

目录

MT4 编程入门 (一): MT4 自定义指标的基本操作

MT4 编程入门 (二): 指标源码中的常用符号说明

MT4 编程入门 (三): 自定义变量

MT4 编程入门 (四): 自定义函数

MT4 编程入门 (五): 自定义数组

MT4 编程入门 (六): 市场数据取值

MT4 编程入门 (七): 判断语句

MT4 编程入门 (八): 循环

MT4 编程入门 (九): MT4 自定义指标的结构

MT4 编程入门 (十): 画一条指标线

MT4 编程入门 (十一): MT4 的报警

MT4 编程入门 (一): MT4 自定义指标的基本操作



MT4 自定义指标的保存目录、指标编辑器与指标调用

一、MT4 自定义指标保存在

这个目录中，

该目录中主要有二种类型的文件，

一种是 mq4 文件，这是 MT4 自定义指标的源码文件，是一种高级语言文件，可以人工编写修改，机器不能直接执行，但能在 MT4 启动时自动编译为 ex4 可执行文件。

另一种就是 ex4 文件，这是 MT4 自定义指标的可执行文件，由 mq4 文件编译得到。

mq4 文件必须编译为 ex4 文件才能被 MT4 执行。

但 ex4 文件是机器码文件，人工不可修改

另外还有一个文件是 mqlcache.dat，它用来保存目录中的文件信息，如果 MT4 无法自动编译，只要删除此文件即可解决问题

mq4 文件编译为 ex4 可执行文件有两种方法：

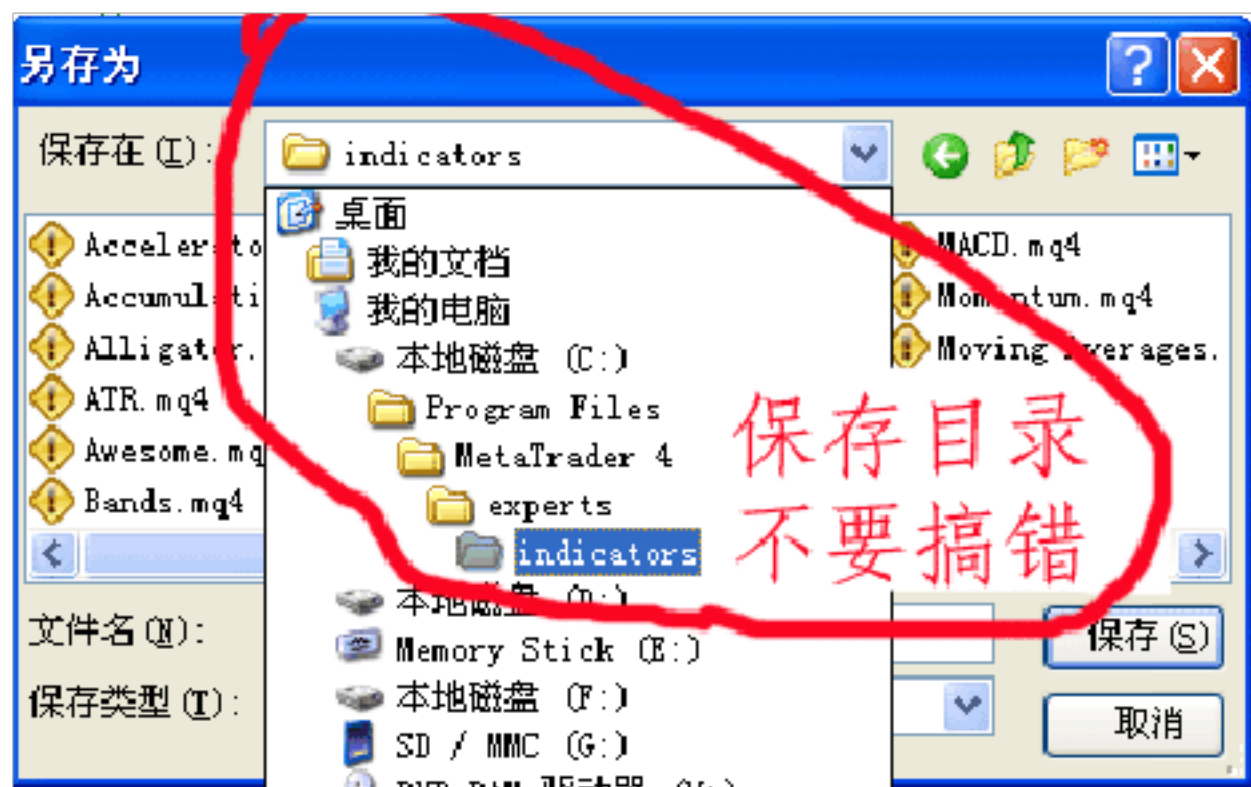
1、MT4 启动时，会自动把此目录中的 mq4 文件编译为 ex4 文件。

2、用 MQ 语言编辑器打开 mq4 文件，点一下“编写”按钮，即可生成 ex4 文件

二、下载保存 mq4 文件、ex4 文件

在论坛上我们经常看到有 mq4 文件、ex4 文件可供下载，我们只要把它们保存到

这个目录，也就是下面这个目录中，就可以了。如果下载的是 ex4 文件，那可以直接调用；如果是 mq4 文件，则需要重新启动 MT4，让系统自动编译生成 ex4 可执行文件。如果下载的是 rar、zip 等类型的压缩文件，还必须下载后解压到此目录才能使用。



三、将源码保存为源码文件

在论坛上，我们也经常能看到指标源码贴在网页上，但没有 mq4、ex4 等文件可以下载，只能复制源码。此时需要我们自己动手将源码保存为源码文件，那也没关系，非常地简单，多点几下鼠标就搞定。

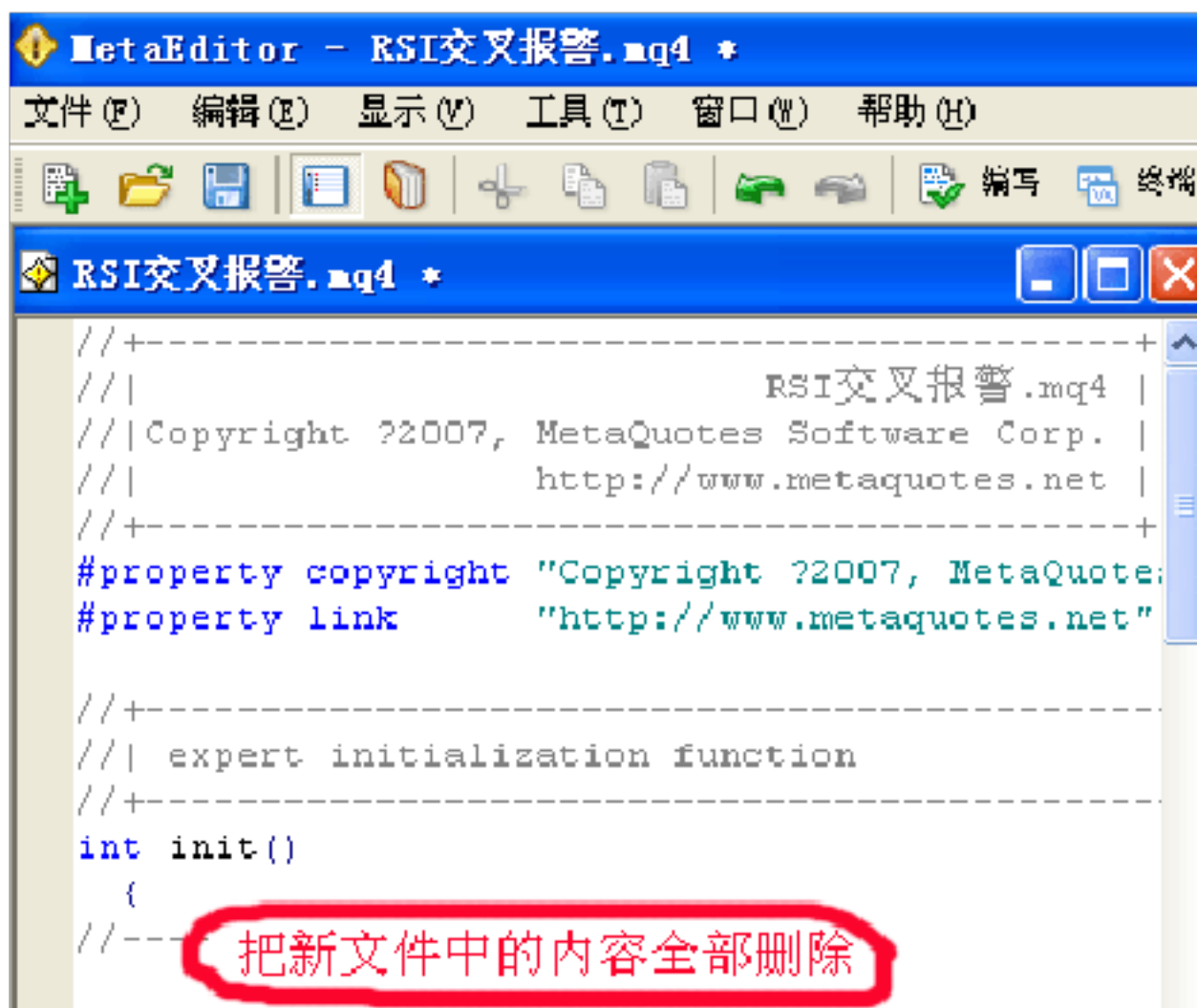
1、打开 MQ 语言编辑器



2、在 MQ 语言编辑器中新建一个文件，文件名一定要输，别的随便操作都没关系。

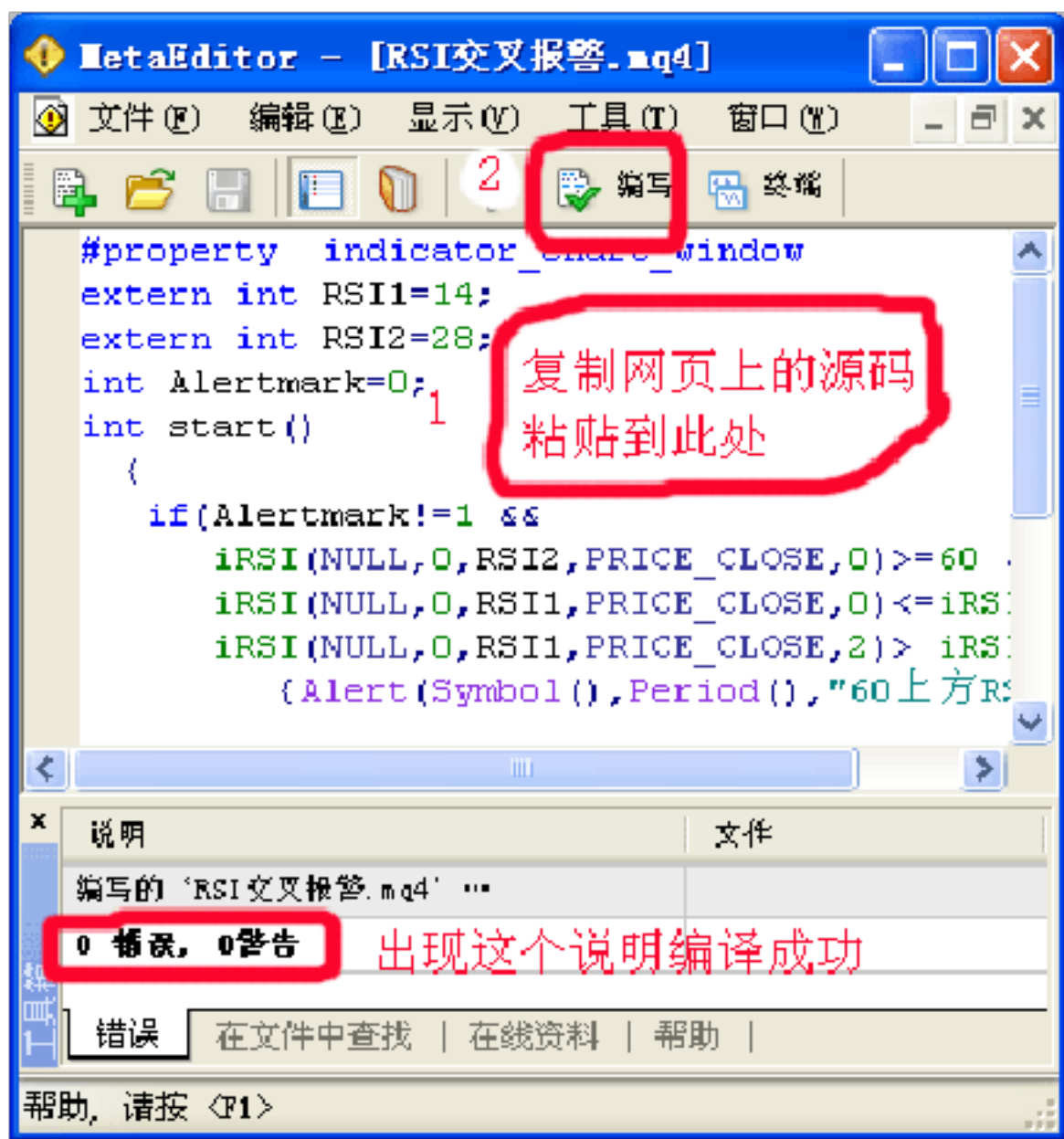


3、删除新文件里面的所有内容，成为一个空文件，

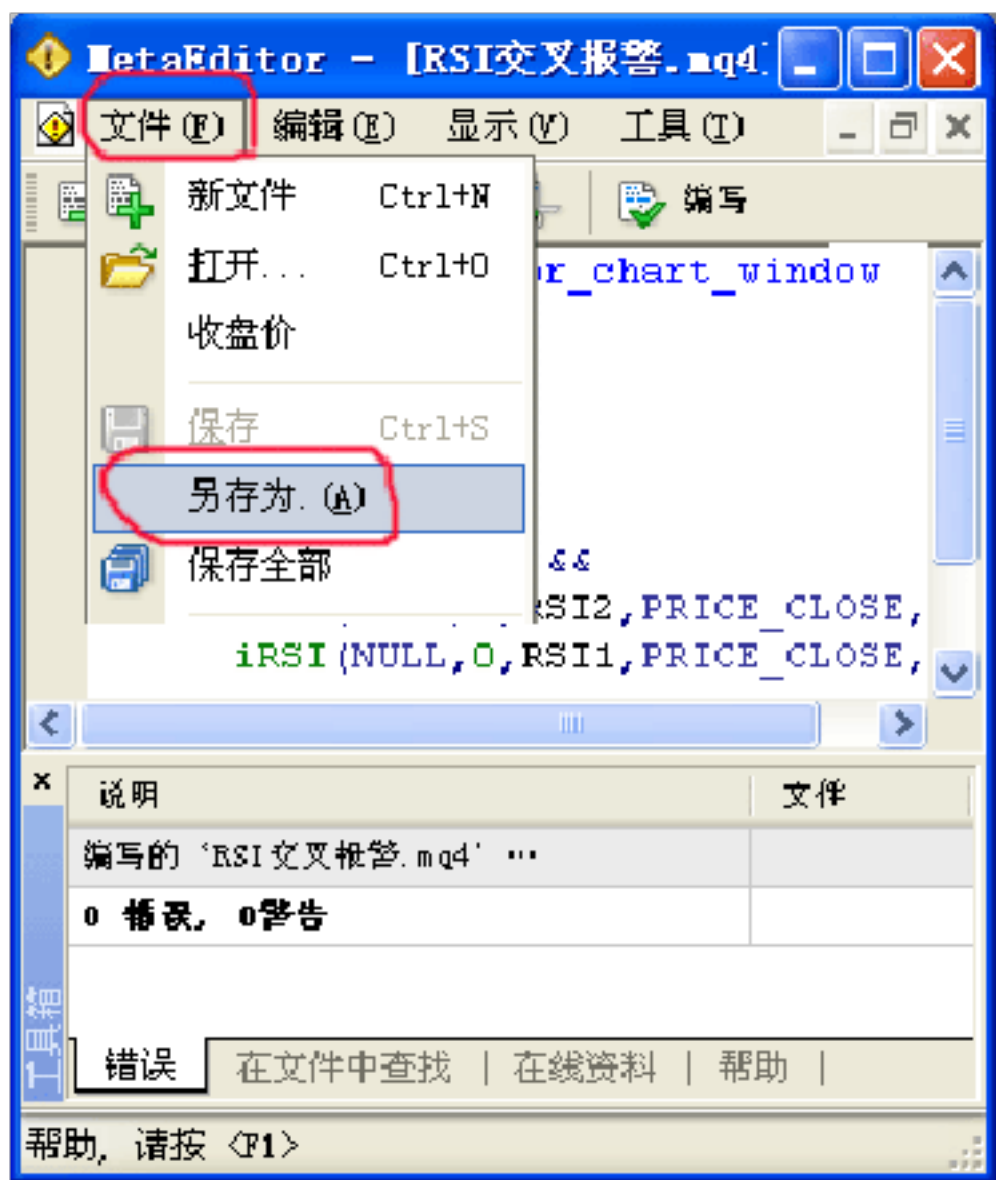


4、接着我们就可以将网页上源码复制粘贴进来

试着点一下“编写”，如果左下角出现“0 错误，0 警告”，那说明源码没问题，编译通过了，成功生成了 ex4 可执行文件



5、还有很重要的一点，就是要把源码文件保存到正确的目录中



保存后，再点一下编写。

到此，源码保存完成。

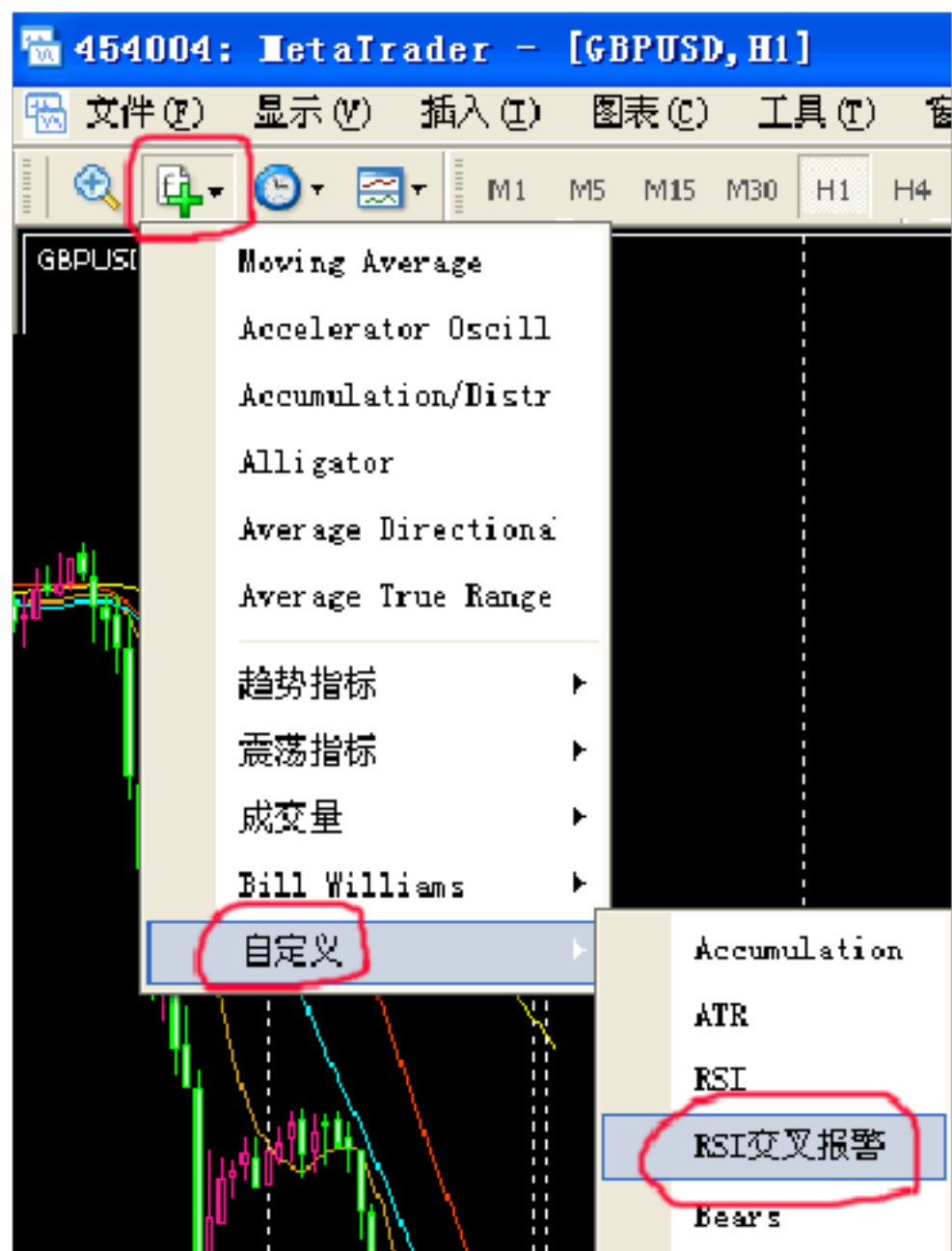
当然，我们也还有另的办法：

可以打开一个原已存在的 mq4 文件，然后另存为你需要的文件名，再删除文件中的所有内容为空文件，把网页上的源贴到文件中保存即可

也可以在记事本中新建一个文件，把网页上的源码贴进记事本，然后取个文件名保存到上面提到的目录，不过文件后缀必须输入 mq4，保存类型选择“所以文件”

四、mt4 程序调用自定义指标

这非常简单



MT4 编程入门 (二): 指标源码中的常用符号说明

一、注释符：单行注释“//”、多行注释“/* */”
程序不执行注释，注释内容仅供阅读或保留不用的语句
在 MQ 语言编辑器中，注释内容以灰色显示

二、分号“;”
表示一个语句结束

三、逗号“,”
表示一个语句内部间隔，不结束语句。

四、等于号“=”
赋值语句，把等号后面的值赋给等号前的变量中

五、加号“+”
整数或小数，则表示值相加；字符串，则表示前后连接。

六、加加 “++”

表示把符号前的变量值加 1，例如：“i++;” 这实际上就是：“i=i+1;”的简写，表示 i 的值加 1

七、加等于 “+=”

表示把符号后的值加到符号前的变量中。例如“i+=2;”，实际上就是“i=i+2;”的简写，表示 i 的值加 2

八、双等号 “==”

等于号。用于逻辑判断，双等号前后值相等，则返回逻辑值“真 true”；双等号前后值不相等，则返回逻辑值“假 false”

九、叹号等号 “!=”

不等于号。用于逻辑判断，不等号前后值不相等，则返回逻辑值“真 true”；不等号前后值相等，则返回逻辑值“假 false”

十、“&& ”

逻辑符“并且”

十一、“||”

逻辑符“或者”

十二、大括号 “{}”

表示函数体。

{ }内部为函数体语句。

MT4 编程入门（三）：自定义变量

一、自定义变量在使用前，必须先创建(定义)，设定变量的类型和名称。

创建时可同时赋值，也可以不赋值。名称可以使用中文字符。

例：

`int mark;` 创建一个整型变量，变量名为“mark”。

`bool 是否报警=true;` 创建一个逻辑型变量，变量名为“是否报警”，同时，赋初值“true”

二、类型和类型定义词

自定义变量的基本类型有：

1、整形(int)：就是整数，包括负整数

2、双精度小数(double)：可以简单地理解为带小数位的数

3、字符串型(string)。

4、逻辑型(bool)：其值分别为“true、false”，即“真、假”，也可以写成“1、0”（1真，0假）。

类型定义词仅在创建变量时使用，后面使用变量时，就不能再加类型定义词了

三、附加变量

附加变量就是在程序投入使用后，变量值允许用户自由更改的变量，一般用作程序参数。

一般指标，例如 MACD(12,26,9)参数输入就是通过附加变量实现，(12,26,9)分别为三个附加变量的默认值。

定义附加变量只需在变量定义语句前加“extern”

例：

`extern int 周期=9;` 创建一个整型附加变量，变量名为“周期”，默认值为 9，允许用户在程序外部修改参数值

MT4 编程入门（四）：自定义函数

自定义函数与数学函数在本质上是一致的

例如：数学函数 $ft(x,y)=3x+2y$ (x,y 为正整数) 写成程序语句就是：

```
int ft(int x,int y)
{
    int temp;
    temp=3*x+2*y;
    return(temp);
}
```

一、和自定义变量一样，自定义函数在使用前，也必须先设立，以定义其类型、名称、参数和运算语句。函数类型、名称、参数位于函数头部(参数间以逗号分隔)，大括号中的是运算语句，也就是函数体部分。

return 语句为函数结束语句，**return** 后面括号中的值为函数返回值，即函数运算结果。

上例中，**int** 为函数类型，表示该函数的返回值(运算结果)是整数；

(x,y)为参数，由于自定义变量使用之前都必须先创建，所以要加上类型定义词 **int**

ft 为函数名，有了函数名，我们就可以在别处用 **ft(2,1)**的形式来调用此函数；

函数体部分有三句：

第一句，定义一个整形变量 **temp**

第二句，计算 **temp** 的值

第三句，函数结束，返回调用点，以 **temp** 的值作为本函数返回值。

此函数也可以简写为：**int ft(int x,int y) {return(3*x+2*y);}**

二、函数类型与类型符

函数类型就是函数返回值(运算结果)的类型，与自定义变量类型差不多

有整型(**int**)、双精度型(**double**)、字符串型(**string**)、逻辑型(**bool**)等，

还有一种函数是以函数运行为目的，运行结束后不需要返回值也没有返回值的，则函数类型可以写成 **void**(无值型)

三、函数中如果定义了自定义变量，那么此变量仅在本函数内部存在，函数运行结束时，自定义变量也就自动注销。

上例中，有 **x**、**y**、**temp** 三个自定义变量，在函数运行时创建，在函数运行结束时自动消失，仅在函数内部有效，与函数外部无关。

四、函数的调用

函数调用非常简单，只要函数名正确、参数个数一致、类型相同就能正确调用

如我们前面创建了 **ft(x,y)**函数，则别处我们可以用 **ft(4,5)**等样式来调用 **ft(x,y)**函数求值。

调用时，传递参数值给 **ft(x,y)**函数(按顺序把 4 传给 **x**，把 5 传给 **y**)，返回结果 22 后继续执行主程序。

MT4 编程入门(五)：自定义数组

一、数组是一系列同类型变量的集合，可以同时存储多个数值。

例：**int aa[4]={4,3,2,1};**

这是一个数据名为 **aa**、包含了 4 个元素的整数型数组(即数组长度为 4)，可以同时存储 4 个数值。

二、自定义数组也必须“先建立，后使用”

创建(定义)时，必须设定类型和名称。

创建(定义)时，数组的元素个数也可以设定，也可以不设定

数组中第一个元素的编号为 0，第二个为 1，以此类推。编号使用方括号括起来

在这上例中 **aa[0]**的值为 4、**aa[1]**的值为 3、**aa[2]**的值为 2、**aa[3]**的值为 1

创建(定义)数组的同时，可以赋初值，也可以不赋初值。

例：

`int bb[];` 创建数组 `bb`，不设定数组元素个数

`int cc[8];` 创建数组 `cc`，设定数组元素为 8 个，不赋初值

`int dd[3]={9,8,7};` 创建数组 `dd`，设定数组元素为 4 个，并赋初值
赋值后，`dd[0]`的值为 9、`dd[1]`的值为 8、`dd[2]`的值为 7

`int ee[9]={0,1,2,3};` 创建数组 `ee`，设定数组元素为 9 个，但只给前 4 个元素赋初值

三、数组使用时的赋值

数组的赋值必须指定明确的赋值对象，除了新建数组时可以批量赋值外，数组元素必须一个一个赋值

例如：`dd[2]=1;`

这是对数组 `dd` 的第 3 个元素 `dd[2]`（`[0]`是第 1 个，`[2]`表示第 3 个）重新赋值为 1

在这里，没有整数型变量定义符 `int`，

这样，数组 `dd` 原有三个值 {9,8,7} 就变为 {9,8,1}

四、数组的引用

数组只是一系列变量的集合，所以每次只能使用数组中的一个元素。

数组的引用必须指定明确的引用对象，一次只能引用一个。如果需要使用整列数组，则必须逐个使用。

例如：数组 `aa[4]` 的值为 {4,3,2,1}

我们用 `aa` 来引用数组 `aa` 中的值

当 `i=0` 时，`aa[i]` 的值为 4

当 `i=1` 时，`aa[i]` 的值为 3

当 `i=2` 时，`aa[i]` 的值为 2

当 `i=3` 时，`aa[i]` 的值为 1

MT4 编程入门(六)：市场数据取值

一、预定义数组（MT4 已经定义好的，可以直接使用）

开盘价、最高价、最低价、收盘价、成交量等历史数据，每根 K 线蜡烛都各有一个，所以必须用数组来解决问题，MT4 有几个预定义数组：

开盘价 `Open[]`、最高价 `High[]`、最低价 `Low[]`、收盘价 `close[]`、成交量 `Volume[]`、所属时间 `Time[]`

类型为双精度 `double` 型（即精确小数）

这里有一个位置的问题，右边第一根 K 线蜡烛（即最新的蜡烛）的编号为 0，第二根蜡烛编号 1，第三根蜡烛编号 2，也就是说从右向左倒着数过去。

`Open[0]`、`High[0]`、`Low[0]`、`Close[0]`，表示最新的开盘价、最高价、最低价、收盘价

`Open[1]`、`High[1]`、`Low[1]`、`close[1]`，表示第 2 根 K 线蜡烛的开盘价、最高价、最低价、收盘价

`Open[2]`、`High[2]`、`Low[2]`、`close[2]`，表示第 3 根 K 线蜡烛的开盘价、最高价、最低价、收盘价

`Open[3]`、`High[3]`、`Low[3]`、`close[3]`，表示第 4 根 K 线蜡烛的开盘价、最高价、最低价、收盘价

`Open`、`High`、`Low`、`close`，表示第 `i+1` 根 K 线蜡烛的开盘价、最高价、最低价、收盘价

以此类推。。。。。

注意：这些是数组，用的是方括号。

二、预定义变量

买入价、卖出价是实时价格，MT4 用预定义变量 `Ask` 和 `Bid` 表示，数值类型为 `double` 双精度

还有一些预定义变量，如：

`Bars` 表示图表中的蜡烛数，类型为 `int` 整数型

`Digits` 表示当前货币对的小数位，类型为 `int` 整数型，无日元币对为 4，有日元币对为 2，黄金石油等一般也为 2

`Point` 表示当前货币对的点值，类型为双精度 `double` 型，无日元币对为 0.0001，有日元币对为 0.01。与 `Digits` 正好相反。

三、指标函数

1、价格、成交量、时间

它们都有三个参数：货币对名、K线周期、位置序号

开盘价：iOpen(symbol,timeframe,shift) 双精度 double 型

收盘价：iClose(symbol,timeframe,shift) 双精度 double 型

最高价：iHigh(symbol,timeframe,shift) 双精度 double 型

最低价：iLow(symbol,timeframe,shift) 双精度 double 型

成交量：iVolume(symbol,timeframe,shift) 双精度 double 型

所属时间：iTime(symbol,timeframe,shift) 日期时间 datetime 型

K线周期为：1分钟图(PERIOD_M1) 5分钟图(PERIOD_M5) 15分钟图(PERIOD_M15) 30分钟图(PERIOD_M30)

1小时图(PERIOD_H1) 4小时图(PERIOD_H4) 日线图(PERIOD_D1) 周线图(PERIOD_W1) 周线图(PERIOD_W1) 月线图(PERIOD_W1)

例：

表示美元兑日元 1 小时图最新 K 线蜡烛的开盘价

表示欧元兑美元 4 小时图第 3 根 K 线蜡烛的收盘价

表示英镑兑美元 1 小时图第 i+1 根 K 线蜡烛的收盘价

iHigh(NULL,0,0)

既不指定商品,也不指定 K 线周期,用在谁就是谁,用在哪就是哪

2、移动平均值。双精度 double 型

iMA(symbol, timeframe, period, ma_shift, ma_method, applied_price, shift)

参数共 7 个，分别为：商品名称、K线周期、均线周期、均线偏移、平均模式、价格种类、位置

均线周期：10 天平均线的均线周期为 10，20 日均线的均线周期为 20

均线偏移：均线位置整体左右移动的位置偏移量

平均模式：简单移动平均(MODE_SMA) 指数移动平均(MODE_EMA) 平滑移动均线(MODE_SMMA) 线性加权移动均线(MODE_LWMA)

价格种类：收盘价(PRICE_CLOSE) 开盘价(PRICE_OPEN) 最高价(PRICE_HIGH) 最低价(PRICE_LOW) 中值(PRICE_MEDIAN) 5(PRICE_TYPICAL) 6(PRICE_WEIGHTED)

例 1：

表示：欧元 1 小时图上，以收盘价计算的，20 小时简单移动均线，最新 K 线所对应位置的值

例 2：iMA(NULL,0,20,0,MODE_EMA,PRICE_CLOSE,2)

表示：在当前商品、当前 K 线周期图表上，以收盘价计算的，20(天)指数移动均线 第 3 根 K 线所对应位置的值

其他如 MACD 指标、威廉指标、RSI SAR 布林线等指标取值都与移动均线指标相类似

3、在数组中求元素的移动平均值。双精度 double 型

iMAOnArray(数组名, 总数, 平均周期, 均线偏移, 平均模式, 位置)

这也与 iMA 函数差不多，不过数据源变为数组

从数组中不但可以求得移动平均值，还可以求得 RSI 等指标值

4、求自定义指标的值

我们经常自己编一些自定义指标，可用 iCustom 函数来取得自定义函数的值

iCustom(商品名,K线周期,自定义指标名称,自定义指标参数 1,参数 2,参数 3,,自定义指标线编号,位置)

如果自定义指标只有一根指标线，则自定义指标线的编号为 0。

，第二条为 1，第三条为 2。。。

例如： (12,26,9)为自定义指标 `mymacd` 的三个参数

表示：求当前图表中，自定义指标 `mymacd(12,26,9)` 的第 3 条指标线在最新位置的值

抛砖引玉，这里只是有代表性地列了几个函数，详细请查阅《MT4 编程手册》

MT4 编程入门(七): 判断语句

一、if 语句

if 语句很常用，也比较简单。

规则：如果小括号中的条件成立，则执行大括号中的语句；如果不成立，则跳过大括号。

例如：

```
if(a==1)
{
    b=c+1;
}
```

我们在编写报警指标的时候，就经常用到这一语句：

如果“价格向上达到指定价位”，则“报警”

如果“MACD上穿”，则“报警”

如果“均线金叉”，则“报警”。等等

例：

```
int mark=0;
```

```
if( High[1]<1.0000 && High[0]>=1.0000 && mark!=1)
```

```
{
    价格向上触及
    mark=1;
}
```

```
if( Low[1]>1.0000 && Low[0]<=1.0000 && mark!=2)
```

```
{
    价格向下触及
    mark=2;
}
```

这是一个价格上、下穿 1.0000 时报警的判断语句：

上穿报警条件：当第二根 K 线最高价小于 1.0000，并且最新 K 线最高价大等于 1.0000

下穿报警条件：当第二根 K 线最低价大于 1.0000，并且最新 K 线最低价小等于 1.0000

这里,mark 是用作报警标记,mark 的初值是 0,

当上穿报警后, mark 的值就改为 1; 当下穿报警后, mark 的值就改为 2;

当 mark=0 时,说明从未报过警,上、下穿都能报警;

当 mark=1 时,说明已经上穿报过警了,不能再上穿报警了,但仍可下穿报警;

当 mark=2 时,说明已经下穿报过警了,不能再下穿报警了,但仍可上穿报警。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/366052242221011004>