

# كلمات بالأمثلة

## الأمثلة - Statements

Print statement	اطبع "مرحباً"، 2+1
Print both strings on the same line	اطبع "مرحباً"، ... اطبع " بكم"
<ul style="list-style-type: none"> <li>- The read statement can have an optional message before the variables to be read</li> <li>- The read statements always reads string values, except when putting a # symbol before the variable name</li> <li>- Variable in Kalimat have no type declarations. the types are for values, not variables</li> </ul>	اقرأ "ما اسمك؟"، س اقرأ #س، #ص
<u>If statements</u> and => س وأيضا ص or => س أو ص not => ليس س nor => ليس س ولا ص	إذا س<5 وأيضا س>20: اطبع "القيمة سليمة" وإلا: اطبع "القيمة خطأ" تم على نفس السطر: إذا ص>20 : اطبع "نعم" وإلا اطبع "لا"
This will print صحيح, the 'true' boolean value.	اطبع 6=6
This will print خطأ, the 'false' boolean value.	اطبع 10<5
<u>Traditional for loop</u> the range is inclusive, in this examples 1,2,3...12 will be printed.	لكل أ من 1 إلى 12: اطبع أ تابع
While loop	أ=1 كرر مادام أ>=12: اطبع أ أ = أ + 1 تابع
Infinite loop	كرر: اطبع "مرحباً" تابع
<u>Defining procedures</u> <ul style="list-style-type: none"> <li>- Procedures return no values</li> <li>- parameters have no type declarations</li> <li>- if the procedure takes no parameters an empty param list must still be supplied (like C)</li> <li>- a procedure definition can be at the beginning of a program, at its end or even at its middle.</li> </ul>	اجراء اطبع.مجموع(س، ص): اطبع س+ص نهاية اطبع.مجموع(12، 13)
<u>Defining functions</u> Functions are like procedures, except they must return a value.	دالة مجموع(س، ص): ارجع ب: س+ص نهاية

The color is optional, it is a value from 0 to 15 of pre-defined colors	ارسم.خط (س 1، ص 1) - (س 2، ص 2)، اللون
The color and fill are optional the fill is a boolean value (صحيح/خطأ), meaning to draw the rectangle & fill it or just draw it	ارسم.مستطيل (س 1، ص 1) - (س 2، ص 2)، اللون، ملء
again, color and fill are optional	ارسم.دائرة (س، ص)، نصف.القطر، اللون، ملء
again, color is optional	ارسم.نقطة (س، س)، اللون
Load a sprite (moving image used e.g. in games) from a picture file like .bmp or .png and store the sprite object in a variable.	ط = حمل.طيف ("اسم الملف")
Put the sprite on screen. a sprite can only exist in one location so drawing the same sprite in 2 different places erases the first one. also, a sprite does not destroy the background behind it.	ارسم.طيف ط في (س، ص)
Show and hide sprites	اخف.طيف (ط) اظهر.طيف (ط)
Various functions to read the position and size of sprites.	يمين.الطيف (ط)، يسار.الطيف (ط)، قمة.الطيف (ط)، قاع.الطيف (ط)، عرض.الطيف (ط)، ارتفاع.الطيف (ط)
Define a class, create an instance and write and read its fields. Instead of obj.field like C++ we can use either field obj or field \$ obj  Not using the \$ is more natural since it sounds like natural Arabic.	فصلة شخص: له اسم، سن نهاية  ن = شخص جديد اسم ن = "سمير" سن ن = 13 اطبع اسم \$ ن اطبع سن \$ ن

### بعض الدوال الجاهزة - Some built in functions

<ul style="list-style-type: none"> <li>- Create a single dimensional array</li> <li>- Create a multidimensional array</li> <li>- Print the number of elements in a 1d array</li> <li>- Write an array element (arrays are one-based)</li> <li>- Write a 2d array element</li> </ul> <p>Array values can be created by listing members between brackets.</p> <p>Note: Like all variables, arrays have no type declarations, and its possible to put differently-typed values in the same array</p>	<p>م = مصفوفة (5)  ن = مصفوفة.متعددة ([2,3])  اطبع عدد (م)  م [1] = 5  ن [1,1] = "مرحبا"</p> <p>م = [1, 2, 3, 4]</p>
---	--

Random number from 0 to n-1	عشوائي(50)
Convert strings to numbers and numbers to strings	كعدد("12") كنص(12)
<ul style="list-style-type: none"> <li>- String length</li> <li>- first n characters of a string</li> <li>- n character last</li> <li>- mid(m, n) → middle n characters starting from position m, zero based</li> <li>- Splitting a string, returning an array of tokens</li> <li>- Trimming a string (remove leading or trailing whitespace).</li> <li>- String replace</li> </ul>	س = "كلية الحاسبات" اطبع طول(س) اطبع أول(س، 4) اطبع آخر(س، 8) اطبع وسط(س، 4، 1) م = تفصيل("دجاجة بقرة أوزة") م = تقليم(" مرحباً ") م = تبديل("أنا أحب المكرونة، وسوف أكل بعض المكرونة"، "المكرونة"، "الفطير")
Sin, cos, tan, square root, log10, ln, remainder	جا(س) جتا(س) طا(س) جذر(س) لو(س) لو.ه(س) باقي.قسمة(أ، ب)
<ul style="list-style-type: none"> <li>- Wait for a number of milliseconds</li> <li>-Erase all drawing (but not sprites)</li> <li>- Erase all text</li> <li>- Set cursor position</li> <li>- Get cursor line</li> <li>- Get cursor column</li> <li>- Set text color (value from 0 to 15)</li> </ul>	انتظر (س) امسح.الشاشة ( ) امسح.الكتابة() حدد.مكان.المؤشر(الصف، العمود) الصف = سطر.المؤشر() العمود = عمود.المؤشر() حدد.لون.النص(اللون)

## Event handling - التعامل مع الأحداث

مثال على التعامل مع ضغط زر الماوس:

إجراء تعامل.مع.الماوس(س،ص): اطبع "أنت ضغطت في"، س، "و"، ص نهاية عند حادثة ضغط.زر.ماوس نفذ تعامل.مع.الماوس اطبع "اضغط بالماوس في أي مكان"
--

## أنواع الأحداث الموجودة:

ضغط.زر.ماوس

رفع.زر.ماوس

تحريك.ماوس

ضغط.مفتاح

رفع.مفتاح  
ادخال.حرف  
ملاحظات:

- Mouse events always take x,y values as parameters
- Keyboard events always take 2 parameters: an integer representing the key code (for the physical key itself, regardless of current input language) and a string representing the input character (depends on input language).

## تصادم الأطياف - Sprite collision detection

ط 1 = حمل.طيف("...")  
ط 2 = حمل.طيف("...")

ارسم.طيف ط 2 في (400،400)

عند حادثة تصادم نفذ الأطياف.اصطدموا

لكل أ من 300 إلى 450:  
ارسم.طيف ط 1 في (أ، 400)  
تابع

إجراء الأطياف.اصطدموا(طيف 1، طيف 2):  
اطبع "تصادم!"  
نهاية

### General notes

Entering Symbols on the Arabic keyboard:

- shift + خ = ×
- shift + هـ = ÷
- shift + ي = ]
- shift + ب = [
- shift + د = >
- shift + ج = <

## Defining methods in classes

فصلة نقطة:

له س، ص

يستجيب ل: ارسم(، حرك(س، ص)  
يرد على المسافة.من(نقطة.اخرى)  
نهاية

استجابة نقطة ن ل: ارسم(:

ارسم.نقطة (س ن، ص ن)  
نهاية

استجابة نقطة ت ل: حرك(س.الجديدة، ص.الجديدة):  
 س ت = س.الجديدة  
 ص ت = ص.الجديدة  
 نهاية

رد نقطة ن على المسافة.من(ن 2):  
 فرق.س = س ن - س ن 2  
 فرق.ص = ص ن - ص ن 2  
 النتيجة = جذر(فرق.س×فرق.س + فرق.ص × فرق.ص)  
 ارجع ب: النتيجة  
 نهاية

ن 1 = نقطة جديد  
 ن 2 = نقطة جديد

س ن 1 = 4  
 ص ن 1 = 3  
 س ن 2 = 1  
 ص ن 2 = 1

ن 1:ارسم()  
 ن 2: حرك(0, 0)

اطبع ن 1:المسافة.من(ن 2)

## الملفات - Files

Create an empty new file for writing	م = انشيء.ملف("...")
Open an existing file for reading	م = افتح.ملف("...")
Opens a file for writing from the end of the existing data.	م = الحق.بملف("...")
Closes the file.	اغلق.ملف(م)
Returns true if at the end of the file (e.g all data has been read).	نهاية.الملف(م)
Print to an open file	اطبع في م: "النتيجة"، س
Read from an open file	اقرأ من م: س، ص
Object-oriented file input/output.	م: اكتب.سطر("هذه هي البيانات") م: اكتب("وهذه أيضا") م: اكتب.بعرض("وهذه"، 5) ب = م: اقرأ.سطر() ن = م: منته()

## Units – الوحدات

<u>Defining units</u> -The unit name is an identifier, not a string.  - The unit name is currently unused, but still mandatory (it could be used in future versions for namespacing).  - Only definitions (procedures, functions, classes, global variables) could be included in a unit, no statements.	(ملاحظة: خزن هذا الملف بإسم myunit.k) وحدة الجمع  دالة مجموع (أ، ب):  ارجع ب: أ+ب  نهاية
<u>Using units</u> - Units are used by their file name, not unit name.  - The file name is a string.  - It is possible to use a unit multiple times or for units to include each other (possibly indirectly).	باستخدام "myunit.k"  اطبع مجموع (12، 13)