ARCTAN**

### **ARCTAN (@vector)**

La funzione ARCTAN restituisce l'arcotangente di un numero (angolo espresso in radianti).

Esempio:

### **ARCTAN (1)**

Lo script emette il valore equivalente a  $\text{PI}/4$  (0.785).

## **CEIL**

### **CEIL (@vector)**

La funzione CEIL restituisce il valore d'ingresso arrotondato all'intero superiore.

Esempio:

### **CEIL (13.2)**

Lo script emette 14.

## **COS**

### **COS (@vector)**

La funzione COS restituisce il coseno di un numero (angolo espresso in radianti).

Esempio:

### **COS (PI/4)**

Lo script emette 0.707.

## **DEGTORAD**

### **DEGTORAD (@degrees)**

La funzione DEG TORAD restituisce il valore espresso in radianti di un angolo espresso in gradi.

Esempio:

**DEG TORAD(45)**

Lo script emette il valore equivalente a  $\pi/4$  (0.785).

## **EXP**

**EXP (@vector)**

La funzione EXP calcola la potenza esponenziale di un numero ( $e^x$ ). La funzione LOG esegue l'operazione inversa.

Esempio:

**EXP (3.26)**

Lo script emette 26.28

## **FLOOR**

**FLOOR (@vector)**

La funzione FLOOR restituisce il valore d'ingresso arrotondato all'intero inferiore.

Esempio:

**FLOOR (13.8)**

Lo script emette 13.

## **FRACPORTION**

**FRACPORTION (@vector)**

La funzione FRACPORTION restituisce la sola parte frazionaria di un numero.

Esempio:

**FRACPORTION (16.789)**

Lo script restituisce 0.789, cioè la parte frazionaria del valore 16.789.

## ***INTPORTION***

**INTPORTION (@vector)**

La funzione INTPORTION restituisce la sola parte intera di un numero.

Esempio:

**INTPORTION (16.789)**

Lo script restituisce 16, cioè la parte intera del valore 16.789.

## ***ISLASTBAR***

**ISLASTBAR()**

La funzione ISLASTBAR è utilizzata per identificare l'ultima barra di dati storici.

Esempio:

**SET C = ISLASTBAR() AND CLOSE > OPEN**

La variabile C conterrà il valore TRUE soltanto nell'ultima barra di dati se il prezzo CLOSE è maggiore del prezzo OPEN, in tutti gli altri casi ed in tutte le altre barre conterrà il valore FALSE.

## ***LOG***

**LOG (@vector)**

La funzione LOG restituisce il logaritmo naturale di un numero positivo. La funzione EXP esegue l'operazione inversa.

Esempio:

**LOG (26.28)**

Lo script emette 3.26

## ***LOG10***

**LOG10 (@vector)**

Restituisce il logaritmo in base 10 di un numero positivo.

Esempio:

**LOG10 (26.28)**

Lo script emette 1.42

## **MOD**

**MOD (@numerator, @denominator)**

La funzione MOD calcola il resto della divisione tra numeri interi. La funzione MOD richiede due parametri, il dividendo e il divisore.

Esempio:

**MOD (12, 5)**

Lo script restituisce 2, cioè il resto della divisione di 12 per 5.

## **NEG**

**NEG (@vector)**

La funzione NEG restituisce il valore assoluto di un numero moltiplicato per -1. I numeri positivi diventano numeri negativi, mentre i numeri negativi rimangono negativi.

Esempio:

**NEG (CLOSE - OPEN)**

Lo script restituisce sempre un numero negativo, anche se il prezzo di apertura è minore del prezzo di chiusura.

## **POW**

**ABS (@base, @exponent)**

La funzione POW calcola l'elevamento a potenza del valore impostato. La funzione POW richiede due argomenti, la base e l'esponente.

Esempio:

**POW (10, 2)**



Lo script emette 100, 10 elevato alla potenza di 2.

## ***RADTODEG***

**RADTODEG (@radiants)**

La funzione RADTODEG restituisce il valore espresso in gradi di un angolo espresso in radianti.

Esempio:

**RADTODEG (PI/4)**

Lo script emette 45.

## ***RND - RANDOM***

**RND (@vector)**

**RANDOM (@vector)**

La funzione RND restituisce un numero casuale compreso tra 0 e un valore massimo.

Esempio:

**RND (100)**

**RANDOM (100)**

Lo script emette un numero casuale da 0 a 100.

## ***ROUND***

**ROUND(@vector, @digits)**

la funzione ROUND calcola l'arrotondamento di un numero. Rispetto alla funzione ROUNDTICK si differenzia perchè permette di settare il numero di cifre decimale al quale arrotondare. La funzione ROUND richiede due parametri, il numero da arrotondare e il numero di cifre decimali da utilizzare nell'arrotondamento.

Esempio:

**ROUND (45,6789, 1)**

**ROUNDTICK (45.6789, 0,2)**

Lo script ROUND restituisce 45,7, ovvero il valore 45,6789 arrotondato ad 1

cifra decimale mentre lo script `ROUNDTICK` restituisce 45,6, ovvero il valore 45.6789 arrotondato al valore 0,2 più vicino.

## ***ROUNDTICK***

**`ROUNDTICK(@vector, @tick)`**

la funzione `ROUNDTICK` calcola l'arrotondamento di un vettore rispetto al tick impostato. Rispetto alla funzione `ROUND` si differenzia perchè permette di settare l'intervallo di arrotondamento. La funzione `ROUNDTICK` richiede due parametri, il numero da arrotondare ed il valore del tick da utilizzare nell'arrotondamento.

Esempio:

**`ROUNDTICK (45.6789, 0,2)`**  
**`ROUND (45,6789, 1)`**

Lo script `ROUNDTICK` restituisce 45,6, ovvero il valore 45.6789 arrotondato al valore 0,2 più vicino, mentre lo script con `ROUND` restituisce 45,7, ovvero il valore 45,6789 arrotondato ad 1 cifra decimale.

## ***SIGN***

**`SIGN (@vector)`**

La funzione `SIGN` identifica il segno di un numero.

Esempio:

**`SIGN (100)`**  
**`SIGN (0)`**  
**`SIGN (-100)`**

Lo script restituisce 1 per il primo caso, 0 per il secondo caso, -1 per il terzo.

## ***SIN***

**`SIN (@vector)`**

La funzione `SIN` restituisce il seno di un numero (angolo espresso in radianti).

Esempio:

**`SIN (PI/4)`**

Lo script emette 0.707.

## **SQRT - SQUAREROOT**

**SQRT (@vector)**  
**SQUAREROOT (@vector)**

La funzione SQRT restituisce la radice quadrata di un numero.

Esempio:

**SQRT (100)**  
**SQUAREROOT (100)**

Lo script emette 10, la radice quadrata di 100.

## **SQUARE**

**SQUARE (@vector)**

La funzione SQUARE calcola l'elevamento a potenza di 2 del valore impostato. La funzione SQUARE richiede un solo argomento, la base, mentre l'esponente è sempre pari a 2.

Esempio:

**SQUARE (5)**

Lo script emette 25, 5 elevato alla potenza di 2.

## **TAN**

**TAN (@vector)**

La funzione TAN restituisce la tangente di un numero (angolo espresso in radianti).

Esempio:

**TAN (PI/4)**

Lo script emette 1.

## **TODAYHIGH**

### **TODAYHIGH ()**

La funzione TODAYHIGH restituisce sempre il prezzo HIGH della giornata, non è dipendente dal timeframe utilizzato.

Esempio:

### **HIGH = TODAYHIGH()**

Lo script restituisce TRUE se il prezzo HIGH della barra attuale è il prezzo HIGH di giornata.

## **TODAYLOW**

### **TODAYLOW ()**

La funzione TODAYLOW restituisce sempre il prezzo LOW della giornata, non è dipendente dal timeframe utilizzato.

Esempio:

### **LOW = TODAYLOW()**

Lo script restituisce TRUE se il prezzo LOW della barra attuale è il prezzo LOW di giornata.

## **TODAYOPEN**

### **TODAYOPEN ()**

La funzione TODAYOPEN restituisce sempre il prezzo OPEN della giornata, non è dipendente dal timeframe utilizzato.

Esempio:

### **OPEN > TODAYOPEN()**

Lo script restituisce TRUE se il prezzo OPEN della barra attuale è maggiore del prezzo OPEN di giornata.

## **DIV**

### **DIV (@numerator, @denominator)**

La funzione DIV esegue la divisione tra numeri interi. La funzione DIV richiede due parametri, il dividendo e il divisore.

Esempio:

**DIV (12, 5)**

Lo script restituisce 2, cioè solo il numero intero del risultato della divisione di 12 per 5.

## **YESTERDAYCLOSE**

**YESTERDAYCLOSE ()**

La funzione YESTERDAYCLOSE restituisce il prezzo CLOSE del giorno precedente. Questa particolare funzione non viene influenzata dal timeframe selezionato, in quanto si riferisce solo e sempre al giorno precedente

Esempio:

**LAST > YESTARDAYCLOSE()**

Lo script restituisce TRUE se il prezzo attuale di oggi è maggiore del CLOSE di ieri.

## **YESTERDAYHIGH**

**YESTERDAYHIGH ()**

La funzione YESTERDAYHIGH restituisce il prezzo massimo del giorno precedente. Questa particolare funzione non viene influenzata dal timeframe selezionato, in quanto si riferisce solo e sempre al giorno precedente

Esempio:

**HIGH > YESTARDAYHIGH()**

Lo script restituisce TRUE se il prezzo HIGH di oggi è maggiore di quello di ieri.

## **YESTERDAYLOW**

**YESTERDAYLOW ()**

La funzione YESTERDAYLOW restituisce restituisce il prezzo minimo del giorno precedente. Questa particolare funzione non viene influenzata dal timeframe

selezionato, in quanto si riferisce solo e sempre al giorno precedente.

Esempio:

**LOW < YESTARDAYLOW()**

Lo script restituisce TRUE se il prezzo LOW di oggi è minore di quello di ieri.

## ***YESTERDAYOPEN***

**YESTERDAYOPEN ()**

La funzione YESTERDAYOPEN restituisce il prezzo di apertura del giorno precedente. Questa particolare funzione non viene influenzata dal timeframe selezionato, in quanto si riferisce solo e sempre al giorno precedente.

Esempio:

**OPEN > YESTARDAYOPEN()**

Lo script restituisce TRUE se il prezzo OPEN di oggi è maggiore di quello di ieri.

## ***HeikinAshiCLOSE***

**HeikinAshiCLOSE  
HACLOSE**

Il vettore HACLOSE restituisce il prezzo CLOSE della candela Heikin Ashi del periodo.

Esempio:

**HACLOSE > SimpleMovingAverage(HACLOSE, 20)**

Lo script restituisce TRUE se il prezzo CLOSE di Heikin Ashi è sopra la sua Simple Moving Average a 20 periodi.

## ***HeikinAshiHIGH***

**HeikinAshiHIGH  
HAHIGH**

Il vettore HAHIGH restituisce il prezzo HIGH della candela Heikin Ashi del periodo.

Esempio:

**HAHIGH > REF(HAHIGH, 1)**

Lo script restituisce TRUE se il prezzo HIGH di Heikin Ashi è in salita.

### ***HeikinAshiLOW***

**HeikinAshiLOW**

**HALOW**

Il vettore HALOW restituisce il prezzo LOW della candela Heikin Ashi del periodo.

Esempio:

**HALOW < REF(HALOW, 1)**

Lo script restituisce TRUE se il prezzo LOW di Heikin Ashi è in discesa.

### ***HeikinAshiOPEN***

**HeikinAshiAOPEN**

**HAOPEN**

Il vettore HAOPEN restituisce il prezzo OPEN della candela Heikin Ashi del periodo.

Esempio:

**LAST > HAOPEN**

Lo script restituisce TRUE se il prezzo LAST è maggiore del prezzo OPEN di Heikin Ashi.

## Funzioni di Analisi Tecnica ed Indicatori

Gli strumenti finanziari possono essere analizzati per mezzo di due tipi di analisi, fondamentale o tecnica e in questo modo si formano due categorie di Traders: coloro che analizzano titoli usando l'analisi fondamentale si basano su dati quali il rapporto prezzo/earnings (PE), dividendi, e resa e Traders che analizzano i titoli usando l'analisi tecnica, utilizzano grafici e su di essi applicano calcoli che spesso derivano dall'utilizzo di indicatori o oscillatori.

L'analisi tecnica è una forma di analisi di mercato che studia la domanda e l'offerta di strumenti finanziari in base ai volumi e all'andamento dei prezzi. I traders tentano di individuare i TREND dei prezzi in un mercato utilizzando uno o più indicatori tecnici.

Ci sono molti diversi tipi di indicatori tecnici e la maggior parte sono integrati come funzioni primitive nel linguaggio di programmazione EasyScript®, come descritto in questo capitolo ed è data massima ed ampia libertà di potersene costruire decine con calcoli propri. Infatti EasyScript® consente di programmare indicatori di analisi tecnica, utilizzando una combinazione di funzioni primitive e di analisi tecnica di cui al capitolo 2 e in questo capitolo.

In questo capitolo si trovano esempi e tecniche per la costruzione di indicatori personalizzati e sistemi di trading.

Quando presenti, le voci lowMark ed highMark identificano livelli significativi nei valori dei diversi indicatori.

Questo capitolo fornisce un elenco completo delle funzioni primitive di analisi tecnica che sono supportate dal linguaggio di programmazione EasyScript®.

Molte funzioni di analisi tecnica presentano nomi abbastanza lunghi, pertanto in molti casi è possibile utilizzare delle abbreviazioni.

Ecco un esempio:

Nome intero:

**SimpleMovingAverage (CLOSE, 30)**

Nome abbreviato:

**SMA (CLOSE, 30)**

SimpleMovingAverage e SMA rappresentano la stessa funzione di analisi tecnica e funzionano allo stesso modo.



## Moving Averages

Le Moving Averages (medie mobili) sono un fondamento dell'analisi tecnica. Queste funzioni calcolano le medie o variazioni delle medie del vettore sottostante. Molti indicatori di analisi tecnica si basano sulle caratteristiche di livellamento di medie mobili come parte del loro calcolo.

Questa sezione illustra la "Simple Moving Average", che è semplicemente un prezzo medio nel corso del tempo, la Exponential Moving Average, che è più complessa e dà più peso ai valori recenti, oltre a diversi altri tipi di medie mobili come Weighted Moving Average, Triangular Moving Average, Time Series Moving Average, e così via.

Ogni funzione di analisi tecnica che utilizza delle "Moving Averages" nel calcolo permette di scegliere quale tipo di "Moving Average" utilizzare.

Ad esempio, l'indicatore "Moving Average Convergence / Divergence (MACD)" di EasyScript® consente di specificare il tipo di "Moving Average" utilizzato all'interno del calcolo della "Signal Line" così come per il calcolo delle Bollinger Bands ecc.

### *Simple Moving Average*

**SimpleMovingAverage (@vector, @periods)**  
**SMA (@vector, @periods)**

MA Type: SIMPLE

Panoramica:

La Simple Moving Average è semplicemente una media di valori per un periodo di tempo specificato.

Interpretazione/Utilizzo:

Si utilizza una Moving Average per ottenere una rappresentazione uniforme dei prezzi degli strumenti finanziari o di altri indicatori di analisi tecnica.

Esempio:

**CLOSE > SMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della SMA a 30 periodi.

### *Exponential Moving Average*

**ExponentialMovingAverage (@vector, @periods)**  
**EMA (@vector, @periods)**

MA Type: EXPONENTIAL

Panoramica:

La Exponential Moving Average è simile ad una Simple Moving Average. La EMA è calcolata applicando una piccola percentuale del valore corrente al valore precedente, quindi una EMA applica un peso maggiore ai valori più recenti.

Interpretazione/Utilizzo:

Una Exponential Moving Average è utilizzata, perchè attraverso i valori medi, si ha una rappresentazione uniforme dei prezzi del sottostante o indicatore esattamente come nella SMA con però il vantaggio che è più reattiva e si riduce il grande problema del ritardo del segnale (lag).

Di contro, essendo meno "elastica", produrrà maggiori falsi segnali.

Esempio:

**CLOSE > EMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della EMA a 30 periodi.

### ***Time Series Moving Average***

**TimeSeriesMovingAverage (@vector, @periods)**

**TSMA (@vector, @periods)**

MA Type: TIME\_SERIES

Panoramica:

La Time Series Moving Average restituisce la media mobile di una serie di dati per un determinato periodo di tempo sulla base di una regressione lineare. (In pratica è identica a Time Series Forecast con offset = 0)

Interpretazione/Utilizzo:

Una Time Series Moving Average è meno influenzata dal valore attuale dei dati che si stanno analizzando e quindi elimina o riduce l'effetto "noise"

Esempio:

**CLOSE > TSMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della TSMA a 30 periodi.

### ***Variable Moving Average***

**VariableMovingAverage (@vector, @periods)**

**VMA (@vector, @periods)**

MA Type: VARIABLE

Panoramica:

Una Variable Moving Average è simile ad una Exponential Moving Average con la differenza che si adatta alla volatilità.

Interpretazione/Utilizzo:

Una Variable Moving Average è utilizzata quando si voglia inglobare nell'analisi anche la volatilità storica dello strumento preso in esame. Sarà quindi condizionata anche dall'estensione delle barre.

Esempio:

**CLOSE > VMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della VMA a 30 periodi.

### ***Triangular Moving Average***

**TriangularMovingAverage (@vector, @periods)**

**TMA (@vector, @periods)**

MA Type: TRIANGULAR

Panoramica:

La Triangular Moving Average è simile ad una Simple Moving Average, tranne che viene dato maggior peso ai prezzi delle barre centrali del periodo preso in considerazione.

Interpretazione/Utilizzo:

Una Triangular Moving Average è utilizzata quando si vuole spostare il calcolo del ciclo dominante verso valori centrali del periodo utilizzato. Questo porta ad avere una reattività maggiore nell'indicazione del TREND, anche se per costruzione molto più smussata.

Esempio:

**CLOSE > TMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della TMA a 30 periodi.

### ***Weighted Moving Average***

**WeightedMovingAverage (@vector, @periods)**

**WMA (@vector, @periods)**

MA Type: WEIGHTED

Panoramica:

La Weighted Moving Average a differenza della Exponential Moving Average cala di 1 unità il peso del valore precedente vediamo il calcolo a 4 barre: i prezzi sono 1.2900, 1.2900, 1.2903, e 1.2904 che darebbero una media mobile di 1.2903 usando il seguente calcolo  $((4 * 1.2904) + (3 * 1.2903) + (2 * 1.2900) + (1 * 1.2900)) / (4 + 3 + 2 + 1) = 1.2903$

Interpretazione/Utilizzo:

Una Weighted Moving Average per avere una risposta media sulla base dati. Si cerca di ottenere meno ritardo e meno falsi segnali.

Esempio:

**CLOSE > WMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della WMA a 30 periodi.

### **Welles Wilder Smoothing Moving Average**

**WellesWilderSmoothing (@vector, @periods)**

**WWS (@vector, @periods)**

MA Type: WILDER

Panoramica:

La Welles Wilder Smoothing Moving Average è simile a una Exponential Moving Average. L'indicatore però non utilizza la formula standard della Exponential Moving Average. Usando ad esempio un periodo pari a 14 barre, la formula di Welles Wilder consiste nel pesare per 1/14 il valore dell'ultima barra e per 13/14 il valore alla barra precedente della Welles Wilder stessa.

Interpretazione/Utilizzo:

Anche se il calcolo è diverso l'interpretazione e l'utilizzo sono i medesimi di una qualsiasi altra Moving Average. Indica pertanto valori alti o bassi rispetto alla media dando indicazioni al trader se si stanno raggiungendo supporti o resistenze di periodo. Spesso si utilizzano come indicazioni di TREND ma non è la soluzione ideale perchè il calcolo stesso della Moving Average non può identificare TREND, come invece riesce ad identificare valori importanti di prezzo in entrambe le direzioni.

Esempio:

**CLOSE > WWS (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della WWS a 30 periodi.

## **VIDYA Moving Average**

**VIDYA (V@vector, @periods, @r2Scale)**

MA Type: VIDYA

Panoramica:

La Vidya Moving Average, sviluppata da Tushar Chande, è una Moving Average derivata dalla regressione lineare  $R^2$ .

Interpretazione:

Una Moving Average è utilizzata, perchè attraverso i valori medi, si ha una rappresentazione uniforme dei prezzi del sottostante o indicatore. Poiché la VIDYA è un derivato della regressione lineare, si adatta rapidamente alla volatilità.

Parametri Standard:

R2Scale è un valore che specifica la scala  $R^2$  da utilizzare nei calcoli di regressione lineare. Chande consiglia un valore compreso tra 0,5 e 0,8 (valore di default è 0,65).

Esempio:

**CLOSE > VIDYA (CLOSE, 30, 0.65)**

Restituisce TRUE quando il CLOSE è maggiore della VIDYA a 30 periodi con un  $R^2$  di 0,65.

## **Double Exponential Moving Average**

**DoubleExponentialMovingAverage (@vector, @periods)**  
**DEMA (@vector, @periods)**

MA Type: DOUBLE

Panoramica:

La Double Exponential Moving Average è calcolata sulla base di una Exponential Moving Average, tuttavia non è semplicemente basata sulla Moving Average di una Moving Average, ma è una combinazione di due Moving Averages una Exponential Moving Averages con il doppio di una Exponential Moving Averages.

Interpretazione/Utilizzo:

Una Double Exponential Moving Average è utilizzata, perchè è una Moving Average più veloce e più adattiva sviluppata con lo scopo di ridurre l'effetto lag prodotto dalle Moving Averages tradizionali.

Esempio:

**CLOSE > DEMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della DEMA a 30 periodi.

### ***Triple Exponential Moving Average***

**TripleExponentialMovingAverage (@vector, @periods)**

**TEMA (@vector, @periods)**

MA Type: TRIPLE

Panoramica:

La Triple Exponential Moving Average è calcolata sulla base di una Exponential Moving Average, tuttavia non è semplicemente basata sulla Moving Average di una Moving Average, ma è una combinazione di tre Moving Averages una Exponential Moving Averages con il triplo di una Exponential Moving Averages.

Interpretazione/Utilizzo:

Una Triple Exponential Moving Average è utilizzata, perchè è ancor più veloce e più adattiva rispetto alla Double Exponential Moving Averages sviluppata con lo scopo di ridurre l'effetto lag prodotto dalle Moving Averages tradizionali.

Esempio:

**CLOSE > TEMA (CLOSE, 30)**

Restituisce TRUE quando il CLOSE è maggiore della TEMA a 30 periodi.

## Linear Regression

Un classico problema statistico è quello di cercare di determinare la relazione tra due variabili casuali X e Y, come il prezzo di chiusura di un titolo nel corso del tempo. La Linear Regression cerca di spiegare il rapporto con una linea retta che si adatta ai dati.

Il modello di regressione lineare descrive che  $Y = a + bX + e$

Quando il "residuo" e è una variabile casuale con media zero, i coefficienti A e B sono determinati dalla condizione che la somma dei residui quadrati sia il più piccolo possibile. Gli indicatori di questa sezione si basano su questo modello.

### **Forecast**

**Forecast (@vector, @periods)**

**LR (@vector, @periods)**

**LinearReg (@vector, @periods)**

**LinearRegressionForecast (@vector, @periods)**

Panoramica:

Restituisce le previsioni di Linear Regression per il periodo successivo, in base al calcolo della Regressione Lineare sul periodo specificato.

Esempio:

**Forecast (CLOSE, 30) > REF (CLOSE,1)**

Restituisce TRUE quando la previsione è superiore al prezzo di chiusura precedente.

### **Intercept**

**Intercept (@vector, @periods)**

**LinearRegressionIntercept (@vector, @periods)**

Panoramica:

Restituisce la posizione di intersezione della Linear Regression con l'asse Y, sulla base del calcolo della regressione lineare su un determinato periodo.

E' il valore (a) nel modello di regressione lineare descrive che  $Y = a + bX + e$

Esempio:

**Intercept (CLOSE, 21) > REF (CLOSE,1)**

Restituisce TRUE se l'intercetto è superiore al prezzo di chiusura precedente.

## ***R<sup>2</sup> (R-Squared)***

**RSquared (@vector, @periods)**

**R2 (@vector, @periods)**

**LinearRegressionRSquared (@vector, @periods)**

Panoramica:

R<sup>2</sup> è il coefficiente di determinazione per il vettore fornito nei periodi specificati. I valori oscillano tra 0 e 1.

Esempio:

**R2 (CLOSE, 30) < 0.1**

Restituisce TRUE quando il coefficiente di determinazione è inferiore a 0,1.

## ***Slope***

**Slope (@vector, @periods)**

**LinearRegressionSlope (@vector, @periods)**

Panoramica:

Restituisce il valore di pendenza della Linear Regression per i dati che vengono analizzati nel periodo specificato. I valori possono oscillare da negativo a positivo.

Esempio:

**SLOPE (CLOSE, 30) > 0.3**

Restituisce TRUE quando la pendenza è maggiore di 0,3.



## Bands Functions

Alcuni indicatori di analisi tecnica sono progettati per sovrapporsi ai grafici dei prezzi in modo da formare una fascia intorno al prezzo dello strumento. Generalmente un cambiamento di TREND è individuato dal fatto che il prezzo dello strumento attraversa una delle bande, in salita o in discesa. Il più popolare indicatore di questo tipo è "Bollinger Bands", sviluppato dall'operatore di borsa John Bollinger nei primi anni 1980.

### *Bollinger Bands*

**BollingerBandsTop (@vector, @periods, @standardDeviations, @maType)**  
**BBT (@vector, @periods, @standardDeviations, @maType)**

**BollingerBandsMiddle (@vector, @periods, StandardDeviations, MAType)**  
**BBM (@vector, @periods, @standardDeviations, @maType)**

**BollingerBandsBottom (@vector, @periods, StandardDeviations, MAType)**  
**BBB (@vector, @periods, @standardDeviations, @maType)**

Panoramica:

Le Bollinger Bands si basano su deviazioni standard al fine di adattarsi alle mutevoli condizioni del mercato. Quando un titolo diventa volatile, le bande si allargano (si allontanano dalla media). Al contrario, quando il mercato diventa meno volatile le bande si contraggono (avvicinarsi alla media). Una diminuzione di ampiezza delle bande è spesso usata come una prima indicazione che la volatilità del titolo sta per aumentare.

Interpretazione/Utilizzo:

Le Bollinger Bands (come con la maggior parte delle bande) possono essere impostate su un prezzo effettivo, o su un altro indicatore. Dopo che i prezzi sono saliti al di sopra della banda superiore, o sono scesi al di sotto della banda inferiore, il rientro dei prezzi all'interno delle bande genera un segnale d'inversione di TREND, SHORT nel primo caso, LONG nel secondo.

Parametri Standard:

Vector: **CLOSE**

Periods: **20**

Standard Deviations: **2**

MA Type: **EXPONENTIAL**

Esempio:

**CLOSE > BBT (CLOSE, 20, 2, EXPONENTIAL)**

Restituisce TRUE quando il CLOSE è maggiore della Bollinger Band Top a 20 periodi calcolato su 2 deviazioni standard, utilizzando una Exponential Moving Average.

## ***Fractal Chaos Bands***

**FractalChaosBandsTop (@periods)**  
**FCBT (@periods)**

**FractalChaosBandsBottom (@periods)**  
**FCBB (@periods)**

Panoramica:

Fractal Chaos Bands si basa sull'assunto (che beeTrader non condivide!) che il mercato si muove in un modo caotico e utilizzando la matematica e le funzioni del chaos, sono state create due bande

Interpretazione/Utilizzo:

Anche questo indicatore può essere usato come tutti gli altri indicatori a bande e utilizzato per identificare se il mercato è in una fase di TREND o di lateralità. Più la pendenza delle bande è accentuata e più il TREND è forte, più le bande sono "piatte" e più siamo in un mercato laterale o comunque calmo e poco volatile.

Parametri Standard:

Periods: **10**

Esempio:

**CLOSE > FractalChaosBandsTop (10)**

Restituisce TRUE quando il valore del CLOSE è superiore al valore della Fractal ChaosBandTop calcolata a 10 periodi, confermando un TREND LONG.

## ***High Low Bands***

**HighLowBandsTop (@periods)**  
**HLT (@periods)**

**HighLowBandsMedian (@periods)**  
**HLM (@periods)**

**HighLowBandsBottom (@periods)**  
**HLB (@periods)**

Panoramica:

Costruisce una banda, che si sovrappone al grafico, calcolata con la media mobile dei valori HIGH e LOW di ogni singola barra.

Interpretazione/Utilizzo:

Come tutti gli indicatori a bande, l'uscita da una delle due parti è indice di aumento di volatilità e conseguentemente di inizio di forti TREND. I valori che rimangono interne alle bande indicano o continuità di TREND o lateralità. In entrambi i casi il mercato è "abituato" e quindi non volatile.

Parametri Standard:

Periods: **10**

Esempio:

**CLOSE > HLT (10) OR CLOSE < HLB (10)**

Restituisce TRUE quando il CLOSE è all'esterno della banda disegnata dall'indicatore.

### ***Keltner Channel***

**KeltnerChannelTop (@periods, @maType, @multiplier)**

**KCT (@periods, @maType, @multiplier)**

**KeltnerChannelMedian (@periods, @maType, @multiplier)**

**KCM (@periods, @maType, @multiplier)**

**KeltnerChannelBottom (@periods, @maType, @multiplier)**

**KCB (@periods, @maType, @multiplier)**

Panoramica:

Il Keltner Channel è calcolato aggiungendo e togliendo al valore della Moving Average il valore dell' Average True Range.

Interpretazione/Utilizzo:

Come altri indicatori di questo tipo, il Keltner Channel può essere impostato su un prezzo effettivo, o su un altro indicatore. Il Keltner Channel fornisce un segnale LONG quando i prezzi sono al di sopra della banda superiore e fornisce un segnale SHORT quando i prezzi sono al di sotto della banda inferiore. Il Keltner Channel può essere interpretato allo stesso modo delle Bollinger Bands.

Parametri Standard:

Periods: **15**

MA Type: **EXPONENTIAL**

Multiplier: **1.3**

Esempio:

**CLOSE > KCT (15, EXPONENTIAL, 1.3)**

Restituisce TRUE quando il CLOSE è sopra la parte superiore del Keltner Channel.

## ***Moving Average Envelope***

**MovingAverageEnvelopeTop (@periods, @maType, @shift)**

**MAET (@periods, @maType, @shift)**

**MovingAverageEnvelopeBottom (@periods, @maType, @shift)**

**MAEB (@periods, @maType, @shift)**

Panoramica:

La Envelope Moving Average è costituita da Moving Averages calcolate dal prezzo dello strumento finanziario, spostata verso l'alto e verso il basso di una percentuale fissa.

Interpretazione/Utilizzo:

La Envelope Moving Average può essere impostata su un prezzo effettivo, o su un altro indicatore. Se impostata sul prezzo dello strumento si guarderà nella serie storica l'ampiezza delle barre che si sono formate, si tarerà lo shift della stessa ampiezza e così si potranno visualizzare (programmare) movimenti di prezzo al di sopra della volatilità media passata.

Parametri Standard:

Periods: **20**

MA Type: **SIMPLE**

Shift: **0,05**

Esempio:

**CLOSE > MAET (20, SIMPLE, 0,5)**

Restituisce TRUE quando il CLOSE è maggiore della Moving Average Envelope Top a 20 periodi calcolato sul 0,5% di una Simple Moving Average.

## ***Prime Number Bands***

**PrimeNumberBandsTop()**

**PNBT()**

**PrimeNumberBandsBottom()**

**PNBB()**

Panoramica:

Questo indicatore identifica il numero primo più vicino ai prezzi HIGH e LOW dopo averli trasformati in un valore compreso tra 10 e 100 ed usa i due valori ottenuti per disegnare due bande, una inferiore ed una superiore.

Interpretazione/Utilizzo:

Le bande sono rappresentate come delle linee continue ed è proprio il loro cambiamento, verso l'alto o verso il basso che indicherà cambiamenti di TREND.

*Nota: il loro funzionamento ricade nei "misteri" dei numeri primi tanto quanto l'apporto dei numeri di Fibonacci.*

Parametri Standard:  
Non previsti

Esempio:

**PNBT <> REF (PNBT,1) OR PNBB <> REF (PNBB,1)**

Restituisce TRUE quando una delle due bande ha una variazione rispetto alla prima barra precedente.

## **Starc**

**StarcTop (@periods, @maType, @multiplier)**

**StarcMedian (@periods, @maType, @multiplier)**

**StarcBottom (@periods, @maType, @multiplier)**

Panoramica:

Lo Starc Channel è calcolato aggiungendo e togliendo al valore della Moving Average il valore dell' Average True Range moltiplicata per un fattore o multiplier che di solito è 1,3.

Interpretazione/Utilizzo:

Come altri indicatori di questo tipo, lo Starc Channel può essere impostato su un prezzo effettivo, o su un altro indicatore. lo Starc Channel fornisce un segnale LONG quando i prezzi sono al di sopra della banda superiore e fornisce un segnale SHORT quando i prezzi sono al di sotto della banda inferiore. lo Starc può essere interpretato allo stesso modo delle Bollinger Bands.

Parametri Standard:

Periods: **15**

MA Type: **EXPONENTIAL**

Multiplier: **1.3**

Esempio:

**CLOSE > StarcTop (15, EXPONENTIAL, 1.3)**

Restituisce TRUE quando il CLOSE è sopra la banda superiore dello Starc Channel.

## Oscillatori

Questa sezione tratta indicatori tecnici che oscillano da un valore ad un altro. La maggior parte degli oscillatori misura la velocità di movimento di prezzo o di volume. Questi indicatori vanno spesso in zone di ipercomprato e ipervenduto, zone in cui è possibile un'inversione. La pendenza dell'oscillatore è solitamente proporzionale alla velocità del movimento di prezzo. Analogamente, la distanza dei movimenti dell'oscillatore è generalmente proporzionale alla grandezza del movimento.

Alcuni di loro possono anticipare il cambiamento del TREND che si rileva dalla loro divergenza: il titolo sta calando mentre l'indicatore cresce sta ad indicare un possibile cambiamento di TREND verso la salita, il titolo sale mentre l'indicatore scende sta ad indicare un possibile cambiamento di TREND verso la discesa.

### *Aroon*

**AroonUp (@periods)**  
**AroonDown (@periods)**

Panoramica:

L' Aroon Indicator è stato sviluppato da Tushar Chande nella metà del 1990. Questo indicatore viene spesso utilizzato per determinare se uno strumento è in TREND o meno e quanto il TREND è stabile.

Interpretazione/Utilizzo:

I TREND sono determinati dai valori estremi (sopra 80) di entrambe le linee (AroonUp e AroonDown), mentre i prezzi instabili vengono determinati quando entrambe le linee sono basse (meno di 20). Una buona lettura di un TREND LONG si avrà se  $AroonUp > 80$  e contemporaneamente  $AroonDown < 20$ . Viceversa per il TREND SHORT.

Parametri Standard:

Periods: **10**

Esempio:

**AroonUp (10) > 80 AND AroonDown (10) < 20**

Restituisce TRUE quando AroonUp è maggiore di 80 e, contemporaneamente, AroonDown è minore di 20.

### *Aroon Oscillator*

**AroonOsc (@periods)**

Panoramica:

L' Aroon Oscillator è un indicatore derivante direttamente da Aroon Indicator, misura la differenza tra Aroon Up e Aroon Down ed assume perciò valori che variano da 100 a -100.

Interpretazione/Utilizzo:

L'interpretazione più comune è quella di considerare inizio di TREND in salita quando l'indicatore ha valori superiori a 50 e TREND in discesa se ha dei valori superiori a -50.

Parametri Funzione

Periods            **10**

Parametri Indicatore beeTrader

Periods            **10**

lowMark           **-50**

highMark          **50**

Esempio:

### **CROSSOVER (AroonOsc(10), 50)**

Restituisce TRUE quando Aroon Oscillator attraversa la linea del 50 dal basso verso l'alto (ieri era minore di 50, oggi è maggiore di 50).

## **Average True Range**

**AverageTrueRange (@periods, @maType)**

**ATR (@periods, @maType)**

Panoramica:

Da come si è potuto leggere nel paragrafo precedente si sarà anche capito che il True Range fine a se stesso ha poca utilità pratica avendo necessità di essere comparato con valori precedenti. Ecco che troviamo l' Average True Range che viene calcolato applicando una Moving Average sull'indicatore True Range, misurando pertanto una volatilità storica.

Interpretazione/Utilizzo:

L' ATR non si interpreta ma si legge e si utilizza come filtro per evitare di entrare in mercati laterali, in sideways, che sono penalizzanti per il trader.

Altro utilizzo dell'ATR si basa sulla considerazione che la volatilità ritorna alla media, per cui possiamo pensare che ad una volatilità bassa seguirà una volatilità alta e viceversa. In pratica, se teniamo sotto controllo una serie di titoli, possiamo entrare a mercato con il giusto tempismo e usufruire di movimenti ampi di prezzo, oppure uscire da posizioni se la volatilità è scesa sotto i nostri parametri di guardia.

Parametri Standard:

Periods: **14**  
MA Type: **SIMPLE**

Esempio:

**ATR(21, SIMPLE) > 2**

Restituisce TRUE quando il valore dell'ATR è maggiore di 2.

### ***Chaikin Volatility***

**ChaikinVolatility (@periods, @rateOfChange, @maType)**  
**CV (@periods, @rateOfChange, @maType)**

Panoramica:

Il Chaikin Volatility è un oscillatore che quantifica la volatilità di un titolo usando la percentuale di variazione (Rate of Change) di una Moving Average calcolata con i prezzi highest high e lowest low di un determinato periodo. Calcola valori positivi e negativi, più i valori sono vicini allo zero e più il titolo è scarico di volatilità

Interpretazione/Utilizzo

Il Chaikin Volatility si adatta alla volatilità, indipendente dal prezzo dello strumento. L'interpretazione più popolare è quella di utilizzarlo come indicatore di top e bottom di mercato. Livelli di alta volatilità indicano che sta per arrivare l'inversione del TREND così come livelli di bassa volatilità indicano continuazione di TREND o e lateralità.

Parametri Standard:

Periods: **10**  
Rate of Change: **10**  
MA Type: **SIMPLE**

Esempio:

**CV (10, 10, SIMPLE) > 0**

Restituisce TRUE quando il valore della volatilità sarà maggiore di zero.

### ***Chande Forecast Oscillator***

**ChandeForecastOscillator (@vector, @periods)**  
**CFO (@vector, @periods)**

Panoramica:

Trova la differenza percentuale esistente tra il prezzo previsto, tramite la



regressione lineare, e il prezzo corrente.

Interpretazione:

Se l'indicatore è maggiore di zero si ipotizza la presenza di un TREND rialzista mentre quando l'indicatore è minore di zero si prevede un TREND ribassista.

Parametri Standard:

Vector: **CLOSE**

Periods: **21**

Esempio:

**CFO (CLOSE, 21) > 0**

Restituisce TRUE se il Chande Forecast Oscillator è in territorio positivo.

### ***Chande Momentum Oscillator***

**ChandeMomentumOscillator (@vector, @periods)**

**CMO (@vector, @periods)**

Panoramica:

Il Chande Momentum Oscillator (CMO), sviluppato da Tushar Chande, è un oscillatore di momentum avanzato derivato dalla regressione lineare.

Interpretazione/Utilizzo:

Il CMO entra in territorio di ipercomprato a 50, e in ipervenduto a -50. È possibile creare segnali BUY/SELL sulla base di una Moving Average del CMO.

Inoltre, valori sempre più elevati di CMO possono indicare che i prezzi sono in TREND fortemente rialzista. Al contrario, valori sempre più bassi di CMO possono indicare che i prezzi sono in TREND fortemente ribassista.

Altro utilizzo è l'attraversamento della linea dello zero: sopra lo zero il titolo sta iniziando una fase di TREND in salita, sotto lo zero indica un TREND di discesa.

Parametri Standard:

Vector: **CLOSE**

Periods: **14**

Esempio:

**REF (CMO (CLOSE, 14) ,1) > 48 AND CMO (CLOSE, 14) < 47**

Restituisce TRUE se il CMO della barra precedente è in fase di ipercomprato e se quella attuale è in discesa.

## ***Detrended Price Oscillator***

**DetrendedPriceOscillator (@vector, @periods, @maType)**  
**DPO (@vector, @periods, @maType)**

Panoramica:

Simile al Price Oscillator ad eccezione che il DPO viene utilizzato per TREND di lungo termine o valori anomali che rendono il prezzo del sottostante difficile da analizzare.

Interpretazione/Utilizzo:

Il segnale BUY si verifica quando l'oscillatore sale. Il segnale SELL si verifica quando l'oscillatore scende.

Parametri Standard:

Vector: **CLOSE**

Periods: **14**

MA Type: **SIMPLE**

Esempio:

**DPO (CLOSE, 14, SIMPLE) > 0**

Restituisce TRUE se il Detrended Price Oscillator è in territorio positivo.

## ***Directional Movement Index***

**ADX (@periods)**  
**DIP (@periods)**  
**DIN (@periods)**

Panoramica:

Il Directional Movement Index contiene tre indicatori, l'ADX, il DIP (DI+) e il DIM (DI-).

L'ADX (Average Directional Movement Index) rappresenta quanta forza è presente nel mercato, più il valore è alto e più è affidabile l'indicazione che perverrà dal DIP (Directional Positive) e dal DIM (Directional Negative).

Il DIP (DI+) misura la presenza di forze rialziste

Il DIM (DI-) misura la presenza di forze ribassiste

Interpretazione/Utilizzo:

Ci sono svariati modi di utilizzo, il più comune è: un segnale BUY viene dato quando DI+ attraversa la DI-, un segnale SELL è dato quando DI- attraversa DI+. In genere il segnale sia di BUY che di SELL viene confermato da un valore di ADX. In seguito, sempre con un valore di ADX storicamente alto o comunque in salita, si avrà un segnale di continuazione di TREND LONG se il DIP è superiore al DIM e un segnale di continuazione di TREND SHORT se il DIM è superiore al DIP.

Parametri Funzione

Periods **14**

Parametri Indicatore beeTrader

PeriodsADX **30**

PeriodsDI **14**

Esempio:

**CROSSOVER (DIN (14), DIP (14)) AND ADX > 18**

Restituisce TRUE quando avviene l'incrocio dal basso verso l'alto del DIN sul DIP e quando anche ADX ha un valore superiore a 18.

### ***Ease of Movement Oscillator***

**EaseOfMovement (@vector, @periods)**

**EOM (@vector, @periods)**

Panoramica:

L' Ease of Movement Oscillator è equiparabile alla funzione di Equivolume: segnala il rapporto che c'è tra il cambiamento dei prezzi ed il volume con cui questo cambiamento è avvenuto. Sappiamo che il volume "sostiene" il TREND per cui diventa interessante ed utile per individuare la forza del trend in atto.

Interpretazione:

Dato che L'Ease of Movement Oscillator, per costruzione, divide i prezzi per i volumi, il suo valore aumenta quando i prezzi tendono al rialzo con bassi volumi, e scende quando i prezzi tendono verso il basso con alti volumi. L'oscillatore sarà vicino allo zero se i prezzi si muovono poco, o se si muovono ma con volumi alti come se, appunto, per muoverli ci volesse una massa importante di scambi.

Quindi in un TREND in salita indica forza se il suo valore è sopra lo zero, in un TREND di discesa indica forza se il suo valore è sotto lo zero.

Parametri Standard:

Vector: **CLOSE**

Periods: **21**

Esempio:

**EOM (CLOSE, 21) > 0**

Restituisce TRUE quando Ease of Movement Oscillator è in territorio positivo.

## ***Elder Force Index***

**ElderForceIndex ()**  
**EFI ()**

Panoramica:

L'Elder Force Index parte dall'assunto che ci sono tre elementi essenziali in un movimento di prezzo: direzione, entità e volume. Questo indicatore tiene conto di tutti e tre questi fattori.

Interpretazione/Utilizzo:

La conferma della salita del valore dell'indicatore aiuterà nella convinzione che il TREND è in salita, così come la discesa del suo valore mi indurrà a prendere atto che il TREND sta finendo o è cambiato. I prezzi che salgono e l'indicatore che scende o viceversa, segnalano delle divergenze che ci faranno capire che il TREND sta per cambiare.

Parametri Standard:

Non previsti

Esempio:

**EFI() > 0**

Restituisce TRUE quando l'Elder Force Index è in territorio positivo.

## ***Fractal Chaos Oscillator***

**FractalChaosOscillator()**  
**FCO()**

Panoramica:

La natura caotica dei movimenti dei mercati azionari spiega il motivo per cui a volte è difficile distinguere i grafici giornalieri dai grafici mensili, se non viene evidenziato il timeframe.

I modelli sono quindi simili indipendentemente dalla risoluzione temporale: ogni livello è come quello precedente, ma la dimensione è diversa. Per determinare ciò che sta accadendo nel livello corrente della risoluzione, può essere utilizzato il Fractal Chaos Oscillator proprio per esaminare questi modelli.

Interpretazione:

Un segnale BUY viene generato quando l'oscillatore assume valori prossimi al massimo valore possibile, e un segnale SELL viene generato quando l'oscillatore assume valori prossimi al minimo valore possibile.

Parametri Standard:

Non previsti

Esempio:

```
FCO() > REF (FCO(), 1)
```

Restituisce TRUE quando c'è una variazione positiva dell'indicatore rispetto alla barra precedente.

### ***High Minus Low***

```
HighMinusLow ()  
HML ()
```

Panoramica:

Questa funzione restituisce il prezzo HIGH meno il prezzo LOW per ogni barra.

Interpretazione:

Questo indicatore viene spesso utilizzato come componente per altri indicatori di analisi tecnica ma può anche essere utilizzato con una Moving Average per mostrare il cambiamento in atto sui prezzi nel tempo.

Parametri Standard:

Non previsti

Esempio:

```
SET A = SMA (HML (), 14)  
A > REF (A, 10)
```

Restituisce TRUE quando l'altezza media delle barre è aumentata rispetto alle barre precedenti.

### ***Moving Average Convergence / Divergence (MACD)***

```
MACD (@shortCycle, @longCycle, @signalPeriods, @maType)  
MACDSignal (@shortCycle, @longCycle, @signalPeriods, @maType)  
MACDS (@shortCycle, @longCycle, @signalPeriods, @maType)
```

```
MACDHistogram (@shortCycle, @longCycle, @signalPeriods, @maType)  
MACDH (@shortCycle, @longCycle, @signalPeriods, @maType)
```

Panoramica:

l'oscillatore MACD è costituito da un istogramma (MACDH) che misura la differenza tra due medie mobili (periodo 26 e 13) e da due medie mobili che sono, la 26 e una sua media mobile a 9 periodi.

Interpretazione/Utilizzo:

Segnali BUY o SELL vengono generati dal crossover (calcolato dalla funzione

MACDSignal) dei livelli di ipercomprato / ipervenduto del MACD e le divergenze tra il MACD e il prezzo dello strumento. E' un indicatore anche di momentum in quanto misura l'ampiezza dei movimenti ed è anche usabile come indicatore TREND following e di divergenza.

L'attraversamento verso il basso della linea dello zero indica debolezza di mercato mentre il superamento indica forza di mercato.

Parametri Standard:

Long Cycle: **26**  
Short Cycle: **13**  
Signal Periods: **9**  
MA Type: **SIMPLE**

Esempio:

```
SET A = MACDSignal (13, 26, 9, SIMPLE)  
SET B = MACD (13, 26, 9, SIMPLE)  
CROSSOVER(A, B) = TRUE
```

Restituisce TRUE se la linea MACDSignal ha attraversato, superandola, la linea MACD (segnale di TREND LONG).

### ***Momentum Oscillator***

```
MomentumOscillator (@vector, @periods)  
MO (@vector, @periods)
```

Panoramica:

Il Momentum Oscillator calcola la variazione di prezzo in un determinato periodo di tempo come un rapporto.

Interpretazione/Utilizzo:

Valori sempre più alti del Momentum Oscillator possono indicare che i prezzi sono in TREND fortemente rialzista. Il Momentum Oscillator è strettamente legato al MACD ed alla velocità del cambiamento prezzo (Price ROC).

Parametri Standard:

Vector: **CLOSE**  
Periods: **14**

Esempio:

```
MO (CLOSE, 14) > 90
```

Restituisce TRUE quando il Momentum Oscillator sul CLOSE è superiore a 90.

## Price Oscillator

**PriceOscillator (@vector, @shortTermPeriods, @longTermPeriods, @maType)**  
**PO (@vector, @shortTermPeriods, @longTermPeriods, @maType)**

Panoramica:

Simile al Volume Oscillator, il Price Oscillator è calcolato sulla base del differenziale tra due Moving Averages.

Interpretazione/Utilizzo:

Il Price Oscillator è fondamentalmente un differenziale tra Moving Averages. Il segnale BUY di solito si verifica quando l'oscillatore si alza, il segnale SELL di solito si verifica quando l'oscillatore scende.

Parametri Standard:

Vector: **CLOSE**

Short Term Periods: **9**

Long Term Periods: **14**

MA Type: **SIMPLE**

Esempio:

**PO (CLOSE, 9, 14, SIMPLE) > 0**

Restituisce TRUE se il Price Oscillator è in territorio positivo.

## Prime Number Oscillator

**PrimeNumberOscillator (@vector)**  
**PNO (@vector)**

Panoramica:

Trova il numero primo più vicino al prezzo, e calcola la differenza tra il numero primo ed il prezzo stesso.

Interpretazione/Utilizzo:

Questo indicatore può essere utilizzato per individuare punti di svolta del mercato. Quando l'oscillatore rimane nello stesso punto alto per due periodi consecutivi in campo positivo, è possibile prevedere un prossimo potenziale TREND in discesa. Viceversa, quando l'oscillatore rimane in un punto basso per due periodi consecutivi nel campo negativo, è possibile prevedere un prossimo potenziale TREND in salita.

*Nota: il suo funzionamento ricade nei "misteri" dei numeri primi tanto quanto l'apporto dei numeri di Fibonacci.*

Parametri Standard:

Vector: **CLOSE**

Esempio:

```
PNO (CLOSE) = REF (PNO (CLOSE), 1)  
AND REF (PNO (CLOSE), 2) <> PNO (CLOSE)
```

Restituisce TRUE quando le ultime 2 barre hanno lo stesso valore di PNO, ma la terzultima ha un valore diverso.

## ***Rainbow Oscillator***

```
RainbowOscillator (@vector, @levels, @maType)  
RBO (@vector, @levels, @maType)
```

Panoramica:

Il Rainbow Oscillator è calcolato in base a 6 periodi di una serie di 6 Moving Average. Si setta il valore più alto, nel caso in esempio 20 e, automaticamente, vengono calcolate le rimanenti 5 Moving Average con periodi che avranno i valori di, 17, 14, 11, 8 ed infine 5.

Interpretazione/Utilizzo:

Il TREND viene individuato dal passaggio del valore dell'indicatore da sotto lo zero (TREND SHORT) a sopra lo zero (TREND LONG). Per evitare/diminuire i falsi segnali c'è chi aspetta la conferma di almeno due barre nello status sopra/sotto lo zero, oppure avendo una ulteriore conferma dalla lunghezza dell'istogramma che, se di dimensioni minime, potrebbe indicare unicamente un rintracciamento.

Parametri Standard:

Vector: **CLOSE**

Levels: **21**

MA Type: **SIMPLE**

Esempio:

```
SET R = RBO (CLOSE, 21, SIMPLE)  
R > 0 AND REF (R, 1) < 0  
# alternativa all'utilizzo di CROSSOVER; notare l'utilizzo di SET e di REF
```

Restituisce TRUE quando il Rainbow Oscillator passa da sotto lo zero a sopra lo zero.



## **Sine Wave**

**SineWave(@vector, @periods)**  
**SW(@vector, @periods)**  
**LeadSineWave(@vector, @periods)**  
**LSW(@vector, @periods)**

Panoramica:

Il Sinewave è un'indicatore recente, ideato da John Ehlers nel 1996 viene utilizzato per evidenziare se il titolo sia in fase di trend o in un ciclo. Il Sine Wave è composto da due grafici il Sine Wave ed il Lead Sine Wave, il primo misura la fase, il secondo la fase incrementata di 45° gradi. Insieme le curve danno indicazioni sullo stato

Interpretazione/Utilizzo:

Un segnale di acquisto si ha quando la Sine Wave incrocia in alto la Lead Sine Wave, un segnale di vendita quando la Sine Wave incrocia in basso la Lead Sine Wave.

Parametri Standard:

Vector: **CLOSE**

Periods: **14**

Esempio:

**CROSSOVER (SW(CLOSE, 14), LSW(CLOSE, 14))**

Restituisce TRUE quando la Sine Wave incrocia in alto la Lead Sine Wave.

## **Stochastic Oscillator**

**SOPK (@%K Periods, @%KSlowPeriods, @%DPeriods, @maType)**  
**SOPD (@%K Periods, @%KSlowPeriods, @%DPeriods, @maType)**

Panoramica:

Lo Stochastic Oscillator è un popolare indicatore che misura il rapporto tra la chiusura e la differenza tra il massimo e minimo del periodo considerato. In questo caso la formula del calcolo semplificherà la comprensione:

$%K = 100 * [(CHIUSURA - MINn) / (MAXn - MINn)]$

$%D = 3 \text{ period Simple Moving Average di } \%K$

Interpretazione/Utilizzo:

Lo Stochastic Oscillator ha due componenti: %K (la funzione SOPK) e %D (la funzione SOPD). %K è spesso visualizzata su un grafico azionario come una linea continua e %D è spesso indicata da una linea tratteggiata.

Un metodo di utilizzo dello Stochastic Oscillator è quello di acquistare quando una componente (%K oppure %D) scende dal valore 80 e vendere quando uno dei componenti sale dal valore 20.

80 è la soglia che indica ipercomprato e 20 quella che indica ipervenduto. Un altro modo di interpretare lo Stochastic Oscillator è quello di acquistare quando %K sale sopra il %D, e viceversa, vendere quando %K scende al di sotto %D.

Naturalmente è validissimo per le divergenze che si verificheranno mentre l'indicatore è nelle zone critiche, oltre 80 o sotto il 20.

Parametri Funzione		Parametri Indicatore beeTrader	
%K Periods	<b>9</b>	%K Periods	<b>9</b>
%K SlowPeriods	<b>3</b>	%K SlowPeriods	<b>3</b>
%D Periods	<b>9</b>	%D Periods	<b>9</b>
MA Type	<b>SIMPLE</b>	MA Type	<b>SIMPLE</b>
		highMark	<b>80</b>
		lowMark	<b>20</b>

Esempio:

**SOPK (9, 3, 9, SIMPLE) > 80 OR SOPD (9, 3, 9, SIMPLE) > 80**

Restituisce TRUE quando lo Stochastic Oscillator è in territorio di ipervenduto, cioè al di sopra della linea 80 che è l'highMark.

## ***Trix Oscillator***

**TRIX (@vector, @periods)**

Panoramica:

Il TRIX Oscillator è un Oscillatore di Momentum disegnato come istogramma. Compara la differenza tra due medie mobili, ciascuna tripla esponenziale, e la disegna come un oscillatore.

Interpretazione/Utilizzo:

L'utilizzo più comune del TRIX Oscillator è quello di entrare a mercato quando l'oscillatore attraversa la linea dello zero, se in salita sarà un segnale BUY, viceversa sarà un segnale SELL.

Viene anche usato per ricercare le divergenze fornendo, in questo caso di utilizzo, segnali BUY con divergenze bullish e segnali SELL con divergenze bearish.

Anche il cambiamento della sua pendenza, ovvero una barra dell'istogramma minore/maggiore della precedente, indica un cedimento/rafforzamento del TREND in atto.

Parametri Standard:

Vector: **CLOSE**

Periods: **9**

Esempio:

**TRIX > REF (TRIX,1) OR TRIX < REF (TRIX,1)**

Restituisce TRUE quando TRIX ha una variazione di pendenza, sia negativa che positiva.

### ***True Range***

**TrueRange ()**  
**TR ()**

Panoramica:

Il True Range misura il range di movimento del prezzo che si è avuto nella barra (in base al timeframe) di contrattazione come differenza tra il prezzo più alto e quello più basso, ovvero (H-L). E' pertanto un indicatore di volatilità molto immediato che, per sopperire alle misurazioni che si hanno nei giorni in cui si hanno ampie escursioni in una sola direzione (days inside), viene considerato SOLO come il dato maggiore tra le seguenti 3 misurazioni:

- differenza tra HIGH e LOW della barra;
- differenza tra CLOSE della barra precedente e LOW della barra attuale;
- differenza tra CLOSE della barra precedente e HIGH della barra attuale.

Parametri Standard:

Non previsti

Esempio:

**TR () > 1.95**

Restituisce TRUE quando il True Range è maggiore di 1,95.

### ***Vertical Horizontal Filter***

**VerticalHorizontalFilter (@vector, @periods)**  
**VHF (@vector, @periods)**

Panoramica:

Il Vertical Horizontal Filter (VHF) può essere utilizzato in due modi: come indicatore di TREND guardando la sua inclinazione (Long se Positiva, Short se Negativa) e come indicatore di Momentum, al pari dell'ADX, per valutare la forza del movimento in atto.

A tale scopo viene disegnato con due bande settabili dall'utente per definire, su di un campione storico, i due livelli che possano identificare la fase laterale e la fase di TREND.

Interpretazione/Utilizzo:

Settati i due livelli che identifichino TREND o non TREND o utilizzando quelli standard che sono 0,2 e 0,3, si può interpretare una salita oltre il valore di 0,2 come l'inizio di un TREND che si rafforzerà al superamento del valore 0,3.

Parimenti, una discesa e il conseguente attraversamento del valore 0,3 indicherà l'indebolimento del TREND e l'inizio di una fase laterale del mercato.

Parametri Funzione

Vector **CLOSE**

Periods **21**

Parametri Indicatore beeTrader

Vector **CLOSE**

Periods **21**

lowMark **0,2**

highMark **0,3**

Esempio:

**VHF (CLOSE, 21) < 0.2**

Restituisce TRUE quando VHF ha un valore inferiore a 0,2.

### **Volume Oscillator**

**VolumeOscillator (@shortTermPeriods, @longTermPeriods, @maType, @PointsOrPercent)**

**VO (@shortTermPeriods, @longTermPeriods, @maType, @PointsOrPercent)**

Panoramica:

Il Volume Oscillator mostra un differenziale tra due Moving Averages di volume differenti in un determinato periodo di tempo.

Interpretazione/Utilizzo:

Offre una visione chiara dell'esistenza o meno di volumi in aumento o in diminuzione.

Parametri Standard:

Short Term Periods: **9**

Long Term Periods: **21**

MA Type: **SIMPLE**

Points or Percent: **PERCENT**

Esempio:

**VO (9, 21, SIMPLE, PERCENT) > 0**

## **Williams %R**

**WilliamsPctR (@periods)**

**WPR (@periods)**

Panoramica:

Sviluppato dal trader Larry Williams, il Williams %R è un indicatore che misura i livelli di ipercomprato / ipervenduto. Questo indicatore è simile allo Stochastic Oscillator. Attenzione che i valori calcolati da questo indicatore possono assumere valori compresi tra 0 a -100, quindi sono valori negativi e ci si può confondere confrontandolo con altri oscillatori come, appunto, lo Stocastico.

Interpretazione/Utilizzo:

Il mercato è considerato in ipercomprato quando la %R è in un intervallo da 0 a -20; è al contrario è in ipervenduto quando la %R è in un intervallo da -80 a -100, l'uscita del %R dalle due zone rappresenta pertanto l'inizio di un nuovo TREND. SHORT se esce da -20 e LONG se esce da -80.

Parametri Funzione

Periods           **14**

Parametri Indicatore beeTrader

Periods           **14**

lowMark           **-80**

highMark           **-20**

Esempio:

**WPR (14) > -80 AND REF (WPR (14), 1) <= - 80**

Restituisce TRUE quando il Williams %R è maggiore di -80 e 1 barra prima era minore/uguale a -80. (esce dall'ipervenduto segnalando un LONG)

## **Williams Accumulation Distribution**

**WilliamsAccumulationDistribution ()**

**WAD ()**

Panoramica:

Anche questo indicatore, sviluppato da Larry Williams, mostra la relazione tra prezzo e volume per identificare le condizioni di mercato dominate da compratori (Accumulation) o da venditori (Distribution).

Interpretazione/Utilizzo:

Quando il Williams Accumulation Distribution si muove al rialzo, indica che si sta assistendo ad una fase di accumulo, quindi la maggior parte del volume è associato al movimento del prezzo diretto verso l'alto. Quando l'indicatore si muove invece al ribasso, esso mostra che si sta assistendo ad una fase di distribuzione, ed in questo caso la maggior parte del volume è associato al

movimento del prezzo in fase discendente.

Parametri Standard:  
Non previsti

Esempio:

**WAD () < 1**

Restituisce TRUE quando WAD è inferiore a 1.

## Funzioni Cromatiche

Questa sezione illustra le funzioni che possono essere utilizzate per colorare gli indicatori, sia da applicare su Chart che su Watchlist.

### **COLORIZE**

**COLORIZE (@vector, @minValue, @max Value, @colorOrder, @centerPoint)**

Panoramica:

La funzione COLORIZE può fornire un'utile indicazione dell'ampiezza di un movimento del proprio Indicator

Interpretazione/Utilizzo:

La funzione COLORIZE serve per associare ad un PLOT un colore calcolato sulla scala tra il rosso ed il verde a seconda del valore del PLOT stesso.

Parametri obbligatori:

@vector: vettore che contiene i valori a cui associare i colori

@minValue: valore minimo da usare nel calcolo della scala dei colori

@maxValue: valore massimo da usare nel calcolo della scale dei colori

Nota: se i valori contenuti all'interno del vettore @vector eccedono i limiti impostati in @minValue e @maxValue, la funzione COLORIZE utilizzerà comunque un valore compreso tra @minValue e @maxValue.

Parametri opzionali:

@colorOrder: indica in quale ordine deve essere calcolata la scala dei colori.

Può assumere i valori COLORIZE\_ASCENDING oppure COLORIZE\_DESCENDING.

@centerPoint: se specificato e diverso da NAN indica che i valori di @vector verranno interpretati come differenza in valore assoluto rispetto a questo valore.

Parametri Standard:

@colorOrder: **COLORIZE\_ASCENDING**

@centerPoint: **NAN**

Esempio:

**SET PLOTCOLOR1 = COLORIZE(PLOT1, -3.0, 3.0)**

Imposta il colore del primo parametro di uscita dell'indicatore con una scala di colore dal rosso al verde, a seconda del valore stesso dell'indicatore.

## **RGB (solo funzione)**

**RGB (@red, @green, @blue)**

Panoramica:

La funzione RGB si utilizza per definire il gradiente del colore da applicare ai parametri di uscita degli indicatori (PLOTCOLORx) nel caso in cui il colore desiderato non sia presente tra quelli predefiniti.

Interpretazione/Utilizzo:

La funzione RGB serve per esprimere un colore in termini numerici. I parametri di ingresso della funzione devono essere numeri interi compresi tra 0 e 255. Su internet si trovano numerosi traduttori o tabelle per trovare l'insieme dei tre parametri necessari per riprodurre il colore desiderato.

Parametri Standard:

Non previsti

Esempio:

**SET PLOTCOLOR1 = RGB (255, 255, 153)**

Imposta il colore del primo parametro di uscita dell'indicatore con la tonalità del colore di sfondo dell'esempio di cui sopra (giallo chiaro).



## Altre Funzioni

Questa sezione illustra gli indicatori di analisi tecnica che non rientrano in nessuno dei precedenti gruppi.

### *Accumulative Swing Index*

**AccumulativeSwingIndex (@limitMoveValue)**  
**ASI (@limitMoveValue)**

Panoramica:

L' Accumulative Swing Index è la somma degli Swing Index in un determinato periodo di tempo. La sua particolarità rimane quindi il confronto dei parametri OHLC di ogni singola barra.

Interpretazione/Utilizzo:

Avendo nel calcolo una serie di Swing Index ecco che diventa ottimo nel trovare divergenze o per conferma del TREND in atto. In pratica la linea dell'indicatore, in fasi di TREND segue la linea dei prezzi e possibili discrepanze daranno segnali di allerta.

Parametri Standard:

Limit Move Value: **1**

Esempio:

**TREND (ASI (1), 30) > UP**

Restituisce TRUE quando l' Accumulative Swing Index è in TREND rialzista su base 30 periodi.

Oppure:

**TREND (ASI (1), 30) = UP AND TREND (CLOSE, 30) = DOWN**

Restituisce TRUE quando il TREND dell'indicatore Accumulative Index ed il TREND del sottostante sono in divergenza. La divergenza è un forte segnale anticipatore di cambio TREND.

### *Center of Gravity*

**CenterOfGravity (@vector, @periods)**  
**COG (@vector, @periods)**

Panoramica:

Questo indicatore permette di avere immediatamente un punto di riferimento sulla direzione presa dal mercato e si basa sull'assunto che "prezzi fatti si

rivedono". In pratica l'autore sostiene che quando i prezzi si scostano di molto dal valore di un periodo precedente, essi tendano a ritornare attorno a quel valore. E' un indicatore di TREND costruito con leggi statistiche, regressioni e deviazioni standard.

Interpretazione/Utilizzo:

Semplicemente sarà l'inclinazione della linea dell'indicatore a dare la direzione del TREND, aiutati, per evitare falsi segnali, dalla Signal Line.

Parametri Funzione

Vector **CLOSE**

Periods **21**

Parametri Indicatore beeTrader

Vector **CLOSE**

Periods **21**

Signal Periods **3**

Esempio:

```
SET C = CenterOfGravity (CLOSE, 21)
```

```
SET SignalLine = SMA (C, 3)
```

```
C > REF (C, 1) AND C > SignalLine
```

Restituisce TRUE quando l'indicatore è in salita ed al di sopra della sua Signal Line, indicando un TREND LONG. (Notare l'utilizzo di SET per il calcolo del COG).

### ***Chande Forecast Oscillator***

```
ChandeForecastOscillator (@vector, @periods)
```

```
CFO (@vector, @periods)
```

Panoramica:

Indicatore statistico basato sulla regressione lineare disegna la differenza percentuale tra il prezzo di una regressione a n periodi ed il prezzo attuale. Quindi sfruttando la proprietà della regressione tende ad anticipare il segnale. Viene costruito con una Signal Line che è in genere una Simple Moving Average a 3 periodi del CFO.

Interpretazione/Utilizzo:

Quando l'indicatore è superiore alla sua Simple Moving Average a 3 periodi indica una fase di salita, viceversa quando l'indicatore è minore della sua Simple Moving Average, indica una fase di TREND al ribasso.

Parametri Funzione

Vector **CLOSE**

Periods **21**

Parametri Indicatore beeTrader

Vector **CLOSE**

Periods **21**

Esempio:

**SET C = CFO (CLOSE, 21)**  
**SET AverageCFO = SMA (C, 3)**  
**C > AverageCFO**

Restituisce TRUE quando il CFO è maggiore della sua Simple Moving Average a 3 periodi. (Notare l'utilizzo di SET per il calcolo del CFO e della sua Simple Moving Average).

### ***Chaikin Money Flow***

**ChaikinMoneyFlow (@periods)**  
**CMF (@periods)**

Panoramica

L'oscillatore Chaikin Money Flow è un indicatore di momentum che indica segnali BUY e SELL calcolando insieme prezzo e volume.

Inoltre, basandosi sulla premessa che se un titolo chiude sopra del suo punto medio,  $(HIGH + LOW) / 2$ , sarà un giorno di accumulo, e se chiude sotto il suo punto medio, sarà un giorno di distribuzione, individua e segnala le fasi di Accumulazione e Distribuzione che sono le fasi che determinano salite e discese dei prezzi.

E' rappresentato come un istogramma che può essere sopra o sotto la linea dello zero.

Interpretazione/Utilizzo:

Un segnale BUY viene generato quando l'indicatore è in aumento ed è in territorio positivo.

Un segnale SELL viene generato quando l'indicatore è in calo ed è in territorio negativo.

Oppure si hanno segnali BUY/SELL all'attraversamento della linea dello zero rispettivamente dal basso verso l'alto oppure dall'alto verso il basso.

Parametri Standard:

Periods: **21**

Esempio:

**CMF (21) > REF (CMF (21), 1)**

Restituisce TRUE quando il Chaikin Money Flow è rialzista. Il valore di oggi è maggiore del valore di ieri.

## Commodity Channel Index

**CommodityChannelIndex (@periods, @maType)**  
**CCI (@periods, @maType)**

Panoramica:

Donald Lambert ha sviluppato l'indicatore CCI. Anche se lo scopo di questo indicatore è quello di individuare turni ciclici delle materie prime, è spesso utilizzato per i titoli. E' un oscillatore che si sviluppa tra i valori attorno al 100 e -100 nel 70/80% dei casi. Questo dipende dalla costruzione che comprende una costante. Più il periodo sarà corto e più aumenteranno i valori maggiori di 100. Il CCI ha anche una media mobile del CCI medesimo che può essere usata sia per smussare i segnali sia per avere indicazioni di TREND.

Interpretazione/Utilizzo:

Come molti altri oscillatori, il CCI fornisce indicazioni di divergenze; queste sono tanto più forti quanto più si verificano in posizione di eccesso dell'oscillatore.

Questo significa che se viene individuata una divergenza ribassista con valori dell'oscillatore superiori a +100 si tratta di un segnale molto forte, ed un'apertura di posizione ribassista una volta tornati al di sotto del valore +100 è considerata da moltissimi analisti come un'ottima opportunità.

Valori di CCI superiori alla sua media indicano TREND in salita e valori inferiori alla sua media indicano TREND in discesa.

E' usato anche per individuare fasi di ipercomprato ed ipervenduto con grande efficacia.

*Nota: il settaggio del parametro Periods si dovrebbe attuare nel seguente modo: si individua il ciclo del TREND in atto utilizzando i vari indicatori di ciclo (Fibonacci, Tirone, ecc ), una volta individuato il numero di barre del ciclo si divide il numero delle barre per 3. Se il ciclo individuato è di 45, il CCI va settato a  $45:3=15$*

Parametri Funzione

Periods           **21**  
MA Type           **SIMPLE**

Parametri Indicatore beeTrader

Periods           **21**  
MA Type           **SIMPLE**  
Signal Periods   **10**  
lowMark           **-80**  
highMark          **80**

Esempio:

**CCI (21, SIMPLE) > 80 AND REF (CCI (21, SIMPLE), 1) < 80**

Restituisce TRUE appena il CCI si è trasferito in territorio di ipercomprato.

## **Comparative RSI**

**ComparativeRelativeStrenghtIndex (@vector1, @vector2)**  
**CRSI (@vector1, @vector2)**

Panoramica:

Questo indicatore compara l'RSI a 14 periodi di un vettore rispetto ad un altro. Viene rappresentato comparando il vettore CLOSE ed il vettore VOLUME ma in ogni caso l'utente potrà decidere a suo piacere.

Interpretazione/Utilizzo:

Dato che comparando l'RSI di due vettori si vuole verificare se e quale vettore possa avere delle divergenze rispetto all'altro, l'interpretazione è la divergenza dell'RSI. Troveremo pertanto valori in salita del CRSI se il primo vettore sarà in crescita o il secondo in diminuzione.

Viene utilizzato il VOLUME perchè è il parametro che sostiene il prezzo e quindi il TREND. Un CRSI in aumento suggerisce di rimanere in posizione.

Parametri Standard:

Vector1: **CLOSE**

Vector2: **VOLUME**

Esempio:

**TREND (CRSI (CLOSE, VOLUME), 14) = UP**

Restituisce TRUE quando il vettore 1 è in aumento o il vettore 2 è in diminuzione.

## **Coppock Curve**

**CoppockCurve (@vector)**  
**CPKC (@vector)**

Panoramica:

Indicatore di Momentum da utilizzare su grafico a timeframe settimanale! E' calcolato semplicemente come Moving Average a 10 periodi della somma del ROC a 14 con il ROC a 11 mesi. E rappresentato come istogramma il cui valore oscilla sopra e sotto la linea dello zero.

Interpretazione/Utilizzo:

Valori positivi indicano TREND LONG e valori negativi indicano TREND SHORT. La dimensione degli istogrammi identificano la forza del TREND in atto.

Parametri Standard:

Vector: **CLOSE**

Esempio:

**SET COP = CPKC (CLOSE)**  
**COP > 0 AND COP > REF (COP, 1)**

Restituisce TRUE se il Coppock Curve è positivo e rialzista. E' maggiore di zero ed il valore di oggi è maggiore del valore di ieri.

## ***Correlation Analysis***

**CorrelationAnalysis (@vector1, @vector2)**  
**CA (@vector1, @vector2)**

Panoramica:

Correlation Analysis viene utilizzata per determinare la correlazione tra due vettori.

Interpretazione/Utilizzo:

La funzione restituisce un valore che indica la correlazione tra due vettori. I vettori possono contenere prezzi, i valori di altri indicatori, o altri valori.

Parametri Standard:

Vector1: **[Any Vector]**

Vector2: **[Any Vector]**

Esempio:

**CA (CLOSE, SMA (CLOSE, 14)) > 0.99**

Restituisce TRUE quando il movimento del CLOSE è altamente correlato con il movimento della SMA a 14 periodi.

## ***Ehler Fisher Transform***

**EhlerFisherTransform (@periods)**  
**EFT (@periods)**

**EhlerFisherTrigger (@periods)**  
**EFTT (@periods)**

Panoramica:

L'indicatore Ehler Fisher Transform, ideato da John Ehler, è di recente ideazione, infatti fu pubblicato per la prima volta nel 2002. Con i punti di svolta chiari e una risposta veloce, l'Ehler Fisher Transform viene utilizzato per identificare inversioni di TREND e si basa sul presupposto che i prezzi si comportano come un'onda quadra e non seguono una distribuzione gaussiana o normale.

Interpretazione/Utilizzo:

Segnali di BUY e SELL vengono generati il Fisher incrocia sopra o sotto la linea

di trigger, che è Fisher di una barra precedente.

Quindi un segnale BUY viene generato quando la linea Fisher incrocia al rialzo la soglia -1 e passa sopra la linea di trigger.

Un segnale SELL è generato quando la linea Fisher incrocia al ribasso la soglia 1 e attraversa sotto della linea di trigger.

Parametri Standard:

Periods: **21**

Esempio:

```
SET T = EFTT (21)  
SET AverageEFTT = SMA (T, 3)  
T > AverageEFTT
```

Restituisce TRUE quando il EFTT è maggiore della sua Simple Moving Average a 3 periodi. (Notare l'utilizzo di SET per il calcolo del EFTT e della sua Simple Moving Average).

## ***Elder Force Index***

**ElderForceIndex (@periods)**

**EFI (@periods)**

Panoramica:

L'indicatore di analisi tecnica Force index, sviluppato da Alexander Elder, calcola la forza dei cambiamenti dei prezzi usando una combinazione di prezzi e del volume. L'indicatore Elder Force Index plotta una linea che oscilla sopra e sotto lo zero. Questo indicatore ha delle caratteristiche particolari poiché utilizza, i cambiamenti dei prezzi, la direzione del cambiamento del prezzo e il volume. L'Elder Force Index è una misura della prevalenza dei compratori o dei venditori sul mercato.

Calcolo: l'Elder Force Index (EFI) è calcolato usando l'ultimo CLOSE, il precedente CLOSE e gli ultimi volumi. Quindi se il CLOSE corrente è superiore al CLOSE precedente, l'Elder Force Index è positivo, e se il CLOSE corrente è inferiore al precedente, l'Elder Force Index è negativo, inoltre maggiori sono i volumi di negoziazione maggiore sarà la sua forza.

Interpretazione/Utilizzo:

In generale, se il prezzo è in rialzo, e il l'Elder Force Index è in salita, il movimento al rialzo è probabile che continui. Invece, se i prezzi sono in TREND al rialzo, ma l'Elder Force Index è vicino allo zero o si muove verso lo zero, si sta creando un'eguaglianza tra compratori e venditori, in tal modo il movimento al rialzo dei prezzi è probabile che entri in una fase di stallo o si potrebbero fermare. Al contrario per TREND in discesa.

Parametri Standard:

Periods: **13**

Esempio:

**EFI (13) > REF (EFI (13), 1)**

Restituisce TRUE quando EFI è in salita rispetto al valore precedente.

### ***Elder Ray Bull Power***

**ElderRayBullPower (@periods, @maType)**

**ERBull (@periods, @maType)**

Panoramica:

Elder Ray Bull Power viene costruito con simili principi di Bear Power, ma con la differenza che Elder Ray Bull Power mette in evidenza i tori nel mercato, i movimenti rialzisti, o meglio nella battaglia giornaliera tra tori e orsi, indica quando vincono i tori misurandone l'andamento. Lo scopo di questo indicatore è misurare la pressione dei TREND rialzisti in un determinato periodo di tempo. L'indicatore di analisi tecnica Elder Ray Bull Power aiuta ad individuare i possibili cambiamenti del TREND.

Interpretazione/Utilizzo:

In un TREND al rialzo Elder Ray Bull Power si colloca sotto lo zero. Questo indicatore è utilizzato per aprire posizioni al rialzo, e inoltre fornisce anche i punti di uscita. Si possono aprire posizioni al rialzo quando l'indicatore oltrepassa la linea dello zero.

Parametri Standard:

Periods: **13**

MA Type: **EXPONENTIAL**

Esempio:

**ERBull (13, EXPONENTIAL) > 0 AND REF (ERBull (13, EXPONENTIAL), 1) < 0**

Restituisce TRUE quando ERBull supera la linea dello zero (oggi è sopra lo zero ieri era sotto lo zero).

### ***Elder Ray Bear Power***

**ElderRayBearPower (@periods, @maType)**

**ERBear (@periods, @maType)**

Panoramica:

Elder Ray Bear Power viene costruito con simili principi di Bull Power, ma con la differenza che Elder Ray Bear Power mette in evidenza i tori nel mercato, i movimenti rialzisti, o meglio nella battaglia giornaliera tra tori e orsi, indica quando vincono i tori misurandone l'andamento. Lo scopo di questo indicatore



è misurare la pressione dei TREND rialzisti in un determinato periodo di tempo. L'indicatore di analisi tecnica Elder Ray Bear Power aiuta ad individuare i possibili cambiamenti del TREND.

Interpretazione/Utilizzo:

In un TREND al ribasso Elder Ray Bear Power si colloca sopra lo zero. Questo indicatore è utilizzato per aprire posizioni al ribasso, e inoltre fornisce anche i punti di uscita. Si possono aprire posizioni al ribasso quando l'indicatore scende al di sotto della linea dello zero.

Parametri Standard:

Periods: **13**

MA Type: **EXPONENTIAL**

Esempio:

**ERBear (13, EXPONENTIAL) < 0 AND REF (ERBear (13, EXPONENTIAL), 1) > 0**

Restituisce TRUE quando ERBear scende al di sotto della linea dello zero (oggi è sotto lo zero ieri era sopra lo zero).

### ***Elder Thermometer Index***

**ElderThermometerIndex (@periods, @maType)**

**ETHERM (@periods, @maType)**

**ElderThermometerAverage (@periods, @maType)**

**ETHERMAVG (@periods, @maType)**

Panoramica:

E' un indicatore di volatilità che viene calcolato sulla escursione H/L della giornata (ETHERM). Il segnale di allerta è dato dalla media mobile dei valori (ETHERMAVG) di ogni singola escursione di barra per il periodo impostato. Elder suggerisce 22 che corrisponde al numero dei giorni di borsa mensili.

Interpretazione/Utilizzo:

L'utilizzo è semplice in quanto è sufficiente guardare gli istogrammi dell'indicatore: se sono sopra il valore della linea della Moving Average allora è un segnale di inizio TREND se sono sotto la linea della Moving Average significa continuità di TREND o fase laterale.

Parametri Standard:

Periods: **22**

Ma Type: **SIMPLE**

Esempio:

**ElderThermometerIndex (22, SIMPLE) > ElderThermometerAverage (22, SIMPLE)**

Restituisce TRUE quando Elder Thermometer Index supera il valore calcolato dalla Simple Moving Average (fase di alta volatilità).

## ***Gann Swing***

**GannSwing (@seriesHigh, @seriesLow)**

**GS (@seriesHigh, @seriesLow)**

Panoramica:

E' un indicatore che mira ad individuare gli Swing del mercato, ovvero le inversioni di trend.

Interpretazione/Utilizzo:

Se l'indicatore ha un andamento in salita identifica un TREND LONG, al contrario identifica un TREND SHORT.

Parametri Standard:

seriesHigh: **HIGH**

seriesLow: **LOW**

Esempio:

```
SET mySwing = GannSwing (HIGH, LOW)
```

```
mySwing < REF (mySwing, 1)
```

Restituisce TRUE quando l'indicatore è in discesa (valore di oggi maggiore di quello di ieri).

## ***Gaussian Filter***

**GaussianFilter (@periods, @periods, @poles)**

**GAUSS (@periods, @periods, @poles)**

Panoramica:

E' un filtro passa basso che tende ad eliminare il ritardo classico degli indicatori preservando la precisione nel calcolo del TREND. Questo indicatore è simile nell'utilizzo, alla distribuzione Gaussiana. Il parametro "poli" viene utilizzato per regolare il ritardo nella risposta.

Interpretazione/Utilizzo:

Si usa come una Moving Average, se il CLOSE è superiore alla linea del filtro il TREND sarà LONG e viceversa nel caso in cui il CLOSE sia inferiore al valore del filtro.

Parametri Standard:

Vector: **CLOSE**

Periods: **20**

Poles: **1**

Esempio:

**SET GF = GaussianFilter (CLOSE, 20, 1)**  
**GF > REF (GF, 1)**

Restituisce TRUE quando l'indicatore è in salita (valore di oggi maggiore di quello di ieri).

### ***Gopalakrishnan Range Index***

**GopalakrishnanRangeIndex (@periods)**  
**GRI (@periods)**

Panoramica:

Indicatore di volatilità e di momentum esprime la variabilità del titolo in esame. Calcola il logaritmo naturale di HighestHigh e divide il risultato per il logaritmo naturale del LowestLow di uno stesso periodo temporale. Il risultato ottenuto viene a sua volta diviso per il logaritmo naturale del periodo scelto.

*Nota: beeTrader® restituisce il valore in valore assoluto.*

Interpretazione/Utilizzo:

Aumento di volatilità corrisponde ad espansione di range sia in salita che in discesa. Da utilizzare affiancato ad altri indicatori di TREND.

Parametri Standard:

Periods: **10**

Esempio:

**GRI (10) > 0,5**

Restituisce TRUE quando il GRI è maggiore di 0,5.

### ***Highest High Value***

**HighestHighValue (@periods)**  
**HHV (@periods)**

Panoramica:

Restituisce il valore più alto dei prezzi HIGH di un numero specifico di periodi.

Interpretazione/Utilizzo:

Utilizzato come componente di calcolo da molti altri indicatori di analisi tecnica.

Parametri Standard:

Periods: **21**

Esempio:

**HIGH = HHV (21)**

Restituisce TRUE quando il prezzo HIGH è il prezzo HIGH più alto degli ultimi 21 periodi.

### ***Historical Volatility Index***

**HistoricalVolatilityIndex (@vector, @periods, @barHistory, @standardDeviations)**  
**HistoricalVolatility (@vector, @periods, @barHistory, @standardDeviations)**  
**HVI (@vector, @periods, @barHistory, @standardDeviations)**

Panoramica:

Semplicemente la volatilità rappresenta il grado di variazione dei prezzi in un determinato periodo temporale. Questa viene misurata con metodi statistici e, in questo indice, tramite la deviazione standard espressa come percentuale annualizzata.

La formula per un indice di volatilità storica a 30 periodi è:

$\text{Stdev}(\text{Log}(\text{Close} / \text{Close Yesterday}), 30) * \text{Sqrt}(365)$

Alcuni traders utilizzano 252 giorni (giorni di mercato effettivi) anziché 365 come parametro della radice quadrata. La base del logaritmo è quella naturale.

Interpretazione/Utilizzo:

Valori elevati di HVI indicano che lo strumento è volatile, mentre valori bassi di HVI indicano che lo strumento è poco volatile o in TREND.

Parametri Standard:

Vector: **CLOSE**  
Periods: **30 o 60 o 90**  
Bar History: **365**  
Standard Deviations: **2**

Esempio:

**HVI (CLOSE, 30, 365, 2) < 0.01**

Restituisce TRUE quando l'indicatore HVI è maggiore di 0,01.

### ***Intraday Momentum Index***

**IntradayMomentumIndex ()**  
**IMI ()**

Panoramica:

L'Intraday Momentum Indicator, ideato da Tushar Chande, ha una costruzione

del tutto simile all'RSI di Welles Wilder: da esso si differenzia per il fatto che prende in considerazione non la differenza fra il CLOSE odierno e il CLOSE del giorno precedente (interday momentum) bensì la differenza fra il CLOSE e l'OPEN di oggi (intraday momentum), ovvero, in sostanza, la lunghezza e il colore della candela che identifica la seduta. Per il resto, RSI e IMI sono fondamentalmente identici. Visto che l'intraday momentum inverte la sua rotta prima dell'interday momentum, ciò sarà visibile dal colore e dalla forma delle candele, ma ancor prima dal comportamento dell'Intraday Momentum Index, che tenderà ad appiattirsi prima, e poi ad allontanarsi dalla zona di eccesso, per approssimarsi alla zona centrale.

Interpretazione/Utilizzo:

Un mercato - o un titolo - che, reduce da un ribasso, dà inizio ad una formazione di inversione che precede il successivo rialzo, vedrà il suo Intraday Mmomentum Index passare da valori bassi e inferiori a 30, a valori che tenderanno man mano verso 70, per il succedersi delle prime candele verdi alle candele rosse; e viceversa, un mercato in fase di topping inizierà a presentare, prima timidamente, poi sempre più decisamente, le candele rosse: in questo caso avremo un Intraday Momentum Index che, raggiunta la fascia superiore di oscillazione, si curverà verso il basso, dirigendosi verso 30.

*Nota: I valori 30 e 70 debbono essere settati e variati dall'utente in dipendenza al titolo in esame.*

Parametri Funzione

Parametri Indicatore beeTrader

lowMark **48**

highMark **52**

Esempio:

**IMI() > 52**

Restituisce TRUE quando IMI ha un valore superiore a 52.

### ***Klinger Volume Oscillator***

**KlingerVolumeOscillator (@shortCycle, @songCycle, @signalPeriods, @maType)**

**KVO (@shortCycle, @longCycle, @signalPeriods, @maType)**

Panoramica:

Il Klinger Volume Oscillator è stato creato da Stephen J. Klinger con lo scopo di analizzare il rapporto tra prezzo e volume nel breve, ma anche nel medio periodo. Il metodo di calcolo si basa su tre diverse rilevazioni: il range di prezzi, il volume e l'accumulazione o la distribuzione nel periodo considerato. Il ragionamento alla base della sua creazione sta principalmente nel fatto che il volume, insieme ai prezzi, causa una maggiore pressione della domanda oppure dell'offerta, motivo per cui occorre dare importanza ai prezzi ma anche alla forza del volume che causano la salita o la discesa delle quotazioni. Nel

calcolo standard per la creazione del Klinger Volume Oscillator si utilizza la differenza tra una "Exponential Moving Average" a 34 periodi e una a 55 periodi con una trigger line a 13 periodi.

Interpretazione/Utilizzo:

Essendo così strettamente legato all'andamento dei prezzi/volume con particolare riferimento alla loro correlazione è abbastanza logico e necessario che l'indicatore venga sfruttato prevalentemente per individuare divergenze che possano evidenziare eccessi di prezzo al rialzo o al ribasso, mostrando quindi nuovi massimi o minimi che non vengono accompagnati da una forza corrispondente sui volumi.

Parametri Standard:

Short Cycle: **8**

Long Cycle: **12**

Signal Periods: **14**

MA Type: **SIMPLE**

Esempio:

```
SET K = KVO (8, 12, 14, SIMPLE)  
K > REF (K, 1) AND TREND (CLOSE, 30) = DOWN
```

Restituisce TRUE quando il KVO è in salita ed il TREND è in discesa, individuando quindi una divergenza.

### ***Lowest Low Value***

**LowestLowValue (@periods)**

**LLV (@periods)**

Panoramica:

Restituisce il valore più basso dei prezzi LOW di un numero specifico di periodi.

Interpretazione/Utilizzo:

Utilizzato come componente di calcolo da molti altri indicatori di analisi tecnica.

Parametri Standard:

Periods: **21**

Esempio:

```
LOW = LLV (21)
```

Restituisce TRUE quando il prezzo LOW è il prezzo LOW più basso degli ultimi 21 periodi.

## **Market Facilitation Index**

**MarketFacilitationIndex ()**  
**MKTFI ()**

Panoramica:

L'indicatore di analisi tecnica Market Facilitation Index, ideato da Bill Williams, vuole verificare l'affidabilità dei movimenti di prezzo in relazione al VOLUME corrispondente della barra. Il calcolo è molto semplice si sottrae all'HIGH di ogni barra il valore LOW e si divide per il VOLUME.

Interpretazione/Utilizzo:

Volume in aumento e MFI in aumento: significa che il mercato si muove principalmente in una direzione e sempre più traders stanno partecipando al TREND. Questo è un buon momento per essere già sul mercato nel senso che i prezzi si stanno muovendo.

Volume in diminuzione e MFI in diminuzione: Il mercato è al minimo, in genere a causa di interessi fading.

Spesso questo avviene verso la fine di un TREND. Tuttavia, la direzione di una nuova tendenza successiva potrebbe essere in entrambe le direzioni.

Volume in diminuzione e MFI in aumento: Il mercato si muove principalmente in una direzione, ma non ci sono nuovi partecipanti generando ulteriore volume. Mentre meno del volume si muove il prezzo, l'attività può essere generata principalmente sul pavimento, fingendo il sentimento reale del mercato off-piano.

Volume in aumento e MFI in diminuzione: Il volume è in aumento, indicando più commercio, ma il prezzo non si muove come fortemente nella stessa direzione. Questo si verifica in genere prima di un passo significativo nella direzione opposta.

Parametri Standard:

Non previsti

Esempio:

**SET M = MKTFI()**  
**M > REF(M, 1)**

Restituisce TRUE quando Market Facilitation Index è in salita (oggi maggiore di ieri).

## **Mass Index**

**MassIndex (@periods)**  
**MI (@periods)**

Panoramica:

Il Mass Index identifica le variazioni di TREND utilizzando la variazione dell'ampiezza delle barre.

Genera un segnale di inversione quando si presenta una diminuzione di volatilità dopo un forte incremento della stessa.  
Come indice di volatilità viene considerato il range della barra come differenza tra HIGH e LOW.

Interpretazione/Utilizzo:

Secondo l'inventore del Mass Index, un'inversione può verificarsi, calcolandolo su 25 periodi, quando il Mass Index supera 26 e scende poi al di sotto di 25. Questi mark sono ovviamente settabili dal trader, come anche il periodo di superamento dell'highMark.

Parametri Funzione

Periods            **25**

Parametri Indicatore beeTrader

Periods            **25**

lowMark           **25**

highMark          **26**

Esempio:

```
SET M = MassIndex (25)  
LASTIF (CROSSOVER (M, 26)) < 10  
AND  
CROSSOVER (25, M)
```

Restituisce TRUE solo se il MassIndex a 25 periodi ha appena attraversato verso il basso il valore lowMark (25) e nelle ultime 10 barre aveva anche attraversato verso l'alto il valore highMark (26).

*Nota: l'istruzione SET che semplifica le espressioni delle condizioni del segnale.*

## **Median Price**

```
MEDIANPRICE ()  
MP ()
```

Panoramica:

Il Median Price è semplicemente una media di valori HIGH e LOW di un periodo.

Interpretazione/Utilizzo:

Il Median Price è spesso usato come un modo alternativo di vedere il movimento dei prezzi e soprattutto come componente di altri indicatori di analisi tecnica.

Parametri Standard:

Non previsti



Esempio:

**CROSSOVER (CLOSE, SMA (MP (), 14))**

Restituisce TRUE se il CLOSE ha attraversato la SMA a 14 periodi del Median Price.

## **Money Flow Index**

**MoneyFlowIndex (@periods)**

**MFI (@periods)**

Panoramica:

L'oscillatore di momentum Money Flow Index misura la forza dei volumi, sia al rialzo che al ribasso associando pertanto il movimento del prezzo al volume con cui questo movimento è stato effettuato.

I valori variano in un range 0-100 e si possono stabilire due livelli di attenzione ai valori 20 e 80.

Interpretazione/Utilizzo:

Si utilizza come indicatore di ipercomprato/ipervenduto in modo da ottenere dei segnali di allerta nelle congestioni di mercato, siano esse al rialzo che al ribasso, congestioni che possono preannunciare esaurimento di TREND.

Nella zona 80 e superiore siamo in ipercomprato mentre siamo in ipervenduto nella zona 20 e minore di 20.

Si utilizza come indicatore di divergenze ed in questo caso è un ottimo anticipatore del mercato. Attenzione e fare pratica utilizzandolo con continuità perchè le divergenze spesso continuano per lunghi periodi prima di generare il segnale.

Parametri Standard:

Periods: **15**

Esempio:

**SET M = MFI (15)**

**CROSSOVER (M, 20)**

Restituisce TRUE quando l'indicatore attraversa dal basso verso l'alto il valore di 20: era in ipervenduto ed ha superato la linea di allerta posta a 20.

## **Negative Volume Index**

**NegativeVolumeIndex (@vector)**

**NVI (@vector)**

Panoramica:

Il Negative Volume Index è simile al Positive Volume Index, tranne che pone l'accento sui periodi in cui il volume si riduce rispetto al periodo precedente.

Interpretazione/Utilizzo:

L'interpretazione del Negative Volume Index è che la maggior parte degli investitori vendono quando l'indice sale, e comprano quando l'indice scende.

Parametri Standard:

Vector: **CLOSE**

Esempio:

**TREND (NVI (CLOSE), 30) = DOWN**

Restituisce TRUE quando NVI è in TREND ribassista, su base 30 periodi.

## ***On Balance Volume***

**OnBalanceVolume (@vector)**

**OBV (@vector)**

Panoramica

L'indice On Balance Volume è un indicatore di momentum e misura la pressione che gli investitori effettuano sull mercato. Aggiunge i volumi nei giorni in cui la chiusura è positiva e li sottrae in quelli in cui è negativa. Cerca di individuare un bilanciamento di volumi comprati e venduti, in modo che le discrepanze, gli eccessi indichino le forze presenti.

Interpretazione/Utilizzo:

L'indice On Balance Volume precede in genere i movimenti dei prezzi.

Un rialzo del suo valore indica infatti un ritorno dei compratori tanto quanto un ribasso indica l'arrivo di venditori. Non è importante il suo valore assoluto ma la sua direzione. Se On Balance Volume sta salendo mentre i prezzi rimangono compressi ecco che ci si attende un rally in salita, se scende ma i prezzi salgono, ecco una divergenza che può anticipare l'inizio di un TREND al ribasso. Una rottura di un livello importante precedente indica un TREND nella direzione della rottura.

L'utilizzo classico si ha individuando i supporti e resistenze del titolo in esame magari applicando i ritracciamenti di Fibonacci e, su quei livelli valutare l'indicazione del On Balance Volume.

Parametri Standard:

Vector: **CLOSE**

Esempio:

**TREND (OBV (CLOSE), 30) = UP AND TREND (CLOSE, 30) = DOWN**

Restituisce TRUE quando OBV è in TREND rialzista ed il sottostante è in TREND ribassista.

Per individuare una divergenza dobbiamo avere il TREND dell'OBV in salita ed il TREND del sottostante in discesa (o all'opposto OBV in discesa e sottostante in salita).

## ***Parabolic Stop & Reversal (Parabolic SAR)***

**ParabolicSAR (@minAF, @maxAF)**  
**PSAR (@minAF, @maxAF)**

Panoramica:

Questo indicatore è sempre sul mercato (ogni volta una posizione viene chiusa, una posizione opposta viene aperta). L'indicatore Parabolic SAR è utilizzato per impostare dei possibili punti d'inversione del TREND in atto.

Uno Stop & Reverse (SAR) si verifica quando il prezzo del titolo raggiunge il livello del Parabolic SAR che cambierà di posizione: se era sopra il prezzo passerà sotto e viceversa.

Interpretazione/Utilizzo:

E' un indicatore che viene comunemente usato per uscire da una posizione infatti si comporta come una specie di Trailing Stop.

Quando il PSAR è superiore al prezzo del sottostante, ogni nuovo massimo del sottostante incrementa la velocità con cui il PSAR converge verso il prezzo stesso. L'incremento di velocità è pari al valore di MinAF, ma la velocità di convergenza verso il prezzo non supererà comunque il valore MaxAF.

Al contrario se il PSAR è inferiore al prezzo del sottostante dove si terrà conto dei nuovi minimi. Rimane identica la descrizione della velocità e dell'accelerazione.

Parametri Standard:

Min AF (Accumulation Factor): **0.02**

Max AF (Accumulation Factor): **0.2**

Esempio:

**CROSSOVER (CLOSE, PSAR (0.02, 0.2)) = TRUE**

Restituisce TRUE se il CLOSE ha attraversato il Parabolic SAR.

## ***Performance Index***

**PerformanceIndex (@vector)**  
**PFI (@vector)**

Panoramica:

Il Performance Index mostra il prezzo di uno strumento come un valore

normalizzato. Se il Performance Index mostra 50, allora il prezzo dello strumento è aumentato del 50% dall'inizio del calcolo. Al contrario, se l'indicatore mostra -50, allora il prezzo dello strumento è diminuito del 50% dall'inizio del calcolo.

Interpretazione/Utilizzo:

Viene utilizzato in abbinata con l'analisi delle candele giapponesi e serve per dare una valutazione normalizzata alla candela stessa.

Ad esempio un HAMMER può avere un valore maggiore/minore di un precedente HAMMER e può essere considerato quindi valido o scartato.

E' quasi come utilizzare il Candle Volume che però non da valori normalizzati e comparabili con lo stesso metro.

Parametri Standard:

Vector: **CLOSE**

Esempio:

**PFI (CLOSE) > 45**

Restituisce TRUE se l'Indice di Performance è oltre il 45%.

### ***Positive Volume Index***

**PositiveVolumeIndex (@vector)**

**PVI (@vector)**

Panoramica:

Il Positive Volume Index pone l'accento sui periodi in cui il volume aumenta rispetto al periodo precedente.

Interpretazione/Utilizzo:

L'interpretazione del Positive Volume Index è che la maggior parte degli investitori comprano quando l'indice sale, e vendono quando l'indice scende.

Parametri Standard:

Vector: **CLOSE**

Esempio:

**TREND (PVI (CLOSE), 30) = UP**

Restituisce TRUE quando PVI è in TREND rialzista su base 30 periodi.

### ***Pretty Good Oscillator***

**PrettyGoodOscillator (@periods)**

## **PGO (@periods)**

Panoramica:

Il Pretty Good Oscillator è un oscillatore che misura la distanza del prezzo CLOSE corrente da una Moving Average, il tutto diviso per l'Average True Range. L'ideatore del Pretty Good Oscillator, Mark Johnson attribuisce segnale BUY se l'indicatore supera 3 e segnale SELL se l'indicatore è inferiore a -3.

Interpretazione/Utilizzo:

Al di là dell'interpretazione dell'ideatore che genererebbe pochi segnali e quasi alla fine del TREND dato il ritardo insito nell'algoritmo di calcolo, viene utilizzato proprio all'incrocio della linea dello zero: se la supera il segnale è BUY se la oltrepassa dall'alto verso il basso, il segnale è SELL.

Una volta che i segnali sono stati generati, si può usare come avvertimento di fine TREND o periodo di sideways (laterale) qualora l'indicatore pur rimanendo nel territorio che ha generato il segnale stesso, diminuisca di valore.

Parametri Standard:

Periods: **14**

Esempio:

**SET PG = PGO (14)**  
**CROSSOVER (PG, 0)**

Restituisce TRUE quando il PGO attraversa dal basso verso l'alto la linea dello zero (ieri era inferiore a zero, oggi è superiore).

## **Price Rate of Change (ROC)**

**PriceRateOfChange (@vector, @periods)**  
**PROC (@vector, @periods)**  
**ROC (@vector, @periods)**

Panoramica:

Il Price Rate of Change, è un indicatore di momentum e anche di volatilità, indica la forza del mercato calcolando la differenza tra il prezzo corrente e il prezzo di uno o più periodi del passato (momentum) ma anche la variazione che c'è stata (volatilità).

Semplicemente se si vuole il ROC a 12 periodi è sufficiente dividere il prezzo di chiusura di oggi con il prezzo delle 12 sedute precedenti. Il risultato ottenuto va moltiplicato per 100.

Interpretazione/Utilizzo:

Il Price Rate of Change si utilizza per avere una misura della forza del mercato. In un TREND positivo avremmo più probabilità di continuità di salita se il ROC è in territorio positivo e in crescita. Viceversa per un TREND di discesa.

Anche l'attraversamento dello zero viene utilizzato per spunti operativi.

Parametri Standard:

Vector: **CLOSE**

Periods: **12**

Esempio:

**ROC (CLOSE, 12) > 0 AND REF (ROC (CLOSE, 12),1) < 0**

Restituisce TRUE se il ROC più recente è entrato in territorio positivo.

### **Price CCI Divergence**

**PriceCCIDivergence(@periods,@cciPeriods, @angle, @highThreshold, @lowThreshold)**  
**PriceCCIDivergence(@periods,@cciPeriods, @angle, @highThreshold, @lowThreshold,**  
**@filterThreshold)**

**PCCIDIV (@periods, @cciPeriods, @angle, @highThreshold, @lowThreshold)**

**PCCIDIV(@periods,@cciPeriods,@angle,@highThreshold,@lowThreshold,**  
**@filterThreshold)**

Panoramica:

Questo indicatore rileva divergenze ribassiste e rialziste tra il CCI ed i prezzi. Contiene due indicatori: il rilevamento delle divergenze ribassiste e di quelle rialziste.

Interpretazione/Utilizzo:

1 indica un possibile trend rialzista;

0 indica un trading range;

-1 indica un possibile trend ribassista.

Parametri Standard:

Periods: **30**

CCI Periods: **20**

Angle: **0**

High Threshold: **100**

Low Threshold: **-100**

Vector: **CLOSE**

Parametri Opzionali:

Filter Threshold: **0.5**

Esempio:

**PCCIDIV (30, 20, 0, 100, -100) > 0**

Restituisce TRUE quando il Price CCI Divergence individua una divergenza verso l'alto tra il CCI ed il prezzo.

## **Price Volume Trend**

**PriceVolumeTrend (@vector)**  
**PVT (@vector)**

Panoramica:

Conosciuto anche come Volume Price Trend, questo indicatore è costituito da un volume cumulativo che aggiunge o sottrae un multiplo della variazione percentuale del prezzo e il volume attuale, a seconda che i movimenti siano rialzisti o ribassisti. PVT viene utilizzato per determinare l'equilibrio tra la domanda e l'offerta di uno strumento.

Interpretazione/Utilizzo:

L'andamento di questo indicatore precede in genere i movimenti dei prezzi reali. La premessa è che investitori ben informati compreranno con l'aumento dell'indice e gli investitori non informati compreranno quando l'indice scende. L'utilizzo è per trader di posizione, coloro che hanno intenzione di effettuare investimenti di lungo periodo.

Parametri Standard:

Vector: **CLOSE**

Esempio:

**TREND (PVT (CLOSE), 20) = UP**

Restituisce TRUE quando PVT è in TREND rialzista su base 20 periodi. (TREND è una funzione di beeTrader®).

## **Qstick**

**QSTICK (@periods, @maType)**  
**QS (@periods, @maType)**

Panoramica:

Il QStick Indicator, ideato da Tushar Chande, viene utilizzato con timeframe intraday e molto spesso come ausilio all'analisi delle Candlestick.

Il QStick Indicator è la Moving Average a 10 periodi della differenza tra CLOSE e OPEN. Differenza che l'autore ricorda essere il cuore dell'analisi Candlestick perché misura e rappresenta la forza e la consistenza dei ribassisti o dei rialzisti. Il QStick Indicator si utilizza su consiglio dell'ideatore con una base temporale di 8 periodi.

Viene disegnato calcolando anche una Signal Line utilizzando una Simple Moving Average a 15 periodi.

Interpretazione/Utilizzo:

I segnali operativi vengono generati dall'attraversamento della linea dello zero. Il segnale BUY viene generato all'attraversamento, da parte del QStick

Indicator, della linea dello zero e quindi al passaggio dal quadrante negativo a quello positivo. Il segnale SELL scatta nel momento in cui l'indicatore scende nuovamente in territorio negativo sotto la linea dello zero.

La Signal Line sarà un ausilio per identificare gli inizi della salita o della discesa, anticipando l'attesa dell'indicatore all'attraversamento dello zero.

Parametri Funzione

Periods **10**  
MA Type **SIMPLE**

Parametri Indicatore beeTrader

Periods **10**  
MA Type **SIMPLE**  
Signal Periods **15**

Esempio:

```
SET QIndicator = QStick (10, SIMPLE)  
SET SignalLine = SMA (QIndicator, 15)  
CROSSOVER (QIndicator, 0) AND SignalLine < QIndicator
```

Restituisce TRUE quando il Qstick attraversa dal basso verso l'alto la linea dello zero e contemporaneamente è maggiore della sua Signal Line (SMA a 15 periodi).

## **Random Walk Index**

```
RandomWalkIndexTop (@periods)  
RWIT (@periods)
```

```
RandomWalkIndexBottom (@periods)  
RWIB (@periods)
```

Panoramica:

Il Random Walk Index è un'indicatore/oscillatore che cerca di determinare se il movimento dei prezzi di uno strumento finanziario sia casuale (random) o conseguenza di un TREND statisticamente significativo e quindi più attendibile. L'indicatore è composto da due funzioni, la RWIT e la RWIB, la prima identifica i TREND long, la seconda i TREND short.

*Nota: si può visualizzare anche come istogramma.*

Interpretazione/Utilizzo:

L'interpretazione è molto semplice, maggiore è l'ampiezza del range tra le due bande disegnate maggiore è la forza del TREND, LONG o SHORT in base a quale delle due risulta sopra la linea dello 0.

Il segnale di entrata può essere generato anche dall'incrocio delle due funzioni e se RWIT incrocia dal basso verso l'alto RWIB ecco scattare un segnale BUY.

Opportuno l'utilizzo congiunto ad un indicatore di momentum che ne elimini i falsi segnali.

Parametri Standard:



Periods: **7 o 14**

Esempio:

```
SET rdTop = RWIT (7)  
SET rdDown = RWIB (7)  
CROSSOVER (rdTop, rdDown)
```

Restituisce TRUE quando rdTop incrocia verso l'alto rdDown (ieri RWIT era inferiore a RWIB ed oggi è superiore).

## **Ravi**

```
RAVI (@vector, @shortCycle, @longCycle)
```

Panoramica:

L'indicatore RAVI (Range Action Verification Index) è un indicatore di TREND su timeframe daily sviluppato da Tushar Chande. Il RAVI è calcolato utilizzando Moving Average di diverse lunghezze. La prima è una Moving Average corta, seguendo le indicazioni dell'ideatore con un periodo di 7 barre. La seconda è una Moving Average lunga, seguendo le indicazioni dell'ideatore con un periodo di 65 barre.

Interpretazione/Utilizzo:

Il mercato è da considerarsi in TREND se il valore RAVI è superiore a 0,3 per un TREND LONG, inferiore a -0,3 in caso di TREND SHORT. In caso contrario, il mercato si sta consolidando.

Parametri Standard:

Vector: **CLOSE**

Short Cycle: **7**

Long Cycle: **65**

Esempio:

```
RAVI (CLOSE, 7, 65) > 0,3
```

Restituisce TRUE quando l'indicatore RAVI è maggiore di 0,3.

## **Relative Strength Index (RSI)**

```
RelativeStrengthIndex (@vector, @periods)  
RSI (@vector, @periods)
```

Panoramica:

L'RSI è un popolare indicatore sviluppato dal trader Welles Wilder. L'RSI è un indicatore che mostra la forza relativa dei prezzi di uno strumento.

Interpretazione/Utilizzo:

Il metodo più utilizzato per interpretare l'RSI è considerare come zone di ipercomprato i valori superiori a 70, mentre saremo in ipervenduto qualora segnasse valori inferiori al 30. Va comunque fatto notare come la dottrina classica dell'analisi tecnica identifica questi livelli a 80 e 20, anziché 70 e 30 che genererebbero alcuni falsi segnali.

Parametri Funzione

Vector           **CLOSE**  
Periods           **14**

Parametri Indicatore beeTrader

Vector           **CLOSE**  
Periods           **14**  
lowMark          **30**  
highMart         **70**

Esempio:

**RSI (CLOSE, 14) > 80**

Restituisce TRUE quando il valore dell'RSI è maggiore di 80.

**RGB (solo funzione)**

**RGB (@red, @green, @blue)**

Panoramica:

La funzione RGB si utilizza per definire il gradiente del colore da applicare ai parametri di uscita degli indicatori (PLOTCOLORx) nel caso in cui il colore desiderato non sia presente tra quelli predefiniti.

Interpretazione/Utilizzo:

La funzione RGB serve per esprimere un colore in termini numerici. I parametri di ingresso della funzione devono essere numeri interi compresi tra 0 e 255. Su internet si trovano numerosi traduttori o tabelle per trovare l'insieme dei tre parametri necessari per riprodurre il colore desiderato.

Parametri Standard:

Non previsti

Esempio:

**SET PLOTCOLOR1 = RGB (255, 255, 153)**

Imposta il colore del primo parametro di uscita dell'indicatore con la tonalità del colore di sfondo dell'esempio di cui sopra (giallo chiaro).

## Schaff Trend Cycle

**SchaffTrendCycle (@vector, @periods, @shortCycle, @longCycle, @maType)**  
**STC (@vector, @periods, ShortCycle, @longCycle, @maType)**

Panoramica:

L'indicatore Schaff Trend Cycle ideato da Doug Schaff nel 2008, si basa sulla supposizione che i TREND accelerino e rallentino ciclicamente, quindi reagisce ai bruschi cambiamento di prezzo, ignorando nel contempo le piccole variazioni. Secondo la concezione di Doug Schaff dell'indicatore, a seguito di un determinato periodo temporale, sui mercati il TREND torna all'originario punto di sviluppo, ed il ciclo dei suoi successivi movimenti inizia a ripetersi.

Interpretazione/Utilizzo:

L'utilizzo principale dell'indicatore Schaff Trend Cycle è rappresentato dalla generazione di un segnale SELL quando la linea dell'indicatore interseca al ribasso la linea del 75%, e la generazione di un segnale BUY quando la linea dell'indicatore interseca al rialzo la linea del 20%. Doug Schaff propone inoltre un filtraggio dei segnali: se la barra che segue la barra del segnale si chiude oltre HIGH precedente, ciò sta a significare una maggiore probabilità del movimento dei prezzi e, di conseguenza, degli acquisti. Il segnale SELL è rappresentato dalla situazione opposta, quando cioè la barra che segue la barra del segnale, si chiude al di sotto del LOW della barra precedente.

Parametri Standard:

Vector: **CLOSE**

Periods: **10**

Short Cycle: **23**

Long Cycle: **50**

MA Type: **SIMPLE**

Esempio:

**SET schaft = STC (CLOSE, 10, 23, 50, SIMPLE)**  
**CROSSOVER (schaft, 25)**

Restituisce TRUE quando Schaff Trend Cycle incrocia verso l'alto la linea del 25 generando un segnale BUY (ieri era minore di 25 oggi è maggiore di 25).

*Nota: utilizzo di SET per l'assegnazione di STC e parametri alla sola parola schaft.*

## Standard Error

**StandardError (@vector, @periods, @smoothPeriods)**  
**StdErr (@vector, @periods, @smoothPeriods)**

Panoramica:

Standard Error è un calcolo statistico che misura l'errore standard in una serie

di dati.

Interpretazione/Utilizzo:

Maggiore è l'errore più i dati saranno lontani dalla Linear Regression.

Parametri Standard:

Vector: **CLOSE**

Periods: **21**

Standard Deviations: **3**

Esempio:

**STDERR (CLOSE, 21, 3) > REF (STDERR (CLOSE, 21, 3), 10)**

Restituisce TRUE quando l'Errore Standard a 21 periodi è maggiore di quello di 10 periodi fa.

### **Standard Deviation**

**StandardDeviations (@vector, @periods, @standardDeviations, @maType)**

**SDV (@vector, @periods, @standardDeviations, @maType)**

**StdDev (@vector, @periods, @standardDeviations, @maType)**

Panoramica:

Standard Deviation è un calcolo statistico comune che misura la volatilità. Molti indicatori di analisi tecnica si basano su deviazioni standard come parte del loro calcolo.

Interpretazione/Utilizzo:

Maggiore è lo scostamento dei prezzi maggiore è la volatilità. Elevati valori di Standard Deviation indicano che il prezzo, o l'indicatore, è più volatile del solito.

Parametri Standard:

Vector: **CLOSE**

Periods: **21**

Standard Deviations: **2**

MA Type: **SIMPLE**

Esempio:

**SDV (CLOSE, 21, 2, SIMPLE) > REF (SDV (CLOSE, 21, 2, SIMPLE), 10)**

Restituisce TRUE quando il valore di 2 Standard Deviations a 21 periodi è superiore a 10 periodi fa.

## Stochastic Momentum Index

**SMIK(@%KPeriods, @%KSmoothPeriods, @%KDoublePeriods, @%DPeriods, @maType, @%DMAType)**  
**SMID(@%KPeriods, @%KSmoothPeriods, @%KDoublePeriods, @%DPeriods, @maType, @%DMAType)**

Panoramica:

Lo Stochastic Momentum Index, sviluppato da William Blau, si è ottenuto modificando marginalmente la formula dello Stocastico Oscillator, ovvero aggiungendo un nuovo parametro (%K Double Periods) usato per smussare ulteriormente il risultato del %K. Il risultato è quello di avere un indicatore che genera meno falsi segnali ma leggermente in ritardo rispetto all'originale.

Interpretazione/Utilizzo:

Lo Stochastic Momentum Index si utilizza ovviamente nello stesso modo dello Stochastic Oscillator perchè rimangono inalterati i componenti:

%K (la funzione SOPK) e %D (la funzione SOPD). %K è spesso visualizzata su un grafico azionario come una linea continua e %D è spesso indicata da una linea tratteggiata. Un metodo di utilizzo dello Stochastic Oscillator è quello di vendere quando una componente (%K oppure %D) scende dal valore 40 e comprare quando uno dei componenti sale dal valore -40. 40 è la soglia che indica ipercomprato e -40 quella che indica ipervenduto. Un altro modo di interpretare lo Stochastic Oscillator è quello di acquistare quando %K sale sopra il %D, e viceversa, vendere quando %K scende al di sotto %D. Naturalmente è validissimo per le divergenze che si verificheranno mentre l'indicatore è nelle zone critiche, oltre 40 o sotto il -40.

Parametri Funzione

%K Periods           **14**  
%K SmoothPeriods   **2**  
%K Double Periods   **3**  
%D Periods           **9**  
MA Type               **SIMPLE**  
%D MA Type           **SIMPLE**

Parametri Indicatore beeTrader

%K Periods           **14**  
%K SmoothPeriods   **2**  
%K Double Periods   **3**  
%D Periods           **9**  
MA Type               **SIMPLE**  
%D MA Type           **SIMPLE**  
lowMark               **-40**  
highMark              **40**

Esempio:

**SMID (14, 2, 3, 9, SIMPLE, SIMPLE) > 40 OR**  
**SMIK (14, 2, 3, 9, SIMPLE, SIMPLE) > 40**

Restituisce TRUE se lo Stochastic Momentum Index è in territorio di

ipercomprato.

## **Simple Super Trend**

**SimpleSuperTrend (@atrLength, @atrMultiplier)**

**SST (@atrLength, @atrMultiplier)**

Panoramica:

Il Simple SuperTrend è una versione semplificata, quindi con meno parametri e una maggiore semplicità di calcolo, dell'indicatore ideato da Oliver Seban che può applicato a pressoché tutti gli strumenti finanziari. E' simile nella sua concezione al Parabolic SAR, con in più il pregio di disporsi in orizzontale nei momenti di trading range, evitando quindi un'uscita anticipata dal mercato, il Simple SuperTrend è un indicatore che può essere molto utile quindi per la corretta uscita da una posizione a mercato.

Interpretazione/Utilizzo:

E' molto facile da interpretare perché è costituito da una serie di punti, uno per ogni barra, sotto il CLOSE che ci indica l'inizio di un TREND rialzista e una serie di punti, sempre uno per ogni barra, sopra il CLOSE che ci da l'indicazione di un TREND ribassista, per cui tutto quello che c'è da fare in un'operatività intraday è aspettare che cambi posizione. Unico neo, il ritardo nel cambiare direzione, quando accade la differenza tra i due valori della barra del grafico che hanno prodotto il cambiamento, sarà molto ampia, producendo un loss percentuale considerevole. L'uso di questo indicatore deve essere combinato con altri indicatori che individuino il cedimento del TREND, ad esempio il "Price ROC".

Parametri Standard:

Atr Length: **9**

Atr Multiplier: **1**

Esempio:

**ST (9, 1) > CLOSE**

Restituisce TRUE quando il Simple SuperTrend è superiore al CLOSE (TREND SHORT).

## **Super Trend**

**SuperTrend (@periods, @strength, @atrMultiplier)**

**ST (@periods, @strength, @atrMultiplier)**

Panoramica:

Il SuperTrend è un indicatore ideato da Oliver Seban che può applicato a pressoché tutti gli strumenti finanziari. E' simile nella sua concezione al Parabolic SAR, con in più il pregio di disporsi in orizzontale nei momenti di trading range, evitando quindi un'uscita anticipata dal mercato, il SuperTrend è un indicatore che può essere molto utile quindi per la corretta uscita da una posizione a mercato.

Interpretazione/Utilizzo:

E' molto facile da interpretare perché è costituito da una serie di punti, uno per ogni barra, sotto il CLOSE che ci indica l'inizio di un TREND rialzista e una serie di punti, sempre uno per ogni barra, sopra il CLOSE che ci da l'indicazione di un TREND ribassista, per cui tutto quello che c'è da fare in un'operatività intraday è aspettare che cambi posizione. Unico neo, il ritardo nel cambiare direzione, quando accade la differenza tra i due valori della barra del grafico che hanno prodotto il cambiamento, sarà molto ampia, producendo un loss percentuale considerevole. L'uso di questo indicatore deve essere combinato con altri indicatori che individuino il cedimento del TREND, ad esempio il "Price ROC".

Parametri Standard:

Periods: **9**

Strength: **5**

Atr Multiplier: **1**

Esempio:

**ST (9, 5, 1) > CLOSE**

Restituisce TRUE quando il SuperTrend è superiore al CLOSE (TREND SHORT).

## **Swing Index**

**SwingIndex (@limitMoveValue)**

**SI (@limitMoveValue)**

Panoramica:

Lo Swing Index è un popolare indicatore che mostra la forza comparativa dei prezzi all'interno di uno strumento, confrontando prezzi attuali OPEN, HIGH, LOW, CLOSE, con i prezzi precedenti.

Interpretazione/Utilizzo:

Lo Swing Index da solo non da molte informazioni e viene pertanto utilizzato come componente dell'Accumulative Swing Index. Si può utilizzare per valutare una barra rispetto all'altra ma senza ricavarne informazioni di medio/lungo periodo.

Parametri Standard:  
Limit Move Value: **1**

Esempio:

**SI (1) > 0**

Restituisce TRUE quando lo Swing Index è in territorio positivo.

### **Trade Volume Index**

**TradeVolumeIndex (@vector, @minimumTickValue)**  
**TVI (@vector, @minimumTickValue)**

Panoramica:

Il Trade Volume Index mostra se un titolo è in fase di accumulazione o distribuzione basandosi sugli scambi che avvengono sul bid o sull'ask. Per questo motivo, per avere una indicazione precisa, si usa su time frame bassi, minuto o ore, e solo dopo che sono iniziati gli scambi. Questo perchè altrimenti non potrebbe distinguere i dati storici che vengono passati.

Interpretazione/Utilizzo:

Quando l'indicatore è in aumento, il mercato è in fase di accumulo. Viceversa, quando l'indicatore scende, il mercato è in fase di distribuzione. I prezzi possono invertire quando l'indicatore attraversa la linea dello zero.

Parametri Standard:

Vector: **CLOSE**

Minimum Tick Value: **0.25**

Esempio:

**TVI (CLOSE, 0.25) > 0**

Restituisce TRUE quando il Trade Volume Index è in territorio positivo.

### **Typical Price**

**TypicalPrice ()**  
**TP ()**

Panoramica:

Il Typical Price è una media dei valori HIGH, LOW, e CLOSE di un periodo.

Interpretazione/Utilizzo:

Il Typical Price è utilizzato come componente per il calcolo dei vari indicatori di analisi tecnica.



Parametri Standard:  
Non previsti

Esempio:

**CROSSOVER (CLOSE, SMA (TP (), 14))**

Restituisce TRUE se il CLOSE ha attraversato la SMA a 14 periodi del Typical Price.

### **Twiggs Money Flow**

**TwiggsMoneyFlow (@periods)**  
**TMF (@periods)**

Panoramica:

Il Twiggs Money Flow è una variante del Chaikin Money Flow, ideato da Colin Twiggs. Rispetto al Chaikin Money Flow, utilizza il True Range anziché il range HIGH-LOW, così facendo si limitano i picchi dovuti ai gaps degli strumenti finanziari. Viene inoltre usata una Welles Wilder Smoothing sui volumi per evitare che picchi di VOLUME inficino il risultato.

Interpretazione/Utilizzo:

Quando il Twiggs Money Flow è superiore a 0, siamo in una fase di accumulazione, e quindi i prezzi si tenderanno al rialzo. Quando Twiggs Money Flow è inferiore a 0, siamo in una fase di distribuzione e i prezzi tenderanno a muoversi verso il basso.

*Nota: a volte i valori sono prossimi allo zero ed allora nell'indicatore di beeTrader® è stato aggiunto il Multiplier per avere valori visibili.*

Parametri Funzione

Periods           **21**

Parametri Indicatore beeTrader

Periods           **21**

Multiplier       **1000**

Esempio:

**SET T = 1000 \* TMF (21)**  
**T > 0 AND REF (T, 1) < 0**

Restituisce TRUE quando il TMF passa da sotto lo zero a sopra lo zero generando un segnale BUY.

*Nota: utilizzo di REF per avere uno script alternativo alla scrittura di CROSSOVER.*

## Ultimate Oscillator

**UltimateOscillator (@cycle1, @cycle2, @cycle3)**

**ULTOSC (@cycle1, @cycle2, @cycle3)**

Panoramica:

Larry Williams ha voluto costruire un oscillatore che non risentisse molto del settaggio dei periodi di calcolo, cosa che spesso inficia i risultati di molti 7, 14, 21 indicatori. Ebbene, nel suo algoritmo tiene conto di tre periodi diversi cercando così di ottenere un oscillatore con pochi falsi segnali.

Il concetto parte dall'osservazione che la pressione, cioè il momentum che fa muovere i prezzi, genera poi un TREND più sviluppato, definito e duraturo.

Per chiarezza mettiamo il calcolo con cui è stato sviluppato affinché si possa meglio comprendere la costruzione:

$$\text{ULTOSC} = 100 * [(4 * \text{avg a } 7) + (2 * \text{avg a } 14) + (\text{avg a } 28)] / (4 + 2 + 1)$$

Interpretazione/Utilizzo:

Come per gli altri oscillatori anche questo ha dei valori che variano in un range da 0 a 100. Ci sono le cui zone di ipercomprato e ipervenduto che sono definite dai valori oltre 70 e sotto il 30. Solitamente si agisce in caso di divergenze tra il grafico dei prezzi e quello dell'indicatore. In particolare abbiamo segnale BUY nel caso di divergenza così formata: l'oscillatore ha un nuovo picco o semplicemente continua a salire mentre il prezzo segna un nuovo minimo. Il segnale SELL è generato con una conformazione contraria a quanto appena scritto.

Parametri Funzione

Cycle1           **7**  
Cycle2           **14**  
Cycle3           **28**

Parametri Indicatore beeTrader

Cycle1           **7**  
Cycle2           **14**  
Cycle3           **28**  
lowMark         **30**  
highMark        **70**

Esempio:

**SET U = ULTOSC (7,14, 28)**

**U > REF (U, 1) AND TREND (CLOSE, 30) = DOWN**

*Restituisce TRUE quando Ultimate Oscillator è in salita ed il TREND è in discesa.*

## Variance

**Variance (@vector, @periods, @maType)**

**VAR (@vector, @periods, @maType)**

Panoramica:

Restituisce la varianza statistica del vettore utilizzato.

Interpretazione/Utilizzo:

Maggiore è lo scostamento dei prezzi maggiore è la volatilità. Elevati valori di Variance indicano che il prezzo, o l'indicatore, è più volatile del solito.

Parametri Standard:

Vector: **CLOSE**

Periods: **21**

MAType: **SIMPLE**

Esempio:

```
VAR (CLOSE, 21, SIMPLE) > REF (VAR (CLOSE, 21, SIMPLE), 10)
```

Restituisce TRUE quando il valore della varianza a 21 periodi è superiore a 10 periodi fa.

### ***Volume Rate of Change (Volume ROC)***

```
VolumeRateOfChange (@vector, @periods)
```

```
VROC (@vector, @periods)
```

Panoramica:

L'indicatore Volume Rate of Change si differenzia dal ROC unicamente perché il calcolo è fatto sul volume anziché sul prezzo del sottostante.

Interpretazione/Utilizzo:

E' risaputo che i volumi supportano i TREND ed ecco che un brusco aumento del Volume ROC può essere utilizzato per individuare una possibile esplosione di prezzi.

Parametri Standard:

Vector: **VOLUME**

Periods: **12**

Esempio:

```
VROC (VOLUME, 12) > 0 AND REF (VROC VOLUME, 12), 1) < 0
```

Restituisce TRUE se il Volume ROC più recente è entrato in territorio positivo.

### ***Weighted Close***

```
WeightedClose ()
```

```
WC ()
```

Panoramica:

Il Weighted Close è una media giornaliera di OPEN, HIGH, LOW, e CLOSE, dove si da maggior peso al CLOSE.

Interpretazione/Utilizzo:

L'indicatore Weighted Close è un metodo semplice per avere una visione pesata dell'andamento dei prezzi di mercato magari sovrapponendolo al grafico tradizionale. Viene tipicamente usato assieme ad una media mobile per individuare i movimenti di prezzi in maniera smussata.

Parametri Standard:

Non previsti

Esempio:

**WC () > REF (WC (), 1)**

Restituisce TRUE quando il Wighted Close è superiore al valore precedente.

### ***Wilder Volatility***

**WilderVolatility (@periods)**

**WV (@periods)**

Panoramica:

Per il calcolo della volatilità aritmetica, non statistica (Historical Volatility Index), Wilder ha realizzato la sua formula partendo dal True Range che calcola la differenza fra barre che abbiano superato al rialzo o al ribasso le barre precedenti, creando un range che sia vero, appunto True Range.

$$WV = ((Periods - 1) * WV[1] + True Range) / Periods$$

Interpretazione/Utilizzo:

Può essere utilizzato come ricerca o valutazione di strumenti finanziari più o meno volatili.

Parametri Standard:

Periods: **30 o 60 o 90**

Esempio:

**WV (30) > WV (60)**

### ***Zero Line (solo funzione)***

**ZeroLine ()**

## **ZERO ()**

Panoramica:

La funzione Zero Line viene utilizzata negli indicatori per disegnare la linea al valore 0.

Interpretazione/Utilizzo:

Come sopra

Parametri Standard:

Non previsti

Esempio:

**SET PLOT3 = ZERO()**

Disegna la linea dello 0 come terzo (PLOT3) parametro di uscita dell'indicatore.

## **Zig Zag**

**ZigZag (@vector, @pointsOrPercent, @retrace)**

Panoramica:

E' un indicatore che mira ad individuare gli Swing del mercato, ovvero le inversioni di trend.

Interpretazione/Utilizzo:

Se l'indicatore ha un andamento in salita identifica un TREND LONG, al contrario identifica un TREND SHORT.

Parametri Standard:

Vector: **CLOSE**

PointsOrPercent: **Percent**

Retrace: **10**

Esempio:

**ZigZag (CLOSE, Percent, 10) > CLOSE**

Restituisce TRUE l'indicatore ZigZag calcolato al 10% è maggiore del CLOSE.

## Strategy Position

Questa sezione del manuale del linguaggio EasyScript® riporta le funzioni relative allo stato della Strategy attiva, possono essere utilizzate per gestire la Strategy nel suo complesso in Money Management. L'utilizzo di queste funzioni è significativo solo nei Signal o nelle User Function utilizzate comunque nei Signal.

Queste funzioni vengono forzano l'esecuzione degli script barra per barra e non tutte le barre in un singolo passaggio.

Esempio di utilizzo delle Strategy Position:

```
INPUTS: @trailAmount(50), @trailPercent(10), @loss(-50)  
  
#Simula Trailing Stop  
SET ru = RunUp()  
SET above = ru - @trailAmount  
SET pcnt = @trailAmount - (above * @trailPercent / 100.0)  
SET TRAIL_COND = (RunUp() > @trailAmount) AND (OpenPosition() < pcnt)  
  
#Simula Stoploss  
SET STOP_COND = DrawDown() < @loss  
  
#Condizione di uscita  
TRAIL_COND OR STOP_COND
```

## Bars Since Entry

### BarsSinceEntry()

Panoramica:

Bars Since Entry rappresenta il numero di barre trascorse dall'ultimo Signal di entrata eseguito (Buy oppure Sell).

Interpretazione/Utilizzo:

La funzione Bars Since Entry può essere utilizzata ad esempio per chiudere posizioni aperte da troppo tempo.

Parametri Standard:

Non previsti

Esempio:

```
BarsSinceEntry() > 25
```

Restituisce TRUE quando sono trascorse più di 25 barre dall'entrata a mercato.

## ***Bars Since Exit***

### **BarsSinceExit()**

Panoramica:

Bars Since Exit rappresenta il numero di barre trascorse dall'ultimo Signal di uscita eseguito (Exit Long, Exit Short, Stop Loss, Take Profit, Trailing Stop oppure Max Position Open).

Interpretazione/Utilizzo:

La funzione Bars Since Exit può essere utilizzata ad esempio per evitare di aprire posizioni troppo ravvicinate nel tempo.

Parametri Standard:

Non previsti

Esempio:

### **BarsSinceExit() < 25**

Restituisce TRUE quando sono trascorse meno di 25 barre dall'uscita dalla posizione a mercato.

## ***Total Net Profit***

### **TotalNetProfit()**

Panoramica:

Il Total Net Profit di una Strategy rappresenta la somma dei Profit e dei Loss consolidati dai trades della Strategy.

Interpretazione/Utilizzo:

La funzione Total Net Profit può essere utilizzata per impostare preventivamente delle soglie di uscite monetarie riferite all'intera Strategy.

Parametri Standard:

Non previsti

Esempio:

### **TotalNetProfit() > 1000**

Restituisce TRUE quando il Total Net Profit della Strategy è maggiore di 1000.

## ***Open Position***

### **OpenPosition()**

Panoramica:

L'Open Position di una Strategy rappresenta il Profit / Loss dell'eventuale trade aperto al netto di commissioni o slippage, oppure 0 se non ci sono trades in essere.

Interpretazione/Utilizzo:

La funzione Open Position può essere utilizzata per impostare preventivamente delle soglie di uscite monetarie riferite al trade in essere.

Parametri Standard:

Non previsti

Esempio:

**OpenPosition() > 500**

Restituisce TRUE quando l'Open Position della Strategy è maggiore di 500.

### ***Average Entry Price***

**AverageEntryPrice()**

Panoramica:

L'Average Entry Price rappresenta il prezzo medio di carico dello strumento finanziario al quale è applicata la Strategy.

Interpretazione/Utilizzo:

La funzione Average Entry Price può essere utilizzata per impostare preventivamente un prezzo di carico massimo o minimo per lo strumento finanziario al quale è applicata la Strategy.

Parametri Standard:

Non previsti

Esempio:

**AverageEntryPrice() > Last**

Restituisce TRUE quando l'Average Entry Price della Strategy è maggiore del prezzo LAST.

### ***Last Entry Price***

**LastEntryPrice()**

Panoramica:

Il Last Entry Price rappresenta il prezzo di carico dell'ultimo ordine eseguito



sullo strumento finanziario al quale è applicata la Strategy.

Interpretazione/Utilizzo:

La funzione Last Entry Price può essere utilizzata per ridurre le quantità di contratti se l'uscita non genera un loss, quindi in una posizione Long quando il prezzo LAST è maggiore del Last Entry Price.

Parametri Standard:

Non previsti

Esempio:

**LastEntryPrice() > Last**

Restituisce TRUE quando il Last Entry Price della Strategy è maggiore del prezzo LAST.

## ***Current Contracts***

**CurrentContracts()**

Panoramica:

Current Contracts di una Strategy rappresenta il numero di contratti attualmente utilizzati dalla Strategy.

Interpretazione/Utilizzo:

La funzione Current Contracts può essere utilizzata per impostare preventivamente un numero di contratti massimo o minimo per lo strumento finanziario al quale è applicata la Strategy.

Parametri Standard:

Non previsti

Esempio:

**CurrentContracts() >= 2**

Restituisce TRUE quando il Current Contracts della Strategy è maggiore o uguale a 2.

**CurrentContracts() <= -1**

Restituisce TRUE quando il Current Contracts della Strategy è minore o uguale a -1.

## ***RunUp***

### **RunUp()**

Panoramica:

Il RunUp di una Strategy rappresenta il Profit non consolidato massimo ottenuto dalla Strategy.

Interpretazione/Utilizzo:

La funzione RunUp può essere utilizzata per impostare preventivamente delle soglie massimo o minime di RunUp per l'intera Strategy.

Parametri Standard:

Non previsti

Esempio:

### **RunUp() > OpenPositionProfit()**

Restituisce TRUE quando il RunUp è maggiore del Profit dell'Open Position.

## ***DrawDown***

### **DrawDown()**

Panoramica:

Il DrawDown di una Strategy rappresenta il Loss non consolidato massimo ottenuto dalla Strategy.

Interpretazione/Utilizzo:

La funzione DrawDown può essere utilizzata per impostare preventivamente delle soglie massimo o minime di DrawDown per l'intera Strategy, il valore di DrawDown va espresso in valore assoluto.

Parametri Standard:

Non previsti

Esempio:

### **DrawDown() > OpenPositionProfit()**

Restituisce TRUE quando il DrawDown è inferiore al Profit dell'Open Position.

## ***GetMultiplier***

### **GetMultiplier()**

Interpretazione/Utilizzo:

La funzione GetMultiplier imposta il numero di contratti da comprare / vendere come numero intero in modo che il controvalore non superi mai il valore del parametro @Amount.

Parametri Standard:

Non previsti

Esempio:

```
SET Qty = FLOOR (@Amount / GetMultiplier())  
SET CONTRACTS = Qty
```

Restituisce il valore del moltiplicatore del contratto utilizzato per l'invio degli ordini.

## ***ConsecutiveEntries***

### **ConsecutiveEntries()**

Panoramica:

Consecutive Entries di una Strategy rappresenta il numero massimo di entrate successive in trend ammesse il Pyramiding abilitato.

Esempio:

```
ConsecutiveEntries() > 2
```

Restituisce TRUE quando le entrate Pyramiding consecutive della Strategy sono maggiori di 2.

## PlayOptions.it Functions

In questa sezione del manuale del linguaggio EasyScript® vengono riportate alcune funzioni ideate e concepite da PlayOptions.it per fornire un supporto agli utenti di beeTrader®.

### *Trend Slope*

**TrendSlope(@price, @periods)**

Panoramica:

L'indicatore Trend Slope è un indicatore di ipercomprato / ipervenduto efficace sia nei timeframe di breve che di lungo periodo.

Interpretazione/Utilizzo:

Valori positivi di Trend Slope indicano trend in salita, al contrario valori negativi di Trend Slope indicano un trend in discesa. Se ne consiglia l'utilizzo affiancato da una trigger line.

Parametri Standard:

Price: **CLOSE**

Periods: **14**

Esempio:

**CROSSOVER (TrendSlope (CLOSE, 14)), 0)**

Restituisce TRUE quando il Trend Slope ha attraversato la linea dello 0, segnalando quindi un possibile trend rialzista.

### *Fiuto Canale*

**FiutoCanaleLowestChannel()**

**FiutoCanaleHighestChannel()**

**FiutoCanaleLowChannel()**

**FiutoCanaleHighChannel()**

Panoramica:

Fiuto Canale è un indicatore nato per il trading in opzioni, tuttavia garantisce ottimi risultati anche per il trading su sottostanti i futures. E' composto da due canali, il canale Highest - High ed il canale Lowest-Low. Il canale Highest - High rappresenta il trend di lungo periodo, mentre il canale Lowest-Low rappresenta il trend di più breve periodo e quindi più suscettibile a variazione.

Interpretazione/Utilizzo:

Un segnale di possibile trend long viene dato quando il canale short lower si stacca in salita dal canale long lower. Al contrario un segnale di possibile trend short viene dato quando il canale short upper si stacca verso il basso dal canale long upper.

Parametri Standard:  
Non previsti

Esempio:

**FiutoCanaleHighChannel() < FiutoCanaleHighestChannel()**

Restituisce TRUE quando il canale High Channel si è staccato al ribasso dal canale Highest Channel.

### ***Fiuto Facile***

**FiutoFacileLowestChannel()**  
**FiutoFacileHighestChannel()**  
**FiutoFacileLowChannel()**  
**FiutoFacileHighChannel()**

Panoramica:

Fiuto Facile è un'indicatore derivato da Fiuto Canale, del quale mantiene tutte le caratteristiche principali.

Interpretazione/Utilizzo:

Rispetto a Fiuto Canale, dal quale deriva, vi è una Linear Regression che può fornire un'anticipazione dell'inversione di trend quando taglia (CROSSUNDER per l'inversione long - short, CROSSOVER per l'inversione short - long) il canale principale Highest - High.

Parametri Standard:  
Non previsti

Esempio:

**FiutoFacileHighChannel() < FiutoFacileHighestChannel()**

Restituisce TRUE quando il canale High Channel si è staccato al ribasso dal canale Highest Channel.

## Global Function

Questa sezione del manuale del linguaggio EasyScript® riporta le funzioni per l'interscambio di vettori tra diversi script. Possono essere per esempio utilizzate per riportare il vettore di uno script applicato ad un Chart di uno strumento su un Chart di uno strumento diverso, oppure ad un Chart con uno specifico timeframe ad un altro script applicato su un Chart ad un altro timeframe.

Nota: le funzioni SetGlobalVar e GetGlobalVar non eseguono alcuna trasformazione dei dati tra diversi timeframe. Il vettore salvato tramite SetGlobalVar viene trasposto senza modifiche quando viene richiamato tramite la funzione GetGlobalVar.

Esempio 1 di utilizzo delle Global Function:

Chart 1: Future DAX timeframe 5 minuti

Buy Script:

#Assegno all'id 1 il vettore

**SET A = SetGlobalVar (1, (EMA(CLOSE,14) > EMA(CLOSE,21)))**

Chart 2: Future DJ EURO STOXX 50 timeframe 5 minuti

Buy Script:

#Assegno alla variabile A l'id 1 da un altro script

**SET B = GetGlobalVar (1)**

**B**

La condizione Buy Script del Signal applicato al DJ EURO STOXX 50 a 5 minuti è ora relativo al DAX con timeframe 5 minuti.

Esempio 2 di utilizzo delle Global Function:

Chart 1: Future DAX timeframe 5 minuti

Buy Script:

#Assegno all'id 1 il vettore

**SET A = SetGlobalVar (1, (EMA(CLOSE,14) > EMA(CLOSE,21)))**

Chart 2: Future DAX timeframe 1 ora

Buy Script:

#Assegno alla variabile A l'id 1 da un altro script

**SET B = GetGlobalVar (1)**

**B**

La condizione Buy Script del Signal applicato al DAX ad 1 ora è ora relativo al

DAX con timeframe 5 minuti.

## ***SetGlobalVar***

**SetGlobalVar(@id, @vector)**

Panoramica:

La funzione SetGlobalVar viene utilizzata per assegnare ad un determinato numero identificativo (id) un vettore (vector) di uno script. L'id assegnato mediante la funzione SetGlobalVar può poi essere richiamato su un altro script anche su un altro Chart mediante la funzione GetGlobalVar.

Interpretazione/Utilizzo:

Le funzioni SetGlobalVar e GetGlobalVar possono essere utilizzate per riportare vettori tra uno script applicato ad un Chart ad un altro script anche applicato ad un altro Chart.

Esempio:

**SET A = SetGlobalVar (1, (EMA(CLOSE,14) > EMA(CLOSE,21)))**

Assegna all'id 1 il vettore  $EMA(CLOSE,14) > EMA(CLOSE,21)$  per poter essere utilizzato su un altro script.

## ***GetGlobalVar***

**GetGlobalVar(@id)**

Panoramica:

La funzione GetGlobalVar viene utilizzata per richiamare nello script in cui viene utilizzata un vettore precedentemente assegnato ad un determinato numero identificativo (id) su un altro script mediante la funzione SetGlobalVar.

Interpretazione/Utilizzo:

Le funzioni SetGlobalVar e GetGlobalVar possono essere utilizzate per riportare vettori tra uno script applicato ad un Chart ad un altro script anche applicato ad un altro Chart.

Esempio:

**SET B = GetGlobalVar (1)**

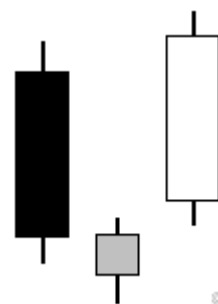
Assegna alla variabile A dello script il vettore con id 1 precedentemente assegnato mediante la funzione SetGlobalVar, nell'esempio di cui sopra il vettore  $EMA(CLOSE,14) > EMA(CLOSE,21)$ .

## I Pattern di Candele Giapponesi

Per disegnare le Candlestick sono necessari quattro prezzi: OPEN, CLOSE, HIGH e LOW. La figura che esce fuori da questi quattro dati è appunto la Candlestick o Candela Giapponese che è formata da un corpo centrale chiamato real-body e da due appendici collegate chiamate shadows, e rispettivamente upper shadow per quella superiore e lower shadow per quella inferiore. L'interpretazione delle candele si basa principalmente su modelli che si formano da periodo a periodo. Gli estremi della figura sono dati dal prezzo LOW e da quello HIGH, mentre il real-body si ottiene dalla differenza tra il prezzo CLOSE e quello OPEN. Se il CLOSE risulta essere superiore all'OPEN allora avremo un rettangolo bianco (o verde), mentre se il CLOSE risulta essere inferiore all'OPEN il real-body sarà nero (o rosso).

Esempio di rappresentazione grafica: Figura 1

Un modello di candele giapponesi è un gruppo di barre di prezzo, come mostrato in figura 1. Tradizionalmente si vedranno candele scure nei giorni in cui il prezzo CLOSE è inferiore a quello OPEN, e chiare nei giorni in cui il prezzo CLOSE è superiore a quello OPEN. A volte nei manuali potreste trovare barre di colore grigio, il che significa che la barra può essere sia bianca che nera.



### Identificazione dei Patterns

Anche se si potrebbe benissimo scrivere degli script per la ricerca di patterns di candlestick, EasyScript® fornisce una semplice funzione in grado di identificare diversi modelli predefiniti:

**CandlestickPattern ()**  
**CSP ()**

Panoramica:

CandlestickPattern() è la funzione per identificare i patterns di candlestick automaticamente. La funzione non ha argomenti e l'uscita è una costante che rappresenta uno dei diversi modelli predefiniti di candele come indicato di seguito.

Esempio:

**CSP () = MORNING\_STAR**



Restituisce TRUE se il Pattern Candlestick è una "Morning Star".

Patterns: La funzione Candlestick restituisce le seguenti costanti:

<b>LONG_BODY</b>	<b>BEARISH_ENGULFING_LINE</b>
<b>DOJI</b>	<b>BEARISH_DOJI_STAR</b>
<b>HAMMER</b>	<b>BEARISH_SHOOTING_STAR</b>
<b>HARAMI</b>	<b>SPINNING_TOPS</b>
<b>STAR</b>	<b>HARAMI_CROSS</b>
<b>DOJI_STAR</b>	<b>BULLISH_TRISTAR</b>
<b>MORNING_STAR</b>	<b>THREE_WHITE_SOLDIERS</b>
<b>EVENING_STAR</b>	<b>THREE_BLACK_CROWS</b>
<b>BEARISH_TWO_CROWS</b>	<b>ABANDONED_BABY</b>
<b>PIERCING_LINE</b>	<b>BULLISH_ENGULFING_LINE</b>
<b>HANGING_MAN</b>	<b>BULLISH_UPSIDE_GAP</b>
<b>DARK_CLOUD_COVER</b>	<b>BULLISH_HAMMER</b>
<b>BULLISH_KICKING</b>	<b>BULLISH_MATCHING_LOW</b>
<b>BEARISH_KICKING</b>	<b>BULLISH_BELT_HOLD</b>
<b>BEARISH_BELT_HOLD</b>	

*Nota: vedi il capitolo 2 per le rappresentazioni visive di questi modelli.*

## Costruire le proprie Funzioni ed Indicatori

### **Premessa**

Il linguaggio di programmazione EasyScript®, come è stato esposto finora, presenta al suo interno una vasta gamma di Function ed Indicator, che l'utente può utilizzare semplicemente interagendo con i parametri elencati nella stringa INPUTS.

Tuttavia per soddisfare anche gli utenti più esigenti, EasyScript® permette la creazione di Function da poter utilizzare come Signal o come Indicator. Sfruttando le variabili primitive e magari le Function già presenti l'unico limite che EasyScript® ha è la creatività dell'utente. Ogni nuova Function / Indicator / Signal creato può essere salvato in modo cifrato con l'utilizzo di una password in modo che il proprio lavoro possa essere utilizzato da altri, ma non visualizzato e copiato.

### **Creazione di proprie Function**

Spesso nei propri Signal o Indicator si utilizzano più volte gli stessi calcoli, o insiemi di essi. Per semplificare la scrittura di Signal, Indicator, Condition ed Expert Advisor, EasyScript® permette di raggruppare questi insiemi di calcoli in un'unica Function, che restituisce il risultato numerico dei calcoli. In EasyScript® una Function deve restituire un solo risultato.

Esempio: supponiamo di voler costruire un'Indicator ed un Signal basati sulla differenza di due Moving Averages. Siccome sia l'Indicator che il Signal utilizzano lo stesso calcolo, è possibile realizzare una Function che esegua questo calcolo. Questa Function si chiama "beeMomentum".

**SET beeMomentum = EMA(@price, @shortPeriods) - EMA(@price, @longPeriods)**

Questa Function prevede tre parametri da definire con la stringa INPUTS assegnando i valori desiderati:

@price: prezzo sul quale calcolare le Moving Averages

@shortPeriods: lunghezza della EMA di breve periodo

@longPeriods: lunghezza della EMA di lungo periodo

**INPUTS: @price(CLOSE), @shortPeriods(7), @longPeriods(14)**

*Nota: Il valore di ritorno della Function deve essere assegnato ad una variabile il cui nome corrisponda a quello della Function stessa oppure alla variabile RETURN.*

La Functions beeMomentum che abbiamo appena realizzato restituirà un valore numerico che rappresenta la differenza tra due Moving Averages. In particolare il risultato sarà positivo quando la EMA di breve periodo è maggiore della EMA di lungo periodo (TREND in salita). Al contrario il risultato sarà negativo quando la EMA di breve periodo è inferiore alla EMA di lungo periodo (TREND in discesa).

### **Creazione di propri Indicator**

Un'Indicator è la rappresentazione grafica del risultato di una o più Function, siano esse integrate in EasyScript® o create dall'utente. Ogni Indicator può definire fino ad un massimo di otto valori di uscita. E' possibile impostare direttamente sul testo dell'Indicator i colori con cui esso verrà disegnato in beeTrader®.

Esempio: supponiamo di voler creare un Indicator che rappresenti graficamente la Function "beeMomentum" creata al paragrafo precedente. L'Indicator avrà un singolo valore di uscita che sarà rappresentato come istogramma verde o rosso a seconda del fatto che il valore sia rispettivamente positivo o negativo.

```
# assegniamo i parametri alle variabili utente utilizzate nella funzione beeMomentum
INPUTS: @price(CLOSE), @shortPeriods(7), @longPeriods(14)

# assegniamo al primo valore di uscita il risultato della Function beeMomentum
SET PLOT1 = beeMomentum (@price, @shortPeriods, @longPeriods)

# impostiamo il colore di uscita in base al segno del valore
SET PLOTCOLOR1 = IF (PLOT1 > 0, COLOR_GREEN, COLOR_RED)
```

Anche l'Indicator, così come la Function, presenta tre parametri d'ingresso da dichiarare utilizzando la stringa INPUTS:

@price: prezzo sul quale calcolare le Moving Averages

@shortPeriods: lunghezza della EMA di breve periodo

@longPeriods: lunghezza della EMA di lungo periodo

*Nota: i valori di uscita degli Indicator devono essere assegnati alle variabili PLOT1, ... , PLOT8; i colori dei valori di uscita degli Indicator devono essere assegnati alle variabili PLOTCOLOR1, ... , PLOTCOLOR8.*

*Ogni Indicator pertanto può avere fino ad un massimo di otto valori di uscita.*

Negli Indicator si possono verificare le seguenti condizioni alle quali assegnare colori differenti:

1. sopra o sotto un valore definito (o zero)

```
SET PLOTCOLOR1 = IF(PLOT1 > @value, COLOR_GREEN, COLOR_RED)
```

2. indicatore in salita o in discesa

```
SET SALITA = PLOT1 > REF(PLOT1, 1)  
SET PLOTCOLOR1 = IF(SALITA, COLOR_GREEN, COLOR_RED)
```

3. indicatore in salita sopra un valore definito (o zero) o in discesa sotto un valore definito (o zero)

```
# Calcolo condizione di salita  
SET COND_SALITA = PLOT1 > @value AND PLOT1 > REF(PLOT1, 1)  
# Calcolo condizione di discesa  
SET COND_DISCESA = PLOT1 < @value AND PLOT1 < REF(PLOT1, 1)  
# Imposto colore in base al superamento del valore definito  
SET PLOTCOLOR1 = IF(PLOT1 > @value, COLOR_GREEN, COLOR_RED)  
# Se la condizione di salita e' vera, metto un colore piu' chiaro  
SET PLOTCOLOR1 = IF(COND_SALITA, COLOR_LIGHT_GREEN, PLOTCOLOR1)  
# Se la condizione di discesa e' vera, metto un colore piu' chiaro  
SET PLOTCOLOR1 = IF(COND_DISCESA, COLOR_LIGHT_RED, PLOTCOLOR1)
```

### ***Creazione di proprie Condition***

Una Condition è semplicemente una espressione che rappresenta una condizione, il cui valore può essere esclusivamente TRUE o FALSE.

Le Condition vengono utilizzate in beeTrader® per creare Alerts e in Stock Scanner.

Esempio: continuando nell'esempio dei paragrafi precedenti, creiamo una Condition per verificare il segno del valore di uscita della Function beeMomentum.

```
INPUTS: @price(CLOSE), @shortPeriods(7), @longPeriods(14)  
beeMomentum (@price, @shortPeriods, @longPeriods) > 0
```

La Condition restituisce TRUE quando "beeMomentum" è positiva, ovvero lo strumento a cui è stata applicata è in TREND di salita.

### ***Creazione di propri Expert Advisor***

Un Expert Advisor permette all'utente di realizzare dei commenti "intelligenti" in modo automatico al verificarsi di determinate condizioni. Ogni Expert Advisor è costituito da due condizioni, BUY e SELL e da un messaggio di testo personalizzabile.

Esempio: continuando nell'esempio dei paragrafi precedenti, creiamo un Expert Advisor che visualizzi un messaggio di testo articolato in base al segno del valore di uscita della Function beeMomentum.

BUY Script

```
INPUTS: @price(CLOSE), @shortPeriods(7), @longPeriods(14)  
beeMomentum (@price, @shortPeriods, @longPeriods) > 0
```

Sell Script

**beeMomentum (@price, @shortPeriods, @longPeriods) < 0**

Message

**Risultato della Function beeMomentum: TREND IN [SALITA|DISCESA]**

L'Expert Advisor al verificarsi della condizione BUY Script visualizza:

Risultato della Function beeMomentum: TREND IN SALITA

L'Expert Advisor al verificarsi della condizione BUY Script visualizza:

Risultato della Function beeMomentum: TREND IN DISCESA

*Nota: nel campo Message degli Expert Advisor è possibile fare in modo che beeTrader® visualizzi un messaggio di testo diverso a seconda di quale delle due condizioni sia verificata. Per fare ciò i due testi alternativi devono essere racchiusi tra [] e separati tra loro dal carattere |.*

### **Creazione di propri Signal**

Un Signal permette di specificare a quali condizioni inviare ordini, in paper trading o in real market. Un Signal è costituito almeno da due condizioni, Buy Script e Sell Script, mentre le altre due condizioni Exit Long ed Exit Short sono opzionali. E' inoltre possibile aggiungere funzionalità di Money Management impostando le variabili relative.

Esempio: continuando nell'esempio dei paragrafi precedenti, creiamo un Signal che generi un ordine BUY o SELL in base al segno del valore di uscita della Function beeMomentum.

BUY Script

**INPUTS: @price(CLOSE), @shortPeriods(7), @longPeriods(14)**

**beeMomentum (@price, @shortPeriods, @longPeriods) > 0**

Sell Script

**beeMomentum (@price, @shortPeriods, @longPeriods) < 0**

Questo Signal genera un segnale BUY quando beeMomentum è positivo, ovvero quando il TREND è in salita; al contrario genera un segnale SELL quando beeMomentum è negativo, ovvero quando il TREND è in discesa.

## Esempi di Trading System & Tecniche

### **Trading Systems**

Un Trading System altro non è che un insieme di regole che vengono applicate all'andamento di un determinato strumento finanziario.

Queste regole includono quando è stato deciso di entrare a mercato comperando lo strumento, quando venderlo, quando venderlo Short (guadagnare prevedendo un ribasso del titolo), quando uscire dalla posizione che si è precedentemente assunta.

Uscire da una posizione può essere determinato da delle regole basate su indicatori oppure su regole di money management.

In qualsiasi caso, all'avvenire della condizione stabilita avverrà l'evento che abbiamo abbinato alla condizione.

Il limite delle condizioni tali per cui deve avvenire uno degli eventi possibili (Buy, Sell, ExitLong, ExitShort, TakeProfit, StopLoss, Target) è limitato solo dalla fantasia e dall'esperienza del trader. Per portare a termine il progetto di un Trading System, oltre alla scrittura delle regole, bisogna affrontare altri tre passaggi:

#### Backtest

Costruito il sistema, si deve verificare su una serie di dati storici per valutarne le molteplici caratteristiche, potrebbe per esempio aver guadagnato ma in un solo trade e questo evento potrebbe non ripetersi, oppure aver generato una equity line in perdita, ecc..

#### Optimizer

Poi si passa alla fase di ottimizzazione, ovvero si definiscono dei range di valori entro i quali beeTrader® calcolerà quale tra questi avrebbe generato la migliore performance.

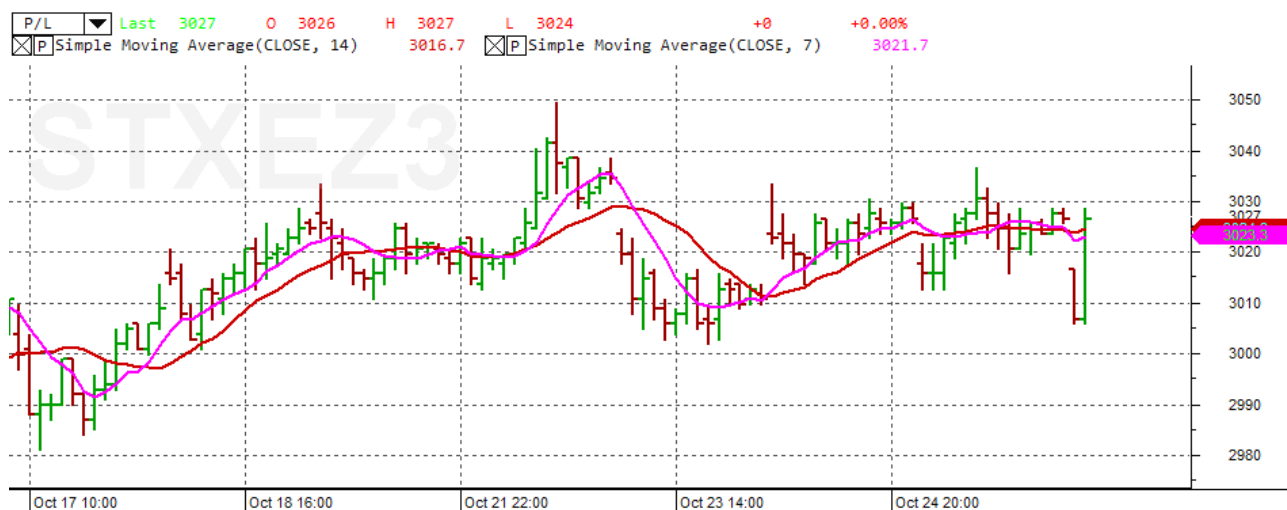
#### Paper Trading

Una volta che abbiamo scelto i parametri dobbiamo fare un ultimo passo che è quello di mandare il sistema che abbiamo costruito, in avanti nel tempo. Con dati nuovi, in tempo reale. Per fare questo si sceglie la funzione di "Strategy" e si seleziona il "Paper Trading" che farà l'esame al sistema con dati reali ma senza impiegare denaro.

### **bee Moving Average Crossover**

Il Moving Average Crossover è forse il più semplice di tutti i trading systems.

Questo sistema utilizza due Moving Averages per generare segnali. Un segnale BUY viene generato quando la Moving Average di breve periodo incrocia al rialzo la Moving Average di lungo periodo, un segnale SELL viene generato quando la Moving Average di breve periodo attraversa al ribasso la Moving Average di lungo periodo.



Il numero di segnali generati da questo trading system è proporzionale alla lunghezza e al tipo di Moving Averages che vengono usate. Moving Averages di breve periodo generano più segnali di Moving Averages di lungo periodo.

Moving Averages calcolate su di un periodo breve generano più segnali di Moving Averages calcolate su di un periodo lungo.

Qui nasce il primo vero problema che poi si ritroverà in moltissimi sistemi, ovvero, utilizzare periodi di calcolo corti generano molti segnali che però, proprio per la frequenza con cui vengono calcolati, potrebbero essere anche errati o, come si dice nel trading, "falsi".

La soluzione più immediata a questa sorta di mediazione tra la precisione e la frequenza, la si può ottenere utilizzando nello stesso sistema diversi indicatori. Facendo eseguire le entrate ad alcuni e le uscite a quelli costruiti proprio per questo scopo.

Nell'esempio vediamo come abbinare il segnale di entrata generato dall'incrocio di due Exponential Moving Averages (EMA) a 20 e 60 periodi con il segnale di uscita dal trade, generato dall'indicatore Parabolic SAR.

### Buy Script

# EMA a 20 periodi incrocia al rialzo EMA a 60 periodi  
**CROSSOVER (EMA (CLOSE, 20), EMA (CLOSE, 60))**

## Sell Script

# EMA a 20 periodi incrocia al ribasso EMA a 60 periodi  
**CROSSOVER (EMA (CLOSE, 60), EMA (CLOSE, 20))**

## Exit Long Script

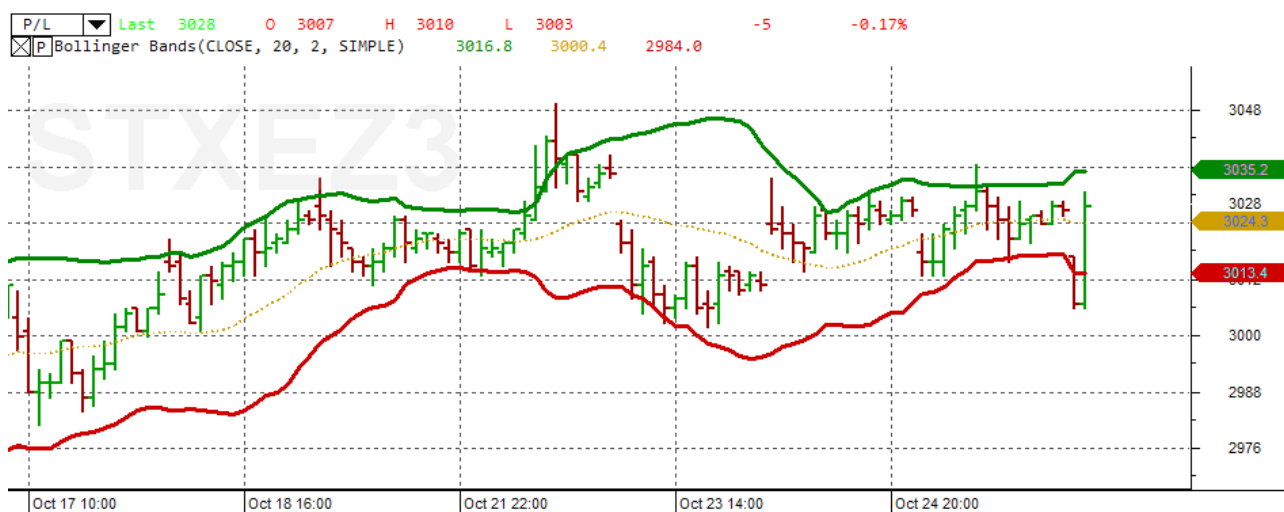
# CLOSE incrocia al ribasso Parabolic SAR  
**CROSSOVER (PSAR (0.02, 0.2), CLOSE)**

## Exit Short Script

# CLOSE incrocia al rialzo Parabolic SAR  
**CROSSOVER (CLOSE, PSAR (0.02, 0.2))**

## bee Bollinger Bands

Le Bollinger Bands sono simili ad una Moving Average eccetto che sono traslate sopra e sotto il prezzo di un certo numero di deviazioni standard per formare una banda intorno al prezzo. E a differenza di una Simple Moving Average o una Moving Average Envelope, le Bollinger Bands sono calcolate in modo tale da permettere ampliamenti e contrazioni basate sulla volatilità del mercato.



I prezzi in genere rimangono contenuti all'interno delle bande. Una strategia è quella di acquistare o vendere quando i prezzi toccano le bande e poi ritracciano perchè un movimento che ha origine in una banda di solito tende a muoversi all'interno della stessa banda.

Un'altra strategia è quella di acquistare o vendere se il prezzo va al di fuori delle bande. In questo caso, il mercato è destinato a proseguire in questa direzione per un certo periodo di tempo.

Il trading system sulle Bollinger Bands descritto in questo esempio utilizza una combinazione di entrambe le strategie. Il sistema genera un BUY se una barra recente ha toccato la banda inferiore e la barra corrente è all'interno bande, e genera un BUY anche se il prezzo HIGH corrente ha superato la fascia superiore di una certa percentuale. Il sistema genera un SELL nelle condizioni opposte.



## Buy Script

```
# Il CLOSE era sotto la banda inferiore ed ora è sopra
SET Bottom = BBB (CLOSE, 20, 2, EXPONENTIAL)
SET Top = BBT (CLOSE, 20, 2, EXPONENTIAL)
((REF (CLOSE, 1) < REF (Bottom, 1)) AND
CLOSE > Bottom) OR
# Il CLOSE è superiore alla banda superiore del 2%
CLOSE > Top * 1.02
```

## Sell Script

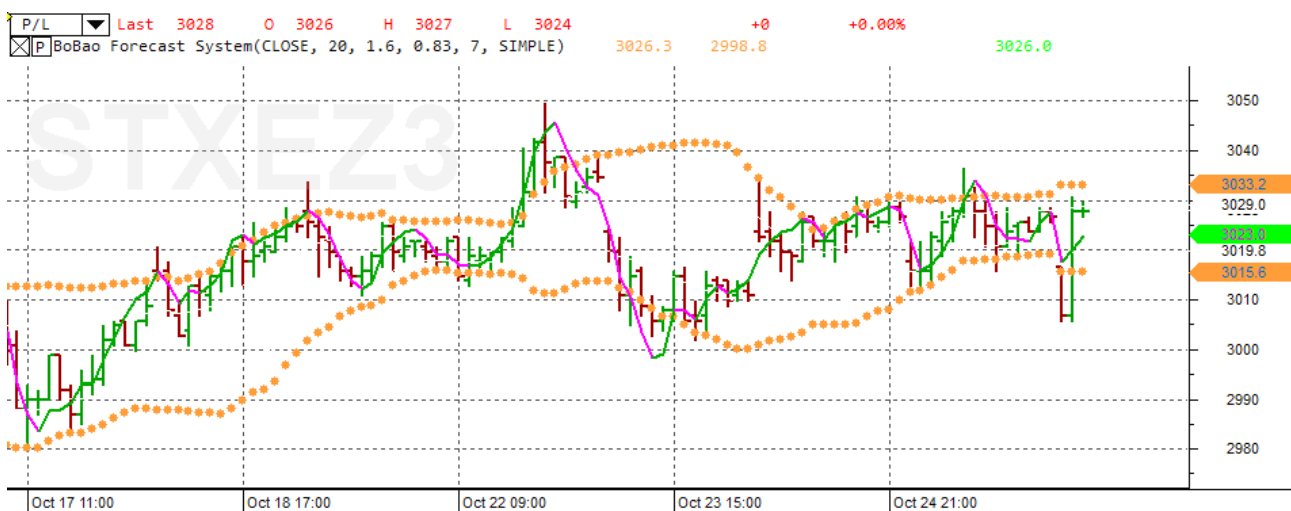
```
# Il CLOSE era sopra la banda superiore ed ora è sotto
SET Bottom = BBB (CLOSE, 20, 2, EXPONENTIAL)
SET Top = BBT (CLOSE, 20, 2, EXPONENTIAL)
((REF (CLOSE, 1) > REF (Top, 1)) AND
CLOSE < Top) OR
# Il CLOSE è inferiore alla banda inferiore del 2%
CLOSE < Bottom * 0.98
```

## Bobao Forecast System

Questo sistema, ideato da Cagalli Tiziano, si basa sull'utilizzo di due coppie di Bollinger Bands a tarature diverse e dall'utilizzo di una Linear Regression.

La Linear Regression genererà un segnale all'attraversamento di una delle due bande della coppia con deviazione standard maggiore.

Il passaggio della linea di regressione sulla seconda coppia di bande con deviazione standard minore, servirà come possibile punto di uscita nel caso non si utilizzino sistemi di money management.



Nel disegno le bande di ingresso sono disegnate a punti e quando la Linear Regression attraversa dall'alto verso il basso la banda superiore verrà generato un segnale SELL. L'attraversamento al contrario genererà un segnale ExitShort. Quando la Linear Regression attraverserà dal basso verso l'alto la banda

inferiore verrà generato un segnale BUY che verrà stoppato da un segnale ExitLong nel caso di attraversamento contrario ovvero quando la linea di regressione attraverserà la banda inferiore dall'alto verso il basso.

#### Buy Script

```
# Definiamo le variabili
INPUTS: @price(CLOSE), @BandPeriods(20, 10, 30, 1), @BigDev(1.6, 1, 3, 0.1),
@SmalDev(0.83, 0.3, 0.9, 0.03), @matype(SIMPLE), @SLperiods(7)

SET BigTop = BollingerBandsTop(@price, @BandPeriods, @BigDev, @matype)
SET BigBottom = BollingerBandsBottom(@price, @BandPeriods, @BigDev, @matype)
SET SmallTop = BollingerBandsTop(@price, @BandPeriods, @SmalDev, @matype)
SET SmallBottom = BollingerBandsBottom(@price, @BandPeriods, @SmalDev,
@matype)
SET SignalLine = LR(@price, @SLperiods)

# Primo Buy
CROSSOVER(SignalLine, BigBottom)
```

#### Sell Script

```
# Definiamo le variabili
SET BigTop = BollingerBandsTop(@price, @BandPeriods, @BigDev, @matype)
SET BigBottom = BollingerBandsBottom(@price, @BandPeriods, @BigDev, @matype)
SET SmallTop = BollingerBandsTop(@price, @BandPeriods, @SmalDev, @matype)
SET SmallBottom = BollingerBandsBottom(@price, @BandPeriods, @SmalDev,
@matype)
SET SignalLine = LR(@price, @SLperiods)

# Primo Sell
CROSSOVER(BigTop,SignalLine)
```

#### ExitLong Script

```
# Definiamo le variabili
SET BigTop = BollingerBandsTop(@price, @BandPeriods, @BigDev, @matype)
SET BigBottom = BollingerBandsBottom(@price, @BandPeriods, @BigDev, @matype)
SET SmallTop = BollingerBandsTop(@price, @BandPeriods, @SmalDev, @matype)
SET SmallBottom = BollingerBandsBottom(@price, @BandPeriods, @SmalDev,
@matype)
SET SignalLine = LR(@price, @SLperiods)

# Uscita ExitLong
CROSSOVER(SmallBottom,SignalLine)
OR CROSSOVER(SmallTop,SignalLine)
OR CROSSOVER(BigBottom,SignalLine)
```

#### ExitShort Script

```
# Definiamo le variabili
SET BigTop = BollingerBandsTop(@price, @BandPeriods, @BigDev, @matype)
SET BigBottom = BollingerBandsBottom(@price, @BandPeriods, @BigDev, @matype)
SET SmallTop = BollingerBandsTop(@price, @BandPeriods, @SmalDev, @matype)
SET SmallBottom = BollingerBandsBottom(@price, @BandPeriods, @SmalDev,
@matype)
SET SignalLine = LR(@price, @SLperiods)

# Uscita ExitShort
```

**CROSSOVER(SignalLine, SmallTop)**  
**OR CROSSOVER(SignalLine, SmallBottom)**  
**OR CROSSOVER(SignalLine, BigTop)**

Potete notare che in questo modo ci sono due condizioni per l'entrata e quattro per l'uscita: le entrate solo dall'attraversamento delle bande superiori, le uscite dall'attraversamento delle stesse bande ma in senso contrario oppure delle bande differenti.

## **bee Commodity Channel Index**

Donald Lambert ha sviluppato l'indicatore CCI. Anche se lo scopo di questo indicatore è quello di individuare turni ciclici di Commodities, è spesso utilizzato per i titoli. E' un oscillatore che si sviluppa tra i valori attorno al 100 e -100 nel 70/80% dei casi. Più il periodo sarà corto e più aumenteranno i valori maggiori di 100.



Costruiamo un semplice ma efficace sistema basato sull'attraversamento delle linee di marcatura delle zone di ipercomprato e di ipervenduto.

Se il CCI attraverserà il livello di ipervenduto dal basso verso l'alto il sistema genererà un BUY. Il SELL è generato dalla situazione contraria ovvero dall'attraversamento del CCI dall'alto verso il basso della linea di marcatura dell'ipercomprato.

### Buy Script

```
# Definiamo le variabili
INPUTS: @periods(21), @matype(SIMPLE), @lowMark(-80), @highMark(80)
SET C = CCI(@periods, @matype)
SET cond = CROSSOVER(C, @lowMark)
#PRINT(C)
#PRINT(cond)

cond
```

## Sell Script

```
# Variabili già definite in Buy Script  
SET C = CCI(@periods, @matype)  
CROSSOVER(@highMark, C)
```

### **bee Swing Trailing**

Questo sistema si basa sugli swing, ovvero sui cambiamenti di trend per generare i segnali di ingresso e di uscita.

Vengono però aggiunti gli ingredienti di money management in modo tale che l'utente possa apprezzare la validità di gestire il denaro durante la durata di un trade.

Utilizziamo il Traling Stop che definisce l'obbiettivo di primo guadagno, punto da cui si metterà a rischio una parte del medesimo definita dal Trailing percent. In questo modo, se si entra in un trend duraturo, si potrà rimanere in posizione più a lungo essendosi assicurati un guadagno (T\_Stop) e mettendone a rischio una parte percentuale(T\_percent) che permetterà di rimanere in posizione in caso di leggere oscillazioni di prezzo.

In caso di entrate errate che non portino subito al Traling Stop ma che invece iniziano a generare delle perdite ecco che si utilizza lo Stop Loss che ci farà uscire dalla posizione con una perdita pari a quanto abbiamo fissato come regola.

### **Il sistemista Consiglia: Money Management**

- MAX\_POSITION\_OPEN il suo utilizzo diventa performante se combinato (AND) con una condizione di prezzo, ovvero: usciamo dalla posizione se dopo X periodi il prezzo di entrata corrisponde circa al prezzo LAST.  
SET MAX\_POSITION\_OPEN = 20 AND LASTENTRYPRICE < LAST \* 1,05 OR LASTENTRYPRICE > LAST \* 0,95
- STOP\_LOSS è la chiusura della posizione ma anche la materializzazione della perdita, va utilizzato dopo un'attenta analisi del MAX INTRADAY DRAWDOWN (vedi beeAnalyzer) e posizionato ad un valore che annulli meno trades possibili. Più lo STOP\_LOSS è stretto e più il sistema perderà. In un Trading System di tipo STOP & REVERSE chiaramente dovrà essere posizionato molto al di sotto della normale entrata STOP & REVERSE perchè deve servire esclusivamente come emergenza se altre cose non hanno funzionato
- TAKE\_PROFIT è una funzione da usare in multi trades ovvero quando il Trading System è applicato a più sottostanti. Il senso è di entrare su un

sottostante per breve tempo, raggiungere l'obiettivo e poi spostarsi su di un altro sottostante

- TRAILING\_STOP & TRAILING\_PERCENT sono due funzioni che si usano assolutamente in contemporanea e principalmente su Trading System che sono stati progettati per generare segnali non su incroci di segnali ma condizioni che si prolungano (se scriviamo di comprare e vendere all'incrocio di due Moving Averages significherà che il Trading System invierà pochi ordini, ovvero ad ogni incrocio e quindi il TRAILING\_STOP ci farebbe uscire e non più rientrare. Viceversa se scriviamo di comprare e vendere se la MA è sopra o sotto ad un'altra MA il Trading System genererà continui segnali di entrata ad un'uscita del TRAILING\_STOP. Attenzione il settaggio del TRAILING\_PERCENT non deve essere tale per cui una variazione di un tick sia superiore al suo valore (DAX: 1 punto = 25 €, movimento minimo 12,5€, settaggio corretto TRAILING\_STOP = 250 e TRAILING\_PERCENT = 10. Settaggio errato TRAILING\_STOP = 125 e TRAILING\_PERCENT = 15) infatti il valore del TRAILING\_PERCENT a cui uscirà raggiunta la soglia di 125€ sarà superiore al movimento minimo, il 15% di 125 = 18,75€ > 12,5€.

EasyScript® è stato sviluppato da  
PlayOptions SRL,  
con sede in viale Tre Martiri 65/Q - 45100 Rovigo - Italy  
e viene concesso in licenza OEM al distributore commerciale di beeTrader®