

Easy Language



Referencing Data from Previous Bars

ערכי מחיר קודמים/היסטוריים וערכים אחרים ניתנים לגישה באמצעות אינדקס בסוגריים מרובעים [n] המוצמדים לשם המשתנה או הפרמטר (מילה השמורה) של הערך להפניה.

The close of the bar 5 bars ago:

Close[5]

Range of the bar 12 bars ago:

High[12] - Low[12] or (High - Low)[12]

but not High - Low[12] !!

Ex-07 Net Change Oscillator

Write an Indicator that plots a 5-bar Net Change (close to close) Oscillator.

Ex-08 Bands

Write an Indicator that plots two bands around the price data.

- ❑ One band adds half the previous bar's range to the current bar's opening.
- ❑ The other band subtracts half the previous bar's range from the current bar's opening.
- Remember: The range of a bar is $\text{High} - \text{Low}$.
- This Indicator should overlay the price bars

משתנים - Variables

- ▶ משתנים הינם כלי תכנותי המאפשרים לשמור ערכי ביניים המתקבלים במהלך החישוב ולהתייחס לערכים אלה בעת הצורך
- ▶ לכל יש משתנה שם ייחודי
- ▶ משתנים יכולים להיות משתנים מספריים, משתנים לוגיים (True/False), או משתנה מחרוזת-טקסט.
- ▶ שימוש במשתנים מאפשר לארגן את הקוד על ידי מתן למשתנים שמות משמעותיים המתארים את אופי החישוב או המטרה של נתונים/ערכים מאוחסנים.

משתנים - Variables

Variables:

LastHigh(0),	{ creates a numeric variable }
NewHigh(false),	{ creates a true/false variable }
HighAlert("");	{ creates a text variable }

השמה למשתנים - Variables Assignment

Vars:

SlowAverage(0), FastAverage(0), MyParam (0);

MyParam = 2;

MyParam = (SlowAverage + FastAverage)/2;

SlowAverage = Average(Close, 18);

FastAverage = Average(Close, 9);

סוגי משתנים - Variables Types

There are three variable, input, and array types:

- ❑ **Numeric** – Holds a simple or complex number, positive or negative. There are three numeric types in EasyLanguage: **Integer**, **Float**, and **Double**.
- ❑ **True/false** – Holds a true/false state, either a true/false expression or the words true or false.
- ❑ **Text String** – Holds a text string, numbers or letters enclosed in quotation marks ("text").

Pre Declared Variables

- ▶ Pre-declared variables are variables that are do not need to be declared and automatically recognized by the language.
- ▶ Pre-declared variables come in two types: numeric and true/false.

- ▶ **Value0 to Value99** – There are 100 pre-declared numeric variables.

Value1 = Average(Price, Length);

- ▶ **Condition0 to Condition99** – There are 100 pre-declared true/false variables.

Condition1 = Value1 >= MyAverage;

Historical Reference of Variables

- ▶ משתנים, פונקציות, ומילות שמורות ניתן להתייחס לערכי ברים קודמים/היסטוריים של עצמם בגרף, באמצעות הוספת הסוגר מרובע [n] לאחר שם המשתנה
- ▶ ניתן לפנות באותו אופן, על ידי שימוש ב [n], לערכים היסטוריים באלמנטי מערך.

דוגמאות:

- ▶ Plot1(Close[5]);
- ▶ Plot2((Close – Open)[1]);
- ▶ Condition1 = Condition2[1] OR Condition2[2];

Ex-09 Bands2

This is a re-write *08 Bands, incorporating user-declared numeric variables.

Write an Indicator that plots two bands around the price data.

- ❑ One band adds half the previous bar's range to the current bar's opening.
- ❑ The other band subtracts half the previous bar's range from the current bar's opening.

Ex-10 Net Change Oscillator-2

This is a re-write of *07 NetChangeOsc, incorporating user-declared numeric variables

Write an Indicator that plots a 5-bar Net Change Oscillator.

FOR לולאת

```
for indx=<init_value> to <final_value> begin
    .
    . { loop body...}
    .
end;
```

- ▶ A for loop repeats one or more statements a specific number of iterations defined by the user in the for statement loop range values.
- ▶ This numeric range is incremented in a loop counter stored in a variable for reference within the block of statements.

FOR לולאת

- ▶ Once the counter reaches the specified limit, the for loop ends and the next statement in the code is evaluated.
- ▶ For loops can iterate the count either ascending or descending depending on which reserved word is used:
 - To** – count ascending
 - DownTo** – count descending

FOR לולאת

Example (ascending):

```
Vars: RangeSum(0), x(0);{ Declare Variables }
RangeSum = 0; { Reset variable each bar}
```

```
For x = 0 to 5 begin
    RangeSum = RangeSum + Range[x];
end;
```

This for loop will iterate through the statement block 6 times (0, 1, 2, 3, 4, 5).

FOR לולאת

Example (descending):

```
Vars: RangeSum(0), x(0);{Declare Variables}
RangeSum = 0; {Reset variable each bar}
```

```
For x = 5 downto 0 begin
    RangeSum = RangeSum + Range[x];
end;
```

This for loop will iterate through the statement block 6 times (5, 4, 3, 2, 1, 0).

שימוש בפונקציות – Using Functions

פונקציות הינן מבנה תוכנה הממשות חישוב/נוסחא והמחזירות ערך מחושב.

הפונקציות נקראות לשימוש בטכניקת ניתוח או באסטרטגיה באמצעות מילות מפתח – שם הפונקציה ומבטל את הצורך ליצור כל פעם מחדש נוסחאות מורכבות

הפונקציות הנפוצות ביותר בשפה הינן חישוב ממוצע, סטיית תקן, RSI, ו ADX

פונקציות רבות דורשות מהמשתמש לספק פרמטרים לפונקציה

שימוש בפונקציות – Using Functions

- ▶ Functions are frequently used formulas (or comparisons) that return numeric values (or true or false). They may be called for use in any Analysis Technique or Strategy with just a few words,
- ▶ eliminating the need to re-create complex formulas.
- ▶ • Functions can be used in any set of EasyLanguage rules or instructions.
- ▶ • Most common mathematical and statistical formulas are stored as Functions, such as

Ex-11 Momentum

Write an Indicator that plots the 10-bar momentum of closing prices.

Ex-12 Real Body Average

Write an Indicator that plots the 10-bar average of the difference between the Close and the Open of each bar, as originally plotted in *04 Real Body.

Ex-13 Envelope

Write an Indicator that plots a 20-bar moving average of the highs and a 20-bar moving average of the lows.

Select names and declare variables for the two moving average values.

Hint: The "Average" function may be dragged in from the Dictionary.

פרמטרי קלט - Inputs

- הצהרת פרמטרי Input בטכניקת ניתוח או אסטרטגיה מוסיפה פרמטרי קלט שבאמצעותם ניתן להזין ערכי קלט משתנים, ומאפשר בכך לשנות את אופי/נתוני החישוב מבלי לשנות את קוד ה-EasyLanguage
- שימוש בפרמטרי Input מאפשר ליצור פונקציות גנריות, ומאפשר לבחון בקלות ביצוע החישוב על סצנריו שונים מבלי לשנות את קוד ה-EasyLanguage
- פרמטרי Input משמשים להזנת ערכי קלט לחישוב, בקרה וקביעת אופן ביצוע החישוב, קביעת אופן הצגה ומיפוי הצבעים, אופטימיזציה של אסטרטגיות, ועוד...

פרמטרי קלט – Inputs

הצהרה על פרמטרי Input :

`Input: InputName(default value);`

- לכל פרמטר Input ניתן ערך ברירת מחדל (מספרי, (text, True/False, המגדיר את טיפוס המשתנה.
- לא ניתן לשנות את ערכי ה Input מתוך הקוד של טכניקת ניתוח או אסטרטגיה, אלא מתוך חלון **Format Analysis Techniques and Strategies**
- דוגמא:

`Input: Factor(1.005);`

`Input: Price(Close);`

Ex-14 Envelope2

Learning objective: Using Inputs in place of fixed values.

Re-write Indicator 13 incorporating Inputs for the lengths of the moving averages.

Ex-15 Trailing Hi Lo

Learning objective: Using Inputs; using the functions Highest and Lowest.

Write an Indicator that plots the highest value of an Input and the lowest value of an Input over the last n bars.

Relational Operators

משמשים לשערוך ביטויי True/False ☐
 EasyLanguage ישנם שמונה Related Operators: ☐

- < Less than
- <= Less than or equal to
- > Greater than
- >= Greater than or equal to
- = Equal to
- <> Not equal to