

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



- اسم اللغة / MK .
- سبب التسمية / نسبة للحرف الأول و الثاني من اسم مبرمجها **M**ohamed **K**amel
- البريد الإلكتروني / [mk\\_cit@hotmail.com](mailto:mk_cit@hotmail.com)
- هاتف / 0599066526
- مواقع المفضلة /

[www.shababit.ps](http://www.shababit.ps)

<http://www.arabteam2000.com>

أولا // أدوات تصميم هذه اللغة المتواضعة :

Java Programming language  
Jdk 1.7  
JCreator 3.5  
IDE - NetBeans

ثانيا // ملاحظات أساسية :

هذه اللغة تحتاج للعديد من التحسينات ،  
حيث قد يجد و لكن نادرا الشخص المجرب لها  
بعض الثغرات و الأخطاء المنطقية القليلة ،  
و التي سأسعى لتحسينها .

هذه اللغة مصممة على لغة الجافا ، و  
المعروف ب أن الجافا بطيئة ، و لذلك هذه  
اللغة أكثر بطأ بكثير حتى من الجافا .

لقد صممت هذه اللغة كمجرد تحدي لقدراتي ،  
وليس لأي غرض مادي آخر. ولذلك لن تكون لها  
أي إصدارات أو أي تطويرات مستقبلا .

سأسعى يوما لان أصمم لغة برمجة قوية وجيدة  
، لكن باستخدام لغة C/C++ و Assembly .

### ثالثا // أمور يجب تجنبها أثناء كتابة الكود :

عند كتابة عملية طرح بين متغيرين أو رقمين ، يجب عليك وضع مسافة بين الإشارة - و بين الرقم الثاني ،  
مثال :

$x = 10 - 5$

لاحظ المسافة (على الأقل واحدة بين الإشارة السالبة ، وبين الرقم ٥) ،  
لو أردت أن تنسب إشارة السالب لرقم معين ، فهذا يجوز بدون مسافات

مثال :

$x = -10$

لاحظ لا يوجد مسافات بن السالب و الرقم ١٠

---

في جمل التكرار و الشرط تجنب كتابة الأقواس لغير معادلات الحسابية  
مثال :

$\text{If } ((10+5) - x) * 20 + 5 < 10 * (-5+30)$

في المثال السابق لم نستخدم قوس عام الذي يشمل كل الشرط ، و إنما أقواس فقط للعمليات الحسابية ، لذلك هو مثال صحيح ،

مثال : (مثال خاطئ)

$\text{If } ((10+5) - x) * 20 + 5 < 10 * (-5+30)$

القوس الأحمر في المثال السابق هو قوس عام ، هذا القوس سيسبب بعض المشاكل .  
وطبعاً و قس المثال السابق على جمل الشرط في التكرار while .

في أثناء عمليات الشرط ، تجنب مقارنة الحروف ( أي المقارنة بين الحروف و غيرها من أنواع المتغيرات ) ، كذلك تجنب مقارنة النصوص ، حيث لم يتم تدعيم هذا النوع من العمليات بشكل كبير لحد الآن .

---

أرجو من الشخص المجرب لهذه اللغة ، في حالة أي وجود أخطاء في التنفيذ عدا تلك المشاكل التي تحدثت عنها في الملاحظات السابقة ، إبلاغي عنها حتى اعمل على تحسينها .

## رابعاً // البداية مع لغة MK :

### ١- الكلمات المحجوزة الرئيسية :

|         |           |         |        |        |
|---------|-----------|---------|--------|--------|
| start   | procedure | int     | use    | return |
| while   | function  | real    | string | char   |
| if      | boolean   | repeat  | times  | show   |
| And     | txtcolor  | ln      | lt     | give   |
| true    | txtstyle  | else    | clear  | or     |
| txtsize | txtfont   | bgcolor | false  |        |

### ٢- رموز مساعدة :

|   |   |   |
|---|---|---|
| ( | ) | ; |
| { | } | , |

### ٣- الإشارات و العمليات :

|    |             |                           |      |
|----|-------------|---------------------------|------|
| +  | الجمع       | 5 + 5                     | = 10 |
| -  | الطرح       | 5 - 5                     | = 0  |
| *  | الضرب       | 5 * 5                     | = 25 |
| /  | القسمة      | 5 / 5                     | = 1  |
| // | باقي القسمة | 5 // 5                    | = 0  |
| ^  | القوة       | 5 ^ 2                     | = 25 |
|    | دمج نصين    | "X := "    20 = "X := 20" |      |

### ٤- إشارات المقارنة :

|    |    |    |
|----|----|----|
| <  | >  | <> |
| <= | >= | =  |

## ٥- الأولويات الحسابية :

1.       ()
2.       ^
3.       \*,/,//
4.       +,-
5.       ||

## مثال على الأولويات الحسابية :

```
x = 10 + 20 - ( 15 + 15 ) // 5 * 2 ^ 2 || " it is X";
```

1.       ( 15 + 15 ) = 30  
x = 10 + 20 - 30 // 5 \* 2 ^ 2 || " it is X";
2.       2 ^ 2 = 4  
x = 10 + 20 - 30 // 5 \* 4 || " it is X";
3.       30 // 5 = 0  
x = 10 + 20 - 0 \* 4 || " it is X";
4.       0 \* 4 = 0  
x = 10 + 20 - 0 || " it is X";
5.       10 + 20 = 30  
x = 30 - 0 || " it is X";
6.       30 - 0 = 30  
x = 30 || " it is X";
7.       30 || " it is X" = "30 it is X"  
x = "30 it is X";

## ٦- التعليقات :

|   |                                |
|---|--------------------------------|
| # | تستخدم للتعليق في سطر واحد فقط |
| @ | تستخدم للتعليق في عدة أسطر     |

مثال :

```
# You are now at MK Documentation.
```

مثال :

```
@ Welcome in MK Language,  
Thanks for using it,  
We hope for you beautiful session @
```

خامسا // شرح اللغة :

## Start

هذه الكلمة تستخدم للدلالة على التركيب الرئيسي للبرنامج ، حيث لا يعمل البرنامج ، ولا يقوم بالتنفيذ إلا في وجو هذا التركيب الرئيسي.

مثال :

```
start
{
    #write any thing.
}
```

## Data Types

|         |             |
|---------|-------------|
| int     | عدد صحيح    |
| char    | قيمة حرفية  |
| real    | عدد حقيقي   |
| boolean | متغير منطقي |
| string  | متغير نصي   |

طريقة تعريف المتغير :

*<Data type> <identifier> [,<identifier>];*

مثال :

```
Start
{
    Int x,y,z;
    Real s;
}
```

مثال :

```
Start
{
    Int x = 10;
    Show x;
    Show x*10;
}
```

مثال :

```
Start
{
    String name;
    Name = "Mohamed KK.";
    Show Name;
}
```

# Show

هذه الكلمة تستخدم لطباعة جملة معينة .  
تأخذ التركيب التالي :

*Show <statement> ;*

مثال :

Start

```
{  
    Show "Hello, You are programmer of  
    MK." ;  
}
```

# Give

جملة القراءة ، حيث تستخدم لإدخال بيانات من  
قبل المستخدم .  
تأخذ التركيب التالي :

*Give <identifier> ;*

مثال :

Start

```
{  
    Int x;  
    Give x;  
    Show x;  
}
```

# ln

تستخدم لطباعة سطر جديد .

*ln ;*

مثال :

```
Start
{
    Int x;
    Give x;
    Show x;
    ln;
    show "finish.";
}
```

# lt

تستخدم لطباعة 8 أحرف Tab .

*lt ;*

مثال :

```
Start
{
    Int x;
    Give x;
    Show x;
    lt;
    show "finish.";
}
```

# txtcolor

تستخدم لتغير لون الخط .

*Txtcolor <Color> ;*

الألوان المستخدمة :

|           |                       |
|-----------|-----------------------|
| blue      | أزرق                  |
| red       | أحمر                  |
| green     | أخضر                  |
| black     | أسود                  |
| white     | أبيض                  |
| pink      | وردي                  |
| yellow    | أصفر                  |
| gray      | رمادي                 |
| darkgray  | رمادي غامق            |
| lightgray | رمادي فاتح            |
| orange    | برتقالي               |
| cyan      | أزرق                  |
| magenta   | أحمر أرجواني - بنفسجي |

مثال :

```
Start
{
    Int x;
    Give x;
    Show x;
    Txtcolor red;
    show "finish.";
    Txtcolor blue;
    show "Thanks.";
}
```

# Bgcolor

تستخدم لتغير لون الخلفية .

*Bgcolor* <Color> ;

مثال :

```
Start
{
    Bgcolor green;
    Int x;
    Give x;
    Show x;
    Txtcolor red;
    show "finish.";
}
```

# Txtstyle

تستخدم لإعطاء نمط معين للخط.

*Txtstyle* <Style> ;

الأنماط المستخدمة :

|                |                        |
|----------------|------------------------|
| <b>bold</b>    | سميك                   |
| <i>italic</i>  | مائل                   |
| <u>uline</u>   | تحت خط                 |
| <u>iline</u>   | يتوسطه خط              |
| <b>cbold</b>   | إلغاء الخط السميك      |
| <i>citalic</i> | إلغاء الخط المائل      |
| <u>culine</u>  | إلغاء نمط الخط السفلي  |
| <u>ciline</u>  | إلغاء نمط الخط المتوسط |
| <b>normal</b>  | خط عادي                |

مثال :

```
Start
{
    Txtstyle bold;
    Txtstyle uline;
    show "Bold & UnderLine Font style..";

    Txtstyle cbold;
    Txtstyle culine;
    Txtstyle italic;
    Txtstyle inline;
    show "Italic & inseite Line Font
    style..";
}
```

## Txtfont

تستخدم لتغيير نوع الخط.

*Txtfont <Font> ;*

الخطوط المستخدمة :

|                  |
|------------------|
| times_new_roman  |
| monotype_corsiva |
| georgia          |
| comic_sans_ms    |
| arial            |
| arial_black      |
| lucida_console   |
| tahoma           |

مثال :

```
Start
{
    Txtfont TIMES_NEW_ROMAN;
    show "Bold & UnderLine Font style..";

    Txtfont LUCIDA_CONSOLE;
    show "Italic & inseide Line Font
    style..";
}
```

## Txtsize

تستخدم لتغيير حجم الخط.

```
Txtsize <size> ;
```

مثال :

```
Start
{
    Txtsize 30;
    show "Bold & UnderLine Font style..";

    Txtsize 10;
    show "Italic & inseide Line Font
    style..";
}
```

# Clear

تستخدم لمسح شاشة التنفيذ.

```
Clear ;
```

مثال :

```
Start
{
    show "Welcome ..";
    Clear;
    show "Thanks for using MK-Lang..";
}
```

## If - Else If - Else

جمل الشرط و التحكم .

```
if <Condition> { <Statements> } <Else>
if>|<Else>
```

مثال :

```
Start
{
    int x = 120;
    if x // 2 = 0
    {
        show x||" is even.";ln;
    }
    else
    {
        show x||" is odd.";ln;
    }
}
```

مثال :

```
Start
{
    int x;
    show "enter the value of x";
    ln;
    give x;
    if x < 0
    {
        show x||" is negative.";
        ln;
    }
    else
    if x > 0
    {
        show x||" is positive.";
        ln;
        if x // 2 = 0
        {
            show x||" is even.";ln;
        }
        else
        {
            show x||" is odd.";ln;
        }
    }
}
```

# Repeat

جملة التكرار و التحكم  
وتستخدم لعملية التكرار لعدد محدد.

*Repeat <Number of loop> times  
<Statement>*

مثال :

```
Start
{
    repeat 10 times
    show "Welcome for you in MK-Lang.";
}
```

مثال :

```
Start
{
    int x,counter;
    int sum;
    show "enter no of times for getting
    the sum";ln;
    give x;
    repeat x times
    {
        sum = sum + counter;
        counter = counter +1;
    }
}
```

# While

جملة التكرار و التحكم  
وتستخدم لعملية التكرار على حسب تحقق الشرط.

*While <Condition> <Statement>*

مثال :

Start

```
{
    int x ;
    while x < 10
    {
        show "x := " || x;
        x=x+1;
    }
}
```

مثال :

Start

```
{
    int x,counter;
    int sum;
    show "enter no of times for getting
    the sum";ln;
    give x;
    while counter < x
    {
        sum = sum + counter;
        counter = counter +1;
    }
    show "sum := " || sum;ln;
}
```

# Procedure

تستخدم لإنشاء دالة (إجراء) بدون إرجاع أي قيمة

```
Procedure <Proc_Name>(<Parameters>)  
{  
<Statements>  
}
```

ملاحظات هامة :

- يتم فقط تعريف الدوال فقط في بداية البرنامج .
- يجب أن تكون الدالة قد أنشأت ، قبل أن يتم استدعاءها ، فهي تشبه تعريف المتغيرات ، حيث لا يتم استدعاء المتغير إلا إذا تم تعريفه .
- هذه النقاط تنطبق أيضا على الوظائف . Functions

مثال :

```
Procedure sum(real x,real y)  
{  
    show "The summation := "||x+y;ln;  
}  
  
Start  
{ }
```

# Function

تستخدم لإنشاء دالة (وظيفة) مع إرجاع أي قيمة

```
Function <Data Type>
<Proc_Name>(<Paramerters>)
{
<Statements>
return <Statement>;
}
```

مثال :

```
Function real sum(real x,real y)
{
    return x*y;
}
Start
{ }
```

# Use

تستخدم لاستدعاء دالة أو وظيفة (Function or Procedure)

```
Use <Proc_Fun_Name>(<Arg>);
```

مثال :

```
Function real sum(real x,real y)
{
    return x*y;
}
Start
{

    real sum = use sum(15.5,50.87);
    show " sum := " || sum;
}
```

مثال :

```
Function real sum(real x,real y)
{
    return x*y;
}
procedure print_welcome(String name)
{
    show "Welcome for you mr/ms
    "||name;ln;
}
Start
{
    string name;
    show "Enter the name.";
    give name;
    use print_welcome(name);
    real sum = use sum(15.5,50.87);
    show " sum := "||sum;
}
```

انتهى الشرح

بحمد الله

