

# ألف 2

لغة البرمجة العربية



©2020

DRAGA Hassan

[www.aliflang.org](http://www.aliflang.org)

Canada





[إصدار تجريبي 2.0.5 ]

25. 4. 2020

## تنبيه

مشروع لغة البرمجة العربية ألف مازال تحت التطوير، هذا الكتاب يشرح آخر نسخة ظهرت، لذى لتحميل آخر نسخة من الكتاب المرجو زيارة الموقع الرسمي

الموقع الرسمي :

<https://www.aliflang.org>

المنتدى الاجتماعي الرسمي :

<https://www.aliflang.org/community>

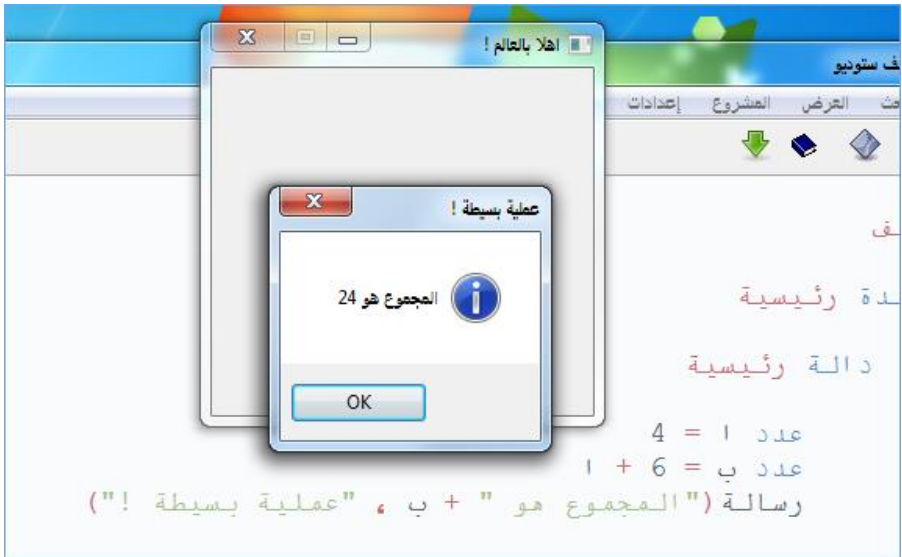
## الفهرس

---

- لمن هذا الكتاب ؟ ..... صفحة 6
- مقدمة ..... صفحة 7
- نقاش و دراسة حول لغات البرمجة العربية ..... صفحة 8
- الفصل الأول - أساسيات عامة في البرمجة ..... صفحة 11
- الفصل الثاني - أساسيات في لغة ألف ..... صفحة 15
- الفصل الثالث - البرمجة بلغة ألف ..... صفحة 31

## لمن هذا الكتاب ؟

هذا الكتاب بمثابة دليل تعليمي للغة البرمجة العربية ألف، حيث يعتبر هذا الكتاب مدخلا للبرمجة، بإستعمال لغة البرمجة العربية ألف.



مثال لتطبيق مكتوب بلغة ألف

### نقاش و دراسة حول لغات البرمجة العربية

منذ أن دخلت عالم المعلوماتية، ما فتئت أسمع أسئلة مثل: هل توجد لغة برمجة عربية، هل يمكن أن توجد، أسئلة تطرح على عدة مستويات، وعلى الرغم من أنّ الفكرة قديمة قدم لغات البرمجة، إلا أنّ المتناقشين ينقسمون إلى مؤيد ومعارض حتى لفكرة إنشاء اللغة البرمجية. وسنستعرض فيما يلي آراء المؤيدين والمعارضين التي استخلصناها من النقاشات المفتوحة عبر الإنترنت في مواقع المنتديات والمواقع الشخصية

يدور في المواقع العربية وفي أوساط المبرمجين العرب منذ تعلمهم البرمجة نقاش حول لغة البرمجة العربية، لماذا وكيف؟ وعلى الرغم من أنّ الفكرة قديمة قدم لغات البرمجة، إلا أنّ المتناقشين ينقسمون إلى مؤيد ومعارض

يقول بعض المعارضين "ما الجدوى أصلا من وجود لغة برمجة عربية"، هل نوجدها فقط من أجل التفاخر

والتباهي، ثم لا تجد لها صدى، لماذا يستخدم جميع الناس في العالم يستخدمون لغات برمجة معروفة ونحاول نحن أن نكون مختلفين منعزلين؟ إذ لا توجد لغات برمجة بأي لغة أخرى سوى الإنجليزية كما انه لا يمكن تصور لغة برمجة بالعربية من اليمين إلى اليسار، ولماذا نبدأ من لماذا نبدأ من الصفر إذا كان غيرنا قد قطع أشواطاً كبيرة

فيرد المؤيدون أنّ الجدوى من وجود لغة برمجة عربية هو الجدوى من لغات البرمجة أصلاً، ويمكننا أن نستعملها في تعليم المبتدئين والأطفال، ومخاطبة الآلة بلغة عربية غير غريبة عن المستخدم العربي، ويمكن إيجاد مجالات للاستخدام عديدة بتعدد مجالات البرمجة، وليس بفرض التباهي، والعمل على لغة

برمجة عربية، ذات استخدامات واسعة أو محدودة لا يعني أننا سننزل عن العالم، إذ يبقى استعمال اللغات العالمية ممكناً أيضاً، ولن يؤدي تصميم لغة عربية إلى هذه العزلة. أم القول بأنه لا توجد لغات برمجة بغير الإنجليزية، فهو غير صحيح لوجود لغات برمجة مختلفة تستخدم الصينية والكورية والعبرية والأسبانية والألمانية والفرنسية والروسية لكتابة أوامرهما، هؤلاء الذين قاموا بإنشاء هذه اللغات لم يستغنوا عن لغات البرمجة الشهيرة، كل ما في



الأمر أنهم قاموا بإنشاء هذه اللغات لكي تتناسب مع احتياجاتهم وثقافتهم، ليتمكن لأي كان تعلم البرمجة دون وجوب تعلم اللغة الإنجليزية

بعد كل هذا فالمعارضون يقولون للمؤيدين، كان الله في عونكم، فتطوير لغات برمجة قوية يحتاج دعماً مؤسسياً ومالياً كبيراً ليس من السهل توافره. وحتى إن توفر المال فلا يمكن تعريب كل المصطلحات البرمجية.

يرد المؤيدون على كل ما سبق بأن التفكير بأسلوب الكل أو لاشي، لا يفيد أبداً في التطور، كل الأمور تبدأ صغيرة ثم تكبر لقد حاول العرب منذ بدايات لغات البرمجة مواكبة هذا التطور بتعريب لغات البرمجة الخاصة بهم أو تصميم لغات مستقلة، لأهداف عديدة كان أهمها الهدف التعليمي، وإن وجود مثل هذه اللغات دليل على إمكانيات وطاقات عربية في جميع المجالات، وإن كانت لغات البرمجة العربية لم تلق انتشاراً كبيراً في زمن مضى، فإن الاستعمال الواسع للإنترنت، يمكن أن يسمح بتظافر الجهود، والتعاون بين الخبراء من أجل إنتاج لغات برمجية عربية راقية في مختلف الميادين

وأخيرا فإن وجود لغة برمجية عربية خطوة صحيحة نحو  
النهضة العربية

- بقلم الأستاذ طه زروقي

---

## الفصل الأول

---

### أساسيات عامة في البرمجة

### تاريخ البرمجة

ظهرت الحواسيب الكهربائيّة الأولى في العصر الحديث في السنوات 1940 (الأربعينيّات)، كانت البرمجة تتم بواسطة لغة الآلة، باستخدام الرقم الصفر أو الواحد فقط، وقد كانت تتطلّب الكثير من الجهد الفكريّ، وكانت تؤدي لأخطاء عديدة، ثم ظهرت لغة التجميع سنة 1947، كانت سهلة مقارنة مع لغة الآلة، حيث بدل استعمال سلاسل من الصفر أو الواحد، كانت تستعمل كلمات انجليزية مفهومة مثل MOV, ADD

مع مرور الوقت تم تطوير لغات برمجة أخرى تعتبر أسهل من لغة التجميع، مثل لغة كوبول (1959)، بيسيك (1964)، سي (1972)، سي++ (1980)، بايثون (1991)، جافا (1995)، سي شارب (2000)، ثم لغة ألف (2018)

<pre>00100101 11010011 00100100 11010100 10001010 01001001 11110000</pre>	<pre>#include &lt;stdio.h&gt;  int main(){     printf(" السلام عليكم ");     return 0; }</pre>
<p>لغة الآلة 1940</p>	<p>لغة سي 1972</p>
<pre>section .text global _start _start:     mov     edx, len     mov     ecx, msg     mov     ebx, 1     mov     eax, 4     int     0x80     mov     eax, 1     int     0x80 section .data msg      db " السلام عليكم ", 0xa len      equ \$ - msg</pre>	<pre># ألف نافذة رئيسية دالة رئيسية رسالة (" السلام عليكم ") نهاية دالة نهاية نافذة</pre>
<p>لغة التجميع 1947</p>	<p>لغة ألف 2018</p>

مثال حول تطور لغات البرمجة

## أدوات عامة في البرمجة

حتى تستعمل أية لغة برمجة كيفما كانت، فإنك ستحتاج إلى مجموعة من التطبيقات و الأدوات تساعدك على استعمال تلك اللغة، هذه بعض التطبيقات و الأدوات المشهورة :

- المترجم

تطبيق يقوم بتحويل لغة ما (عالية المستوى)، مثلا لغة سي، إلى لغة التجميع (منخفضة المستوى)

## • الجامع

تطبيق يقوم بتحويل لغة التجميع (منخفضة المستوى)، إلى لغة الآلة (0, 1)

## • المفسر

تطبيق يقوم بتنفيذ شيفرة لغة مفسرة سطرا سطرا مباشرة، مثلا بايثون، وهذا بعكس المترجم والذي يقوم بتحويل لغة إلى أخرى

## • المكتبات

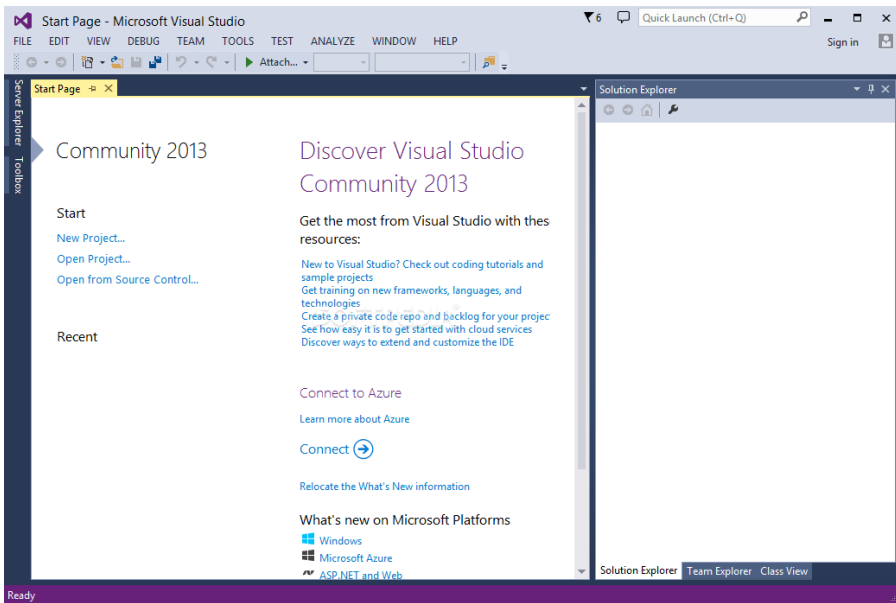
ملفات تحتوي على شيفرة دوال شائعة الاستخدام، مثلا دوال طباعة النصوص، دوال التعامل مع الملفات..، و الغرض منها هي بدلا من أن يكتب المبرمج الدوال لكل غرض، يكتفي باستعمال الدوال الموجودة في هته المكتبات

## • المنقح

تطبيق يقوم بالبحث عن الأخطاء البرمجية و يخبرك عنها و في أي سطر، و يعطيك معلومات إضافية تساعدك على فهم و تصحيح هذا الخطأ البرمجي

### ما هو تطبيق بيئة تطوير متكاملة ؟

تطبيق بيئة تطوير متكاملة هو عبارة عن مجموعة من التطبيقات و الأدوات التي تساعد المبرمج على استعمال لغة برمجة محددة، مثلا توفر للمبرمج تطبيق المترجم أو المفسر، مكتبات، منقح.. ، فبدلا أن يقوم المبرمج بتثبيت كل هته التطبيقات و الأدوات، يكتفي بتثبيت تطبيق واحد (بيئة تطوير متكاملة) و هو يحتوي على كل هته الأدوات



مايكروسوفت فيجيوال ستوديو هو عبارة عن بيئة تطوير متكاملة



---

## الفصل الثاني

---

### أساسيات في لغة ألف

### تعريف

لغة ألف (حرف أ)، هي لغة برمجة عربية تتميز بالبساطة و السهولة في استخدامها، مفتوحة المصدر، متعددة المنصات بواجهة احترافية، تدعم البايثون، جافا سكريبت، و سي بلاس بلاس

### تاريخ لغة ألف

لغة ألف هي حديثة العهد، حيث صممها مهندس البرمجيات حسن دراكة في كندا سنة 2018، لكن الفكرة بدأت حبرًا على ورق سنة 2010، و مرة بمراحل عديدة و انقطاعات حتى ظهرت أول نسخة سنة 2018

# الف	2
	3
نافذة رئيسية	4
	5
دالة رئيسية	6
	7
عدد ا = 4	8
عدد ب = 6 + ا	9
رسالة ("المجموع هو " + ب )	10
	11
نهاية دالة	12
	13
نهاية نافذة	14

## مميزات لغة ألف

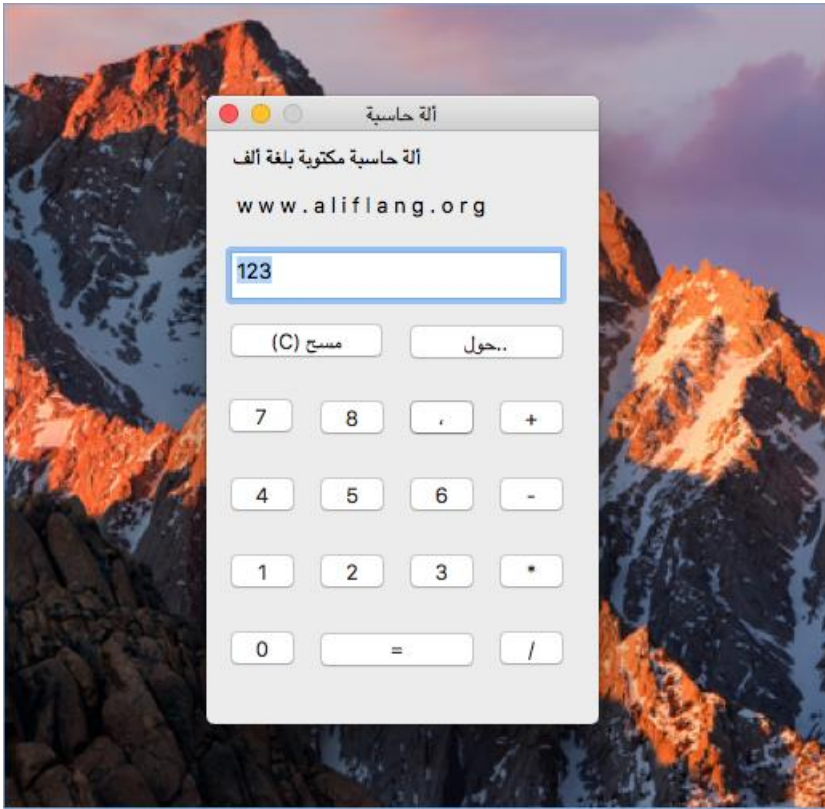
### • السرعة

تعتبر لغة ألف عمليا لغة سريعة جدا مقارنة بلغات عديدة مثل فيجوال بيسك دوت نت، جافا، بير، بايثون، سي شارب، و هذا راجع إلى عدت أسباب، منها أن لغات مثل بايثون و بير هي لغات مفسرة أي تحتاج إلى أن تُفسر باستخدام برنامج يدعى المفسر، وبالتالي تعتبر غير سريعة، أما لغات مثل فيجوال بيسك دوت نت، جافا، سي شارب، فهيا تنفذ على آلة افتراضية، وبالتالي تعتبر غير سريعة كذلك، أما

لغة ألف فهي مبنية على لغة سي++, و لغة سي++ تترجم إلى لغة الآلة، و تنفذ مباشرة من دون إي وسيط، وبالتالي تعتبر سريعة للغاية

## • دعم الواجهة الرسومية

لغة ألف تدعم الواجهة المستخدم الرسومية باللغة العربية، حيث يمكن تصميم تطبيقات تحتوي على نوافذ و أزرار مرئية للمستخدم، و هذه الخاصية لا تدعمها كثير من لغات برمجة عالمية إلا باستخدام مكتبات معقدة و صعبت الفهم و الاستيعاب



مثال على دعم الواجهة الرسومية

## • السهولة

لغة ألف لغة سهلة التعلم للغاية، أولاً لأنها عربية قلباً وقالباً، ثم أنها صممت بشكل تسهل على المبرمج الوصول للهدف، حيث تجعل المبرمج يركز على حل المشكل و الوصول للهدف أكثر من التركيز حول اللغة بحد ذاتها

## • متعددة المنصات

لغة ألف هي لغة متعددة المنصات، أي يمكن للمبرمج أن يكتب تطبيقا على نظام التشغيل ويندوز مثلا، ثم يستعمل نفس الشيفرة من دون أي تغيير لكتابة نفس التطبيق على نظام التشغيل ماك أوس، أو لينكس

### هل يصح مقارنة لغة ألف بلغات إنجليزية عالمية ؟

لا يصح ذلك، و إلا ستكون مقارنة غير عادلة، لأن لغة ألف بدأت بمجهود فردي و هي في طور التطوير، بينما اللغات الإنجليزية العالمية صممت بمجهود عالمي من مختلف المؤسسات، الأصح هو مقارنة لغة ألف بلغات البرمجة العربية الأخرى، أو بلغات الغير إنجليزية العالمية

### لماذا لغة ألف تعتمد على لغة سي++ ؟

إذا أردنا صناعة لغة برمجة مستقلة و لا تعتمد على أية لغة أخرى، لتوجب علينا إذا تصميم لغة تترجم من لغة عالية المستوى (لغة ألف) إلى لغة منخفضة المستوى

(لغة التجميع). و هذا يتطلب سنين طويلة من العمل و دعم من مؤسسات حكومية و فريق متكامل

بما أن المشروع بدأ بمجهود فردي، فظل مؤسس لغة ألف عدم اختراع العجلة و البدء من حيث انتهى الآخرون، حيث صمم لغة ألف لترجم من لغة عالية المستوى (لغة ألف) إلى لغة عالية المستوى أخرى (سي++) ثم إلى لغة منخفضة المستوى (لغة التجميع)، و استغرق هذا ثلاثة سنوات من العمل، فكانت النتيجة مذهلة و لقيت ترحيبا واسعا و انتشرت في مواقع التواصل الاجتماعي

## هل لغة ألف هي مجرد ترجمة حرفية ؟

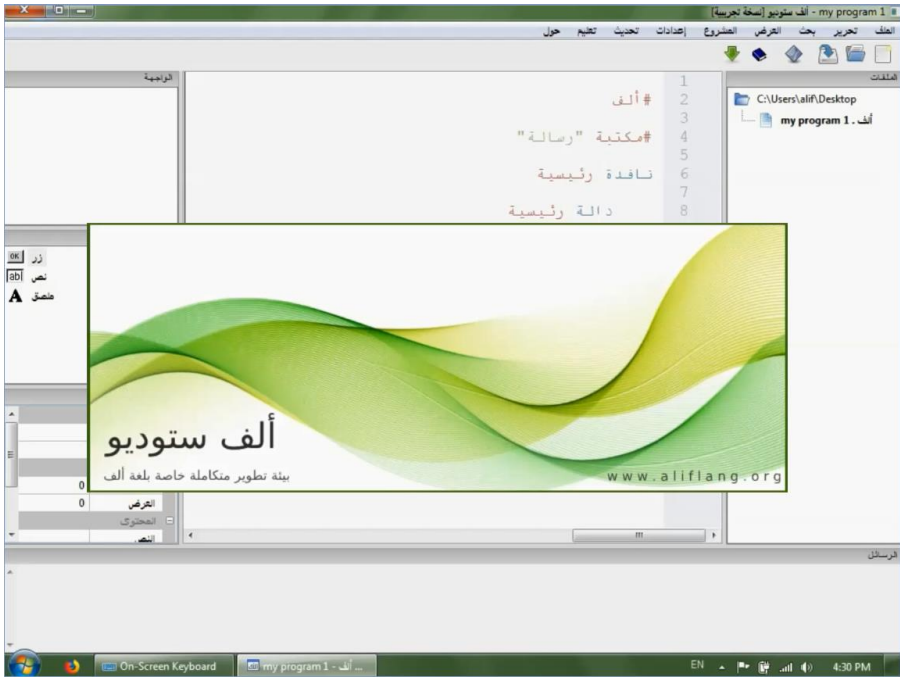
لا، فرغم أن لغة ألف مكتوبة بشكل كامل بلغة سي++، لن نجد أي علاقة بين لغة ألف و لغة سي++ في الشكل، فمثلا لغة ألف تنتهي أوامرها بنهاية السطر، بنما لغة سي++ تنتهي بـ '؛'، كذلك لا يمكن نقل المعلومات بين متغيرين مختلفين، أو إنشاء واجهة رسومية في سي++ إلا باستعمال مكتبات خاصة، بينما هي ممكنة و موجودة سلفا في لغة ألف، هذا كله لأن لغة ألف ليست ترجمة حرفية للغة سي++ أو أية لغة أخرى، بل صممت بشكل خاص و متميز حتى يسهل على الطلبة تعلمها بسهولة

## هل توجد بيئة تطوير متكاملة خاصة بلغة ألف ؟

نعم، تسما تطبيق "ألف ستوديو"، و هو تطبيق مجاني و سهل الاستعمال، متعدد المنصات، أي أنه يعمل على نظام التشغيل ويندوز، ماك أوس، و لينكس

يحتوي ألف ستوديو على مترجم ألف، المكتبات، منقح، و كل ما يحتاج المبرمج ليسهل عليه كتابة تطبيقات و تحرير ملفات الشيفرة، و كذلك يحتوي على تطبيق التحديث ليقوم بتحميل آخر إصدار





## ألف ستوديو

هو عبارة عن بيئة تطوير متكاملة خاصة بلغة ألف

لتحميل ألف ستوديو المرجو زيارة الموقع الرسمي:

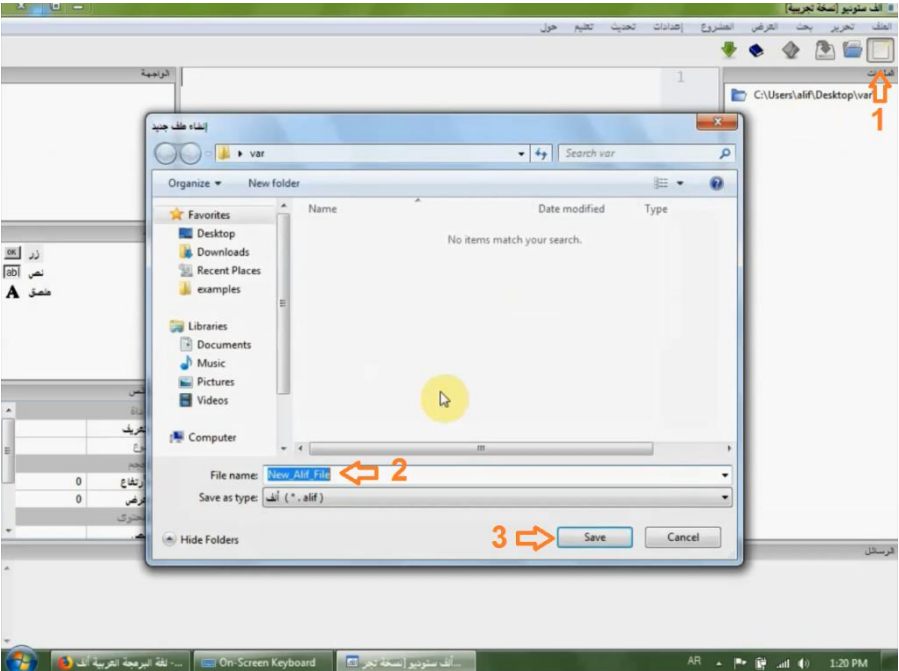
<https://www.aliflang.org>

# كيفية استعمال ألف ستوديو

حتى نتعرف على كيفية استعمال ألف ستوديو , سنقوم بإنشاء ملف جديد, ثم ترجمته و تنفيذه

## 1. الخطوة الأولى - إنشاء ملف جديد

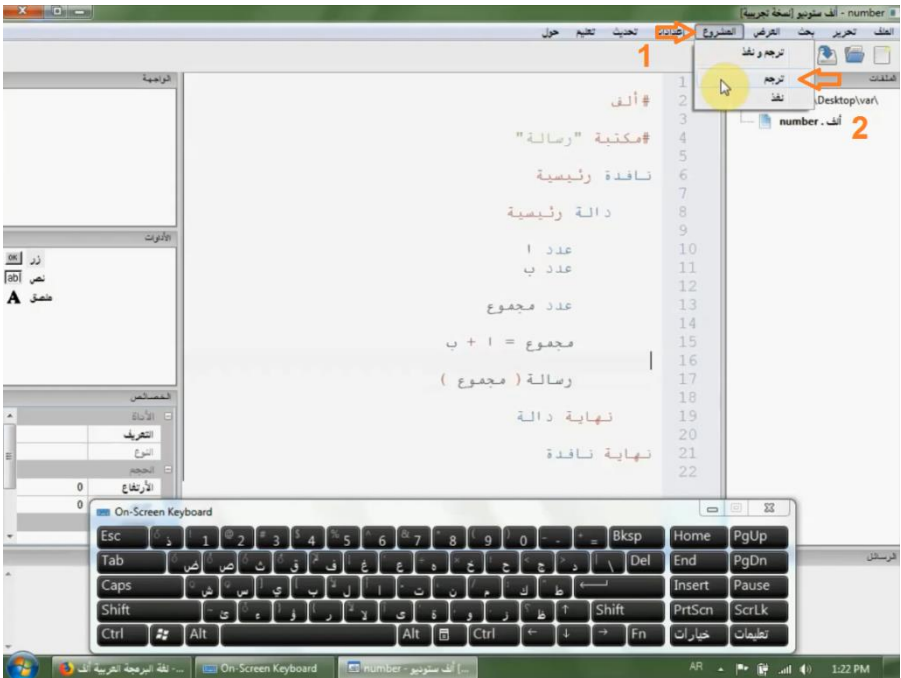
لا شك أنك قد قمت مسبقا بتحميل تطبيق ألف ستوديو, و تثبيته, الآن نقوم بتشغيله ثم إنشاء ملف جديد



- 1 - نضغط على زر "إنشاء ملف جديد"
- 2 - نختار اسم الملف
- 3 - نضغط على زر "حفظ" من أجل إنشاء الملف الجديد

## 2. الخطوة الثانية - الترجمة

هنا، يقوم مترجم ألف، بترجمة الشيفرة إلى لغة الآلة، فينتج لنا مترجم ألف ملف جديد عبارة عن ملف تنفيذي، و اللذي يعني ببساطة "تطبيق" أو "برنامج"



1 - نضغط على قائمة "المشروع"

2 - نضغط على اختيار "ترجم"

### 3. الخطوة الثالثة - تنفيذ

هنا، و بعد أن قمنا بترجمة الشيفرة، نعلم أن الملف تنفيذي قد تم إنشائه، لذي لم يبقى لنا سوى تنفيذه، أي تشغيله و تجربته

1 - نضغط على قائمة "المشروع"

2 - نضغط على اختيار "نفذ"

## الدخول إلى علوم البرمجة

قبل أن نشرع في كتابة أول شيفرة بلغة ألف، لابد من فهم منطق البرمجة أولاً، ثم إنشاء خوارزمية بسيطة

### منطق البرمج

أفضل طريقة لفهم منطق البرمجة و الدخول إلى علوم البرمجة، هي محاولة تصميم تطبيق يقوم بحل مشكل بسيط للغاية، على سبيل المثال حساب مجموع  $4 + 6$ ، ثم إعطائنا الجواب ليس إلا!

نريد معرفة مجموع رقمين، مثلاً  $4 + 6$

الآن لنتخيل ثلاثة صناديق!، الصندوق الأول نضع فيه الرقم 4، الصندوق الثاني نضع فيه الرقم 6، ثم نطلب من الحاسوب أن يحسب محتوى الصندوق الأول + محتوى الصندوق الثاني، ثم يضع النتيجة في الصندوق الثالث، إذا عملية سهلة للغاية!

4

الصندوق الأول

6

الصندوق الثاني

الصندوق الثالث

## الخوارزمية ستكون بهذا الشكل المبسط :

إنشاء الصندوق الأول

إنشاء الصندوق الثاني

إنشاء الصندوق الثالث

4 = الصندوق الأول

6 = الصندوق الثاني

الصندوق الثالث = الصندوق الأول +  
الصندوق الثاني

رائع، تبدو الفكرة سهلة و منطقية !

## تنبيه!

الخوارزمية هي فقط للشرح و تعلم منطق البرمجة بشكل مبسط، لا يمكن استعمال ألف ستوديو (مترجم ألف) لترجمة و تنفيذ الخوارزمية

بذلا من ذلك سنتعلم كيفية كتابة شيفرة بلغة ألف في الفصل الثالث (صفحة 43)

الآن نريد من الحاسوب أن يظهر لنا محتوى الصندوق الثالث على شكل رسالة مثلا، ستكون الخوارزمية بهذا الشكل



إنشاء الصندوق الأول  
إنشاء الصندوق الثاني  
إنشاء الصندوق الثالث

4 = الصندوق الأول  
6 = الصندوق الثاني

الصندوق الثالث = الصندوق الأول +  
الصندوق الثاني

إظهار رسالة فيها(محتوى الصندوق الثالث)

رائع !، قمنا بحساب مجموع عددين، ثم أظهرنا النتيجة  
على شكل رسالة، لكن قبل أن نشرع في التطبيق، يجب أن  
نعرف بضع نقاط بسيطة !

أولاً، مترجم ألف يبحث عن شعار خاص، ليتأكد من أن  
الشفيرة هي بلغة ألف، و هذا الشعار هو **#ألف**، لذى  
سنضيفه إلى الخوارزمية المبسطة فتصبح بهذا الشكل

## #ألف

إنشاء الصندوق الأول

إنشاء الصندوق الثاني

إنشاء الصندوق الثالث

4 = الصندوق الأول

6 = الصندوق الثاني

الصندوق الثالث = الصندوق الأول +  
الصندوق الثاني

إظهار رسالة فيها(محتوى الصندوق الثالث)

ثانيا، البرنامج اللذي نصممه يبدو مكتملا، فعلا فقد قمنا بحساب مجموع عددين، ثم أظهرنا النتيجة على شكل رسالة، لكن البرنامج مازال ينقصه الشكل الخارجي (الواجهة الرسومية)، أي النوافذ، الأزرار، الألوان.. ، لذى سوف ننشأ نافذة و نضع داخلها عملية الحساب اللتي صممناها سلفا

## #ألف

إنشاء نافذة رئيسية

إنشاء الصندوق الأول

إنشاء الصندوق الثاني

إنشاء الصندوق الثالث

4 = الصندوق الأول

6 = الصندوق الثاني

الصندوق الثالث = الصندوق الأول +  
الصندوق الثاني

إظهار رسالة فيها (محتوى الصندوق الثالث)

نهاية نافذة رئيسية

أخيرا، إذا قمنا بترجمة الشيفرة، ثم تشغيل التطبيق، فإن أول شئ يقوم به الحاسوب هو البحث عن نافذة رئيسية

و دالة رئيسية، أي البحث عن أين يقع أول سطر يجب تنفيذه قبل السطور الأخرى، لهذا السبب ننشئ دالة رئيسية و نضع داخلها عملية الحساب اللتي صممناها سلفا

## #ألف

إنشاء نافذة رئيسية

إنشاء دالة رئيسية

إنشاء الصندوق الأول

إنشاء الصندوق الثاني

إنشاء الصندوق الثالث

4 = الصندوق الأول

6 = الصندوق الثاني

الصندوق الثالث = الصندوق الأول +  
الصندوق الثاني

إظهار رسالة فيها(محتوى الصندوق الثالث)

نهاية دالة رئيسية

نهاية نافذة رئيسية

## ماذا يحدث عند الترجمة ؟

عند الترجمة، يقوم مترجم ألف بالبحث عن إشارة #ألف، فإن وجدها يقوم بترجمة الشيفرة إلى لغة الآلة

## ماذا يحدث عند تشغيل التطبيق ؟

عند تشغيل التطبيق يقوم الحاسوب بإنشاء النافذة الرئيسية، ثم يبحث عن الدالة الرئيسية، فإن وجدها يقوم بتنفيذ الشيفرة الموجودة داخل هته الدالة

---

## الفصل الثالث

---

### البرمجة بلغة ألف

### تذكير

في **الفصل الأول** من هذا الكتاب، تعرفنا على أساسيات عامة في البرمجة، نبذة عن تاريخ لغات البرمجة، و دور الأدوات المهمة في عملية الترجمة

في **الفصل الثاني** من هذا الكتاب، تعرفنا على أساسيات في لغة ألف، معلومات عامة عن لغة ألف، و كيفية استعمال ألف ستوديو، ثم كيفية كتابة خوارزمية بسيطة

الآن، بعد أن تعرفنا على الأساسيات، نستطيع كتابة أول شيفرة بلغة ألف و التعرف على مختلف مزايا هته اللغة



## تطبيق بسيط

أول شيفرة بلغة ألف، عبارة عن مثال بسيط، سنشرح هته الشيفرة بشكل سهل و مبسط، يمكنكم نسخ، و لصق هته الشيفرة في تطبيق ألف ستوديو، ثم ترجمتها و تنفيذها

#ألف

نافذة رئيسية

نهاية نافذة

## الشرح

#ألف

كل ملفات شيفرة ألف، تبدأ ب **#ألف** 'هاش ألف' و هو العلم أو الإشارة ألتتي تبين للمتجم أن هذا الملف هو عبارة ملف يحتوي على شيفرة بلغة ألف موجهة

للحاسوب، في الإصدارات القادمة إن شاء الله ستضاف  
#ألف\_أندرويد

من دون هذا العلم 'الإشارة' لن يتم المترجم عملية  
الترجمة

### نافذة رئيسية

إذا قمنا بترجمة الشيفرة، ثم تشغيل التطبيق، فإن أول  
شيء يقوم به نظام التشغيل هو البحث عن مدخل  
التطبيق، أي البحث عن أين يقع أول سطر يجب تنفيذه قبل  
السطور الأخرى، في لغة ألف، تعتبر النافذة الرئيسية هي  
مدخل التطبيق

كل التطبيقات المكتوبة بلغة ألف لابد أن تحتوي على  
نافذة واحدة معرفة بـ 'رئيسية'

### نهاية نافذة

أمر 'نهاية'، نقصد به نهاية شيء ما ثم إنشائه مسبقا،  
مثلا نهاية نافذة، نهاية دالة، نهاية إذا، نهاية صنف.. الخ،  
فهو تشبه '}' في لغات البرمجة الأخرى

## المتغيرات الرقمية

حاليا لغة ألف تدعم نوعا واحد من المتغيرات الرقمية (يأخذ 8 بايت في الذاكرة). هذا المتغير يقبل جميع أنواع الأعداد، الصحيحة و السالبة.. الخ، يمكن إذا إجراء جميع أنواع العمليات الحسابية باستعمال هذا المتغير الرقمي

للإعلان و إنشاء متغير رقمي نستعمل الأمر : **عدد**  
مثال : المتغيرات الرقمية

#ألف

عدد أ

عدد ب = 3.1415

أ = ب \* 2

النتيجة :

أ = 6.283

## المتغيرات النصية

المتغيرات النصية هي متغيرات تقبل كلمات، أرقام، جمل، حروف عربية، إنجليزية أو أي لغة أخرى، فهي متغيرات تستعمل بشكل عام لتخزين أو قراءة نص كتابي، و لا يمكن إجراء عمليات حسابية بواسطتها

للإعلان و إنشاء متغير نصي نستعمل الأمر : نص

مثال : المتغيرات النصية

#ألف

نص أ = "أحب "

نص ب = أ + "العلم"

النتيجة :

ب = أحب العلم

الدالة، بشكل مبسط هي عبارة عن مجموعة من الأوامر و العمليات الحسابية، فمثلا لنقل أننا نجري عملية حسابية طويلة باستعمال المتغيرات الرقمية، هذا يعني أننا سنكتب نفس العملية الحسابية في كل مرة نحتاجها، بذلك، نكتبها مرة واحدة داخل دالة، ثم ننادي على هذه الدالة في كل مرة نحتاجها

يوجد نوعين أساسيين من الدوال، النوع الأول يجري عملية حسابية، فقط، و تسمى دالة من نوع 'عادم'، النوع الثاني يجري عملية حسابية، ثم يرجع قيمة (الجواب)، و تسمى دالة من نوع 'عدد' أو 'نص'

مثال : دالة من نوع عادم

#ألف

دالة إيقاف\_تشغيل ()

...

نهاية إيقاف\_تشغيل

مثال : دالة من نوع عدد

#ألف

دالة عدد حساب\_مساحة\_المربع (عدد ط, عدد ع)

إرجاع ط \* ع

نهاية حساب\_مساحة\_المربع

عدد م = حساب\_مساحة\_المربع(6, 4)

النتيجة :

م = 24

مثال : دالة من نوع نص

#ألف

دالة نص ترحيب (نص اسم)

إرجاع " أهلا بك " + اسم

نهاية ترحيب

نص ج = ترحيب ("محمد")

النتيجة :

ج = أهلا بك محمد



## الجمل الشرطية

أحيانا، لا نريد أن يتم تنفيذ شيفرة ما، إلا إذا تحقق شرط محدد، مثلا إذا كان العمر أكبر من 13 سنة، فنكتبها على هذا الشكل " إذا العمر < 13 "

نستطيع أن نستعمل الأوامر التالية في الجمل الشرطية

**إذا** ( [شرط] و [شرط] ) **أو** [شرط]

... الشيفرة ...

**أو إذا** [شرط]

... الشيفرة ...

**وإلا**

... الشيفرة ...

**نهاية إذا**

#ألف

عدد العمر = 21

نص الجواب

إذا العمر < 13 و العمر > 18

الجواب = " ما بين 13 و 18 سنة "

أو إذا العمر < 18

الجواب = " أكبر من 18 سنة "

وإلا

الجواب = " أصغر من 13 سنة "

نهاية إذا

النتيجة :

الجواب = أكبر من 18 سنة

## الجمل التكرارية

أحيانا، نود أن يتكرر تنفيذ شيفرة ما عدة مرات، و لا تتوقف حتى لا يتحقق شرط محدد، هذه العملية تسمى الجمل التكرارية، و هي تكتب على هذا الشكل :

كلما [شرط]

... الشيفرة ...

نهاية كلما

مثال : الجمل التكرارية

#ألف

عدد ج = 0

نص ك

كلما ج > 10

ك = ك + "ج + "

ج = ج + 1

نهاية كلما

النتيجة :

ك = [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

في الإصدار القادم إن شاء الله من هذا الكتاب :

- شرح كيفية البرمجة الكائنية بلغة ألف
- شرح كيفية استعمال الواجهة الرسومية
- شرح كيفية استعمال المكتبات
- شرح كيفية استعمال أنواع أخرى من المتغيرات
- شرح كيفية استعمال أنواع أخرى من الجمل التكرارية
- شرح كيفية برمجة مكتبات جديدة
- شرح كيفية دمج شيفرة سي++ مع شيفرة ألف
- شرح كيفية دمج شيفرة جافا سكريبت مع شيفرة ألف
- شرح كيفية دمج شيفرة البايثون مع شيفرة ألف

إذا كانت لديكم اقتراحات أو استفسارات، المرجو زيارة  
المنتدى الاجتماعي الرسمي :

<https://www.aliflang.org>

شكرا لكم،



©2020 DRAGA Hassan

[www.aliflang.org](http://www.aliflang.org)

Canada