

## الباب الثالث

# المتغيرات المعروفة مسبقاً

انطلقوا إلى القمة



• **Ask** سعر الطلب

• **Bars** القضبان

• **Bid** سعر البيع

• **Close** سعر الاغلاق

• **Digits** الارقام العشرية

• **High** القمة

• **Low** القاع

• **Open** سعر الافتتاح

• **Point** النقطة

• **Time** الوقت

• **Volume** الزخم

## تعريف أولى

يوجد داخل كل برنامج قابل للتنفيذ مصمم بواسطة لغة MQL4 عدد من المتغيرات المعرفة مسبقاً المدعومة لتعكس حالة الشارت ( الرسم البياني ) عند بدء تشغيل البرنامج سواء كان خبير ( اكسبيرت ) او سكربت او مؤشر خاص

تستخدم المكتبات **Libraries** المتغيرات

ولكى يكون هناك وصول امن وسريع لهذه البيانات ، توفر منصة العميل نسخ داخلية من المتغيرات المعرفة مسبقاً لكل برنامج على حده ويتم تحديث هذه البيانات فى كل تشغيل

FX Arabia  
إف إكس أرابيا

## سعر الطلب Ask

وهو آخر سعر مقدم من البائع ( سعر الطلب ) للزوج الحالى او السلعة الحالية  
محدث باستمرار

انطلق إلى القمة

مثال :

```
if(iRSI(NULL,0,14,PRICE_CLOSE,0)<25)
{
  OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-StopLoss*Point,Ask+TakeProfit*Point,
    "My order #2",3,D'2005.10.10 12:30',Red);
  return;
}
```

FX Arabia  
إف إكس أرابيا

## القضبان Bars

عدد القضبان ( البديل عن الشموع البيانية ) على الشارت

مثال :

الخطة إلى القمة

```
int counter=1;  
for(int i=1; i<=Bars; i++)  
{  
    Print (Close[i-1]);  
}
```



## سعر البيع Bid

وهو آخر سعر مقدم من المشتري ( سعر العرض ) للزوج الحالى او السلعة الحالية  
محدث باستمرار

انطلقوا إلى القمة

مثال :

```
if (iRSI (NULL, 0, 14, PRICE_CLOSE, 0) > 75)
{
    OrderSend ("EURUSD", OP_SELL, Lots, Bid, 3, Ask + StopLoss * Point, Bid - TakeProfit * Point,
        "My Order #2", 3, D'2005.10.10 12:30', Red);
    return (0);
}
```

FX Arabia  
إف إكس أرابيا

## سعر الاغلاق Close

وهو مجموعة مسلسلة تحتوى على أسعار الاغلاق لكل شمعة ( أو واحد من القضبان )  
للشارت الحالى

يتم فهرسة عناصر المصفوفة هذه فى ترتيب عكسى اى من اخر عنصر لأول عنصر  
ويتم فهرسة أقدم بار ( أو شمعة حيثما كان ) بالقيمة ١ .. ويتم فهرسة اخر بار أى  
احدث بار بالقيمة ٠

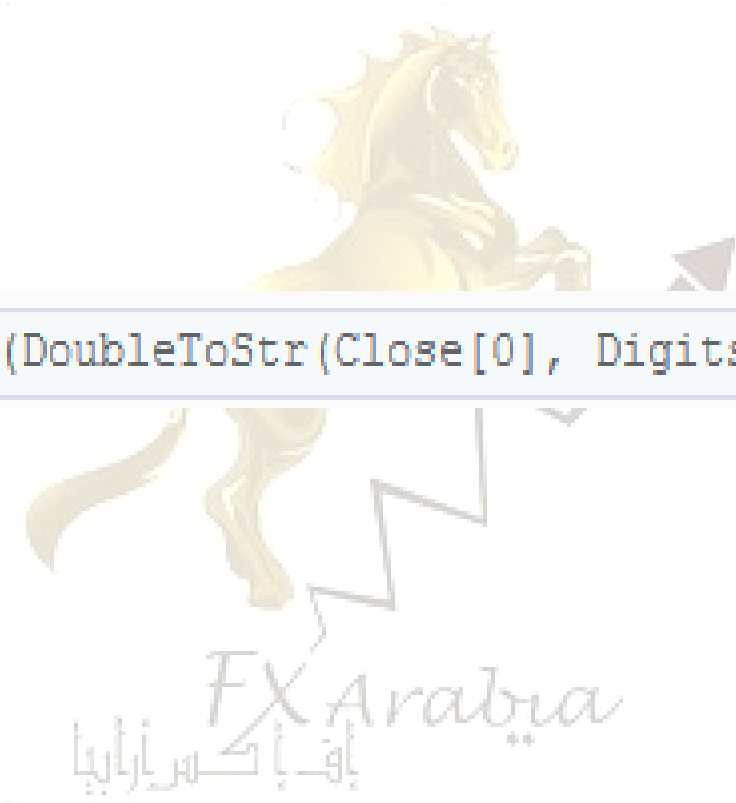
مثال :

```
int handle = FileOpen("file.csv", FILE_CSV|FILE_WRITE, ";");
if(handle>0)
{
    // table column headers recording
    FileWrite(handle, "Time;Open;High;Low;Close;Volume");
    // data recording
    for(int i=0; i<Bars; i++)
        FileWrite(handle, Time[i], Open[i], High[i], Low[i], Close[i], Volume[i]);
    FileClose(handle);
}
```

## الارقام العشرية Digits

وهي عدد الأرقام العشرية بعد الفاصلة ( النقطة العشرية ) للزوج الحالى أو السلعة الحالية

انطلقوا إلى القمة



مثال :

```
Print(DoubleToStr(Close[0], Digits));
```

## القمة High

وهو مجموعة مسلسلة تحتوى على أعلى أسعار وصلت لها الشموع ( أو واحد من القضبان ) للشارت الحالى

يتم فهرسة عناصر المصفوفة هذة فى ترتيب عكسى اى من اخر عنصر لأول عنصر ويتم فهرسة أقدم بار ( أو شمعة حيثما كان ) بالقيمة ١ .. ويتم فهرسة اخر بار أى احدث بار بالقيمة ٠

مثال :

```
//---- maximums counting
i=Bars-KPeriod;
if(counted_bars>KPeriod) i=Bars-counted_bars-1;
while(i>=0)
{
double max=-1000000;
k = i + KPeriod-1;
while(k>=i)
{
price=High[k];
if(max<price) max=price;
k--;
}
HighesBuffer[i]=max;
i--;
}
//----
```



## القاع Low

وهو مجموعة مسلسلة تحتوى على أقل أسعار وصلت لها الشموع ( أو واحد من القضبان ) للشارت الحالى

يتم فهرسة عناصر المصفوفة هذه فى ترتيب عكسى اى من اخر عنصر لأول عنصر ويتم فهرسة أقدم بار ( أو شمعة حيثما كان ) بالقيمة ١ .. ويتم فهرسة اخر بار أى احدث بار بالقيمة ٠

مثال :

```
//---- minima counting
i=Bars-KPeriod;
if(counted_bars>KPeriod) i=Bars-counted_bars-1;
while (i>=0)
{
    double min=1000000;
    k = i + KPeriod-1;
    while (k>=i)
    {
        price=Low[k];
        if(min>price) min=price;
        k--;
    }
    LowesBuffer[i]=min;
    i--;
}
//----
```

## سعر الافتتاح Open

وهو مجموعة مسلسلة تحتوى على الاسعار الحالية المفتوحة وصلت لها الشموع للشارت الحالى

يتم فهرسة عناصر المصفوفة هذه فى ترتيب عكسى اى من اخر عنصر لأول عنصر ويتم فهرسة أقدم بار ( أو شمعة حيثما كان ) بالقيمة ١ .. ويتم فهرسة اخر بار أى احدث بار بالقيمة ٠

مثال :

```
i = Bars - counted_bars - 1;
while (i >= 0)
{
    double high = High[i];
    double low = Low[i];
    double open = Open[i];
    double close = Close[i];
    AccumulationBuffer[i] = (close - low) - (high - close);
    if (AccumulationBuffer[i] != 0)
    {
        double diff = high - low;
        if (0 == diff)
            AccumulationBuffer[i] = 0;
        else
        {
            AccumulationBuffer[i] /= diff;
            AccumulationBuffer[i] *= Volume[i];
        }
    }
    if (i < Bars - 1) AccumulationBuffer[i] += AccumulationBuffer[i + 1];
    i--;
}
```

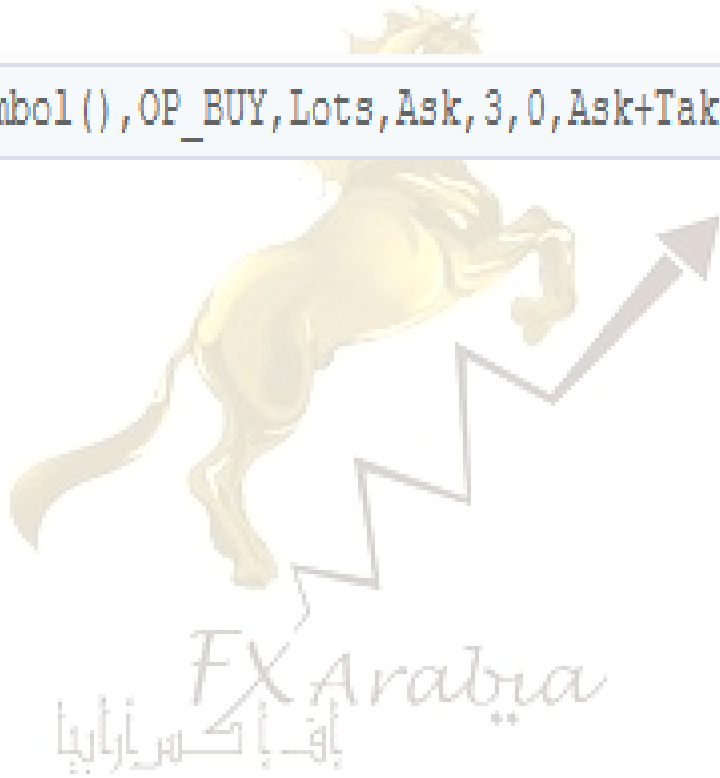
## النقطة Point

وهى قيمة النقطة للزوج الحالى أو السلعة الحالية

مثال :

انطلقوا إلى القمة

```
OrderSend(Symbol(),OP_BUY,Lots,Ask,3,0,Ask+TakeProfit*Point);
```



## الوقت Time

وهو مجموعة مسلسلة تحتوى على وقت الافتتاح لكل بار ( او شمعة ) على الشارت  
الحالى

يتم فهرسة عناصر المصفوفة هذه فى ترتيب عكسى اى من اخر عنصر لأول عنصر  
ويتم فهرسة أقدم بار ( أو شمعة حيثما كان ) بالقيمة ١ .. ويتم فهرسة اخر بار أى  
احدث بار بالقيمة ٠

مثال :

```
for(i=Bars-2; i>=0; i--)
{
    if(High[i+1] > LastHigh) LastHigh = High[i+1];
    if(Low[i+1] < LastLow) LastLow = Low[i+1];
    //----
    if(TimeDay(Time[i]) != TimeDay(Time[i+1]))
    {
        P = (LastHigh + LastLow + Close[i+1])/3;
        R1 = P*2 - LastLow;
        S1 = P*2 - LastHigh;
        R2 = P + LastHigh - LastLow;
        S2 = P - (LastHigh - LastLow);
        R3 = P*2 + LastHigh - LastLow*2;
        S3 = P*2 - (LastHigh*2 - LastLow);
        LastLow = Open[i];
        LastHigh = Open[i];
    }
    //----
    PBuffer[i] = P;
    S1Buffer[i] = S1;
    R1Buffer[i] = R1;
    S2Buffer[i] = S2;
    R2Buffer[i] = R2;
    S3Buffer[i] = S3;
    R3Buffer[i] = R3;
}
```

## الزخم Volume

وهو مجموعة مسلسلة تحتوى على عدد التحركات التى حدثت على البار ( او شمعة )  
على الشارت الحالى

يتم فهرسة عناصر المصفوفة هذة فى ترتيب عكسى اى من اخر عنصر لأول عنصر  
ويتم فهرسة أقدم بار ( أو شمعة حيثما كان ) بالقيمة ١ .. ويتم فهرسة اخر بار أى  
احدث بار بالقيمة ٠

مثال :

```
if(i==0 && time0<i_time+periodseconds)
{
    d_volume += Volume[0];
    if(Low[0]<d_low)    d_low = Low[0];
    if(High[0]>d_high) d_high = High[0];
    d_close = Close[0];
}
last_fpos = FileTell(ExtHandle);
last_volume = Volume[i];
FileWriteInteger(ExtHandle, i_time, LONG_VALUE);
FileWriteDouble(ExtHandle, d_open, DOUBLE_VALUE);
FileWriteDouble(ExtHandle, d_low, DOUBLE_VALUE);
FileWriteDouble(ExtHandle, d_high, DOUBLE_VALUE);
FileWriteDouble(ExtHandle, d_close, DOUBLE_VALUE);
FileWriteDouble(ExtHandle, d_volume, DOUBLE_VALUE);
```

إنتهى الباب الثالث الخاص بالمتغيرات المعرفة مسبقاً

انطلقوا إلى القمة

يُتَّبَع

