

قواعد اللغة ٢ Syntax

بدأنا حديثنا في الدرس السابق عن قواعد اللغة التي يجب أن يتبعها مبرمج الإيم كيو إل عند كتابة برنامج الإيم كيو إل. و أكدنا أنها قواعد صارمة يجب الالتزام بها و لا يمكن أبدا كسرهما. و أسهللنا تلك القواعد بمعرفة بنية السطر البرمجي في لغة الإيم كيو إل. و أعتقد أنه بإمكان المبتدأ تلافي ٩٠% من أخطاء البرمجة بإلمامه فقط بالبنية الصحيحة للسطر في لغة الإيم كيو إل و خاصة قاعدة إنتهاء السطر بفاصلة منقوطة و أستثناءاتها.

اليوم نكمل حديثنا عن قواعد لغة الإيم كيو إل و نبدؤه ببنية Format برنامج الإيم كيو إل بشكل عام.

بنية برنامج الإيم كيو إل Format :

تعطيك لغة الإيم كيو إل الحرية في استخدام المسافات Spaces و التابات Tabs و السطور الفارغة Empty Lines في برنامجك كما تحب و كما يبدو لك أنه البنية الأكثر توافقا مع قراءة و فهم البرنامج.

فيمكنك أن تكتب هذا السطر هكذا:

```
double Macd_Current, Macd_Pervious, Ema_Current, Ema_Previous;
```

و يمكن تقسيمه إلي عدة سطور هكذا:

```
double
Macd_Current,
Macd_Pervious,
Ema_Current,
Ema_Previous;
```

و يمكن كتابته هكذا:

```
double      Macd_Current,      Macd_Pervious,      Ema_Current,      Ema_Previous;
```

يمكنك اختيار البنية التي تحب و وضع المسافات بالطريقة التي تفضل و لكن!! لكن يفضل أن تتبع الطريقة التي يتبعها معظم المبرمجين و ذلك حتي يتم فهم الكود إذا قام بقراءته أحدا غيرك.

و لكن هذه الحرية – و كأي قاعدة – له أستثناءات:

١ - لا يمكنك أن تستخدم سطر جديد و بدلا من المسافة المستخدمة في أمر من أوامر المعالج Preprocessor فمثلا السطر الآتي هو سطر صحيح:

غير أن السطر القادم غير صحيح في لغة الإم كيو إل:

```
#property  
copyright "MetaQuotes Software Crop."
```

ففي السطر السابق سوف تحصل علي خطأ عند محاولة معالجة البرنامج Compiling the program لأنك قد قسمت أمر المعالج #property copyright إلي سطرين. و لكن هذا لا يمنع أن تستخدم مسافة أو أكثر بين كلمتي أمر المعالج:

```
#property      copyright "MetaQuotes Software Crop."
```

٢ - الاستثناء الثاني – والذي يبدو بديهيا – أنك لا تستطيع أن تستخدم مسافات أو سطر جديد في وسط:

الثوابت Constants

المعرفات Identifiers

مفردات اللغة Keywords

سوف نتعلم لاحقا ماذا تعني الثوابت و المعرفات و مفردات اللغة و لكننا الا بصدد شرح هذه القاعدة. لاحظ هذا السطر الصحيح في لغة الإم كيو إل:

```
extern int MA_Period = 13;
```

في السطر السابق **extern** و **int** هما مفردات لغة **keywords**. بينما **MA_Period** هو معرف **identifier**. أما الرقم **13** فهو ثابت **constant**.

فطبقا لهذه القاعدة كل السطور القادمة هي سطور غير صحيحة في الإم كيو إل:

```
ex  
tern int MA_Period = 13; // فصل السطر أحد مفردات اللغة
```

```
extern int MA_Period = 1 3; // ٣ الثابت المسافة
```

```
extern int MA_ Period = 13; // فصلت المسافة المعرف
```

الثوابت Constants:

في لغة الإم كيو إل (و كل لغات البرمجة) الثوابت هي قيم ثابتة لا تتغير عبر حياة البرنامج بعكس المتغيرات Variables و التي ما هي إلا قيم متغيرة عبر حياة البرنامج. الثوابت أيضا هي القيم الافتراضية default values التي يعطيها المبرمج لأحد المتغيرات: كما الرقم 13 في الأمثلة السابقة و هو ما يسمى variable's constant. النوع الثالث للثوابت هي ثوابت اللغة: و التي هي قيم هامة كثيرة الاستخدام خصصت لها لغة الإم كيو إل مفردات لغة سهلة الحفظ. علي سبيل المثال ثوابت الفترات Period constants و ثوابت الأخطاء Error codes. إن حديثنا عن الثوابت في هذا الفصل من باب عرض عناصر لغة الإم كيو إل و لكن للحديث بقية في فصل المتغيرات.

المعرفات Identifiers :

المعرف هو الاسم الذي تعطيه لمتغير Variable name او ثابت Constant name أو وظيفة Function name فهذا الاسم هو الطريقة الوحيدة للنداء (استخدام) هذه المتغيرات و الثوابت و الوظائف.

ففي هذا المثال MA_Period هو معرف identifier :

```
extern int MA_Period = 13;
```

قواعد تسمية المعرف:

يمكن أن تختار أي اسم تشاء للمعرف الذي تنشئه في لغة الإيم كيو إل علي أن تلتزم بهذه القواعد:

- ١- أقصى عدد من الحروف يمكن استخدامه للمعرف هو ٣١ حرف.
- ٢- المعرف يجب أن يبدأ بحرف إنجليزي كبتل Capital كان أو سمول Small أو أندر سكور _.
- ٣- فلا يمكن أستهلل اسم المعرف برقم أو أي رمز غير الأندر سكور.
- ٤- لا يمكن أستخدام أي مفردة من مفردات لغة الإيم كيو إل المحجوزة Reversed words و التي سنتحدث عنها بعد قليل.

ملحوظة



هناك أستثناء لقاعدة عدم إمكانية إستخدام نفس المعرف مرتين و هو ما يطلق عليه مجال Scope المعرف. سوف نتحدث بإسهاب إن شاء الله عن هذا الأستثناء في درس المتغيرات.

حساسية حروف المعرفات:

المعرفات في لغة الإيم كيو إل حساسة لحالة الحروف case sensitive و هو ما يعني أنه الحرف الكابتل ليس كالحرف السمول و لا يعتبر متطابقا معه. ماذا يعني هذا؟؟
يعني هذا أن المعرف MA_Period يختلف عن المعرف ma_period او عن المعرف MA_PERIOD
لذا عليك الانتباه جيدا لحالة الحروف عن إنشاء المعرف و عند أستدعاءه.

مفردات اللغة Keywords :

مفردات لغة الإيم كيو إل هي كلمات ذات معني مقصور في اللغة و ذات أستخدام محدد و فريد. فلا يمكن أستخدام نفس الكلمات كأسماء للمعرفات لذلك تسمى تلك المفردات الكلمات المحجوزة Reversed words.
هذه المفردات متعددة: فمنها مفردات حلقات التكرار كوايل while و فور for و المفردات الشرطية if و else و مفردات تعيين نوع بيانات المتغيرات double و int و string إلي أخره.

هذه قائمة بمفردات لغة الإيم كيو إل:

bool
color
datetime
double
int

string
void
extern
static
break
case
continue
default
else
for
if
return
switch
while
true
false

التعليقات Comments :

التعليقات جزء هام جدا في أي لغة من لغات البرمجة .. و هي إمكانية كتابة كلمات أو سطور داخل كود البرنامج بغرض التعليق علي الكود أو التذكير بشئ ما أو حتي الإشارة إلي معلومات ما عن المبرمج أو البرنامج.

هذه التعليقات سوف يتم تجاهلها تماما عند معالجة Compiling البرنامج من قبل المعالج Compiler و ذلك لتقليل حجم البرنامج.

أنواع التعليقات في الإم كيو إل:

في لغة الإم كيو إل (تماما كما لغة السي) هناك نوعان للتعليقات:

١ – تعليق السطر الواحد Single line comment :

هذا النوع من التعليقات يبدأ بشرطتين مانلتين // و ينتهي بنهاية السطر بداية سطر جديد .. مثال:

```
extern int MA_Period = 13; // هذا تعليق في نفس السطر
```

و هذا مثال آخر لتعليق يأخذ سطر كامل:

```
// هذا تعليق في سطر منفرد
```

و يستخدم هذا النوع من التعليقات إذا أردنا أن نكتب التعليق في نهاية سطرنا .. أو إذا أردنا أن نفرّد سطر واحد للتعليق.

٢ - تعليق عدة سطور Multi lines comment :

هذا النوع من التعليقات يبدأ بهاتين العلامتين /* و ينتهي بهاتين العلامتين */ و يمكن أن يكون في سطور منفصلة و يمكن أن يكون داخل سطر الكود نفسه و ذلك لأنه لا يتطلب أن ينتهي بنهاية سطر و بداية سطر جديد. و يستخدم هذا النوع من التعليقات إذا أردنا أن نكتب تعليق يمتد لأكثر من سطر أو عندما نريد أن نكتب تعليق داخل السطر نفسه (كما في المثل الثاني).

بل أن يستخدم أيضا بكثرة عندما نريد أن نعطل جزء من الكود بدون أن نحذفه من الكود و ذلك بوضع **/*** في بداية السطر الأول من الجزء المراد تعطيله و ***/** في نهاية الجزء المراد تعطيله.

أمثلة:

```
/*  
هذا  
التعليق  
أكثر  
من سطر  
*/
```

```
int MA_Period /* هذا تعليق داخل السطر */ = 13;
```

بهذا نكون قد أنهينا من قواعد اللغة و في الدرس القادم إن شاء الله سوف نتناول المتغيرات.

أحمد سليمان
٢٠٠٨/٠٣/٢٣