

هـ- المتغيرات

يتم الاعلان عن المتغيرات قبل إستخدامها ، ويستخدم أسماء مخصصة وفريده للمتغيرات ويستخدم وصف للمتغيرات ليتم تعريفها ، وتذكر ان الوصف ليس

متحكم Operator

الانواع الرئيسية هي :

* قيم منطقية **bool** وتأخذ القيمتي **true , false**

* سلسلة وهي السلاسل الحرفية

* رقم بفاصلة **double**

مثال :

```
string MessageBox;  
int Orders;  
double SymbolPrice;  
bool bLog;
```

الأنواع الاضافية :

* اللون ويمثل الصيغة **RGB** للألوان كما ذكرنا

* الوقت والتاريخ ويحدد الوقت والتاريخ كما شرحناه مسبقاً

انطلقوا إلى القمة

مثال :

```
datetime tBegin_Data = D'2004.01.01 00:00';  
color     cModify_Color = C'0x44,0xB9,0xE6';
```

المصفوفات

وهي السلسلة المفهرسة من بيانات متماثلة من حيث النوع

FX Arabia
إف إكس أرابيا

مثال :

```
int    a[50];           // A one-dimensional array of 50 integers.  
double m[7][50];       // Two-dimensional array of seven arrays,  
                        //each of them consisting of 50 integers.
```

الرقم الصحيح فقط هو الذى يمكن ان يكون سلسلة مفهرسة ، ولا يسمح بوجود
اربع تسلسلات للمصفوفة الواحدة
ترقيم عناصر المصفوفة يبدأ من صفر

العنصر الاخير من مجموعة ذات تسلسل واحد يكون اقل عدداً ب ١ من حجم
المصفوفة

وهذا يعنى ان استدعاء العنصر الاخير من مصفوفة تتألف من { ٥٠ } ستبدو
وكأنها { ٤٩ }

وتنقسم المتغيرات الى :

* المتغيرات الداخلية

* المتغيرات الثابتة

* المتغيرات الشاملة

* المتغيرات المعرفة خارجياً

* تهيئة وضبط المتغيرات

* تعريف الدوال الخارجية

المتغيرات الداخلية :

أى متغير يتم الاعلان عنه داخل الدالة يعتبر متغير داخلى ، ويقتصر نطاق عمله على الوظيفة الداخلية التى يتم تعريفه لها

يمكن تهيئة أى متغير داخلى بواسطة النتائج الناشئة عن أى تعبير سابق

expression

يتم تخزين المتغيرات الداخلية فى منطقة الذاكرة للدالة المقابلة

مثال :

```
int somefunc ()
{
  int ret_code=0;
  ....
  return (ret_code);
}
```

المتغيرات الثابتة

تصنيف الذاكرة الثابتة يحدد ويعرف على انه متغير ثابت ويتم الاعلان عنه قبل الاعلان عن البيانات الخاصة به

مثال :

```
int somefunc()  
{  
    static int flag=10;  
    ....  
    return(flag);  
}
```

يتم تخزين المتغيرات الثابتة في الذاكرة الدائمة ، ولا تفقد بياناتها او قيمها المخزنة بمجرد الانتهاء من الوظيفة الخاصة بها
المتغير الثابت يتم تهيئته قبل تعريف النوع المعاكس له
على عكس المتغير الداخلى البسيط والذي يمكن قبل أى تعبير
وفى حالة عدم وجود تهيئة واضحة يتم وضعها مبدأياً ك صفر
يتم تهيئة المتغيرات الثابتة مرة واحدة قبل استدعاء الوظيفة الاولى لها
قيمة هذا المتغير لا تُفقد أبداً.

المتغيرات الشاملة:

. يتم تعريف المتغيرات الشاملة في نفس مرحلة الدوال **functions**

مثال :

```
int GlobalFlag=10; // global variable
int start()
{
  ...
}
```

قسم المتغيرات الشاملة هو برنامج كامل خاص بذاته ، يمكن الوصول اليها من جميع الدوال الموجودة والمعلن عنها مسبقاً

يتم وضع قيمة الابتدائية ك صفر إذا لم يتم تعيين قيمة له مسبقاً

يتم تهيئة قيمة فقط بثابت يتوافق فقط مع نوعه

يمكن تهيئة وضبط قيمة المتغيرات الشاملة مرة واحدة فقط بعد تحميل البرنامج

في ذاكرة المنصة الخاصة بالعمل

تهيئة وضبط المتغيرات :

يمكن تهيئة وضبط اى متغير فى تعريفه (اى عند الاعلان عنه) ، لاحظ انه يتم

وضع القيمة الافتراضية لأى متغير ب صفر اذا لم تحدد قيمته

يتم تهيئة وضبط المتغيرات الشاملة والمتغيرات الثابتة ب ثابت من نوع مقابل

المتغيرات الشاملة والثابتة تُضبط مرة واحدة فقط

المتغيرات الداخلية تُضبط كل مره بإستدعاء الداله المقابلة لها

مثال :

```
int    n      = 1;
double p      = MarketInfo(Symbol(),MODE_POINT);
string s      = "hello";
double f[]    = { 0.0, 0.236, 0.382, 0.5, 0.618, 1.0 };
int    a[4][4] = { 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4 };
```

ملحوظة : توضع عناصر المصفوفة داخل اقواس مغلقة ، وقيم العناصر الابتدائية

لها تساوى الصفر

واذا لم تتم تعريف حجم المصفوفة ، سيتم تعريفه بناء على حساب المصفوفة

الفعلى

يمكن تهيئة وضبط جميع المصفوفات بواسطة الثوابت فقط

تعريف الدوال الخارجية :

يتم تعريف الدوال الخارجية في جزء خاص منفصل بها ، ولو حدث ولم يُعثر هذا التعريف سيتسبب بلا شك في حدوث أخطاء عند مرحلة التجميع أو الربط أو التنفيذ للبرنامج

عند وصف كائن خارجي (دالة أو ما الى غير ذلك) لا بد من ذكر الكلمة

import

مثال :

```
#import "user32.dll"
int    MessageBoxA(int hWnd ,string szText,string szCaption,int nType);
int    SendMessageA(int hWnd,int Msg,int wParam,int lParam);
#import "lib.ex4"
double round(double value);
#import
```


لاحظ أن :

يستخدم الاستدعاء **import** لوصف داله من **DLLS** خارجي او من مكتبة

EX4 الاكواد

مؤشرات المتغيرات يمكن ان تمر عبر كل الدوال المستوردة ، واذا كان هناك حاجة لتمرير البيانات من نوع الارقام بالفاصلة العشرية او من نوع الصحيح يجب ان يتم تمرير المصفوفة ذات البعد الواحد للنوع المقابل كعلامة مرجعية

مثال :

```
#import "some_lib.dll"
void PassIntegerByref(int& OneInt[]);
#import
int start()
{
    int array[1];
    //...
    PassIntegerByref(array);
    Print(array[0]);
    //...
}
```

يُتَبَع