# ماذا تعرف عن لغة البرمجة الشهيرة C

لغة البرمجة C هي لغة شعبية ومنتشرة بشكل لا يصدق، ومن السهل معرفة سبب ذلك. حيث تعتبر البرمجة بلغة C فعّالة، وتمنح المبرمج قدراً كبيراً من السيطرة. وقد وضِعَت العديد من لغات البرمجة الأخرى مثل **C++**، جافا **Java**، وبايثون **Python** باستخدام C.

في حال كنت مبرمجاً، فإنّ فرص عدم استخدامك للغة البرمجة C على وجه الخصوص تتزايد بشكل يومي، إلّا أنّ هنالك العديد من الأسباب التي تجعل من لغة البرمجة C مفيدة للغاية، حتى لو لم تكن تستخدم هذه اللغة بانتظام. وإليك هذه الأسباب:

سيكون بإمكانك قراءة وكتابة التعليمات البرمجية (الكود) للبرمجيّات التي يمكن استخدامها على أنواع مختلفة ومتعدّدة من المنصّات

الحاسوبيّة، بدءاً من المتحكّمات المصغّرة **Microcontrollers** وحتّى الحواسيب المكتبيّة، والحواسيب المحمولة وأنظمة تشغيل الهواتف المحمولة.

سيكون بمقدورك أن تدرك بشكل أفضل ما تفعله اللغات عالية المستوى في الظل، مثل إدارة الذاكرة وتحرير خلاياها من المعلومات غير المستخدمة والخاصّة بالبرامج المغلقة، وذلك من أجل زيادة مقدرة الحاسوب على تشغيل برامج أكثر في آنٍ معاً **Garbage Collector**. وهذا الإدراك يمكن أن يساعدك على كتابة برامج تعمل بكفاءة أكبر.

إذا كنت متخصصًا في تكنولوجيا المعلومات IT، فقد تستفيد أيضاً من تعلّم لغة البرمجة C. حيث يقوم محترفو تكنولوجيا المعلومات غالباً بكتابة وصيانة وتشغيل برامج نصيّة **scripts** كجزء من عملهم اليومي. البرنامج النصي هو مجموعة من التعليمات التي يتوجّب على نظام التشغيل الحاسوبي اتباعها.

من أجل تشغيل بعض البرامج النصية، يضع الحاسوب بيئة تنفيذ متحكّم بها تدعى **Shell**. وبما أنّ معظم أنظمة التشغيل تشغّل الـ**Shells** بالاعتماد على لغة **C**، فإنّ الكتابة بلغة **C** ـبالتاليـ تعد عملاً طبيعيًا يقوم به المتخصصون بتكنولوجيا المعلومات.

تغطي سلسلة المقالات هذه التاريخَ وراء لغة **C**، كما تشرح سبب تصنيف لغة البرمجة **C** من اللغات المهمّة، وتطرح أمثلة مكتوبة بلغة **C** كما تستكشف بعض السمات الهامة لهذه اللغة، بما في ذلك أنواع البيانات، والعمليات، والدوال (التوابع)، والمؤشرات وإدارة الذاكرة. وبالرغم من أنّ هذه السلسلة ليست دليلاً إرشاديًا للبرمجة بلغة **C**، إلّا أنّها توضّح الأسباب التي تجعل البرمجة بلغة **C** فريدة من نوعها بشكل مغاير عن الطرق التقليدية التي تبدأ فيها الفصول الأولى من الكتب المتخصصة بالبرمجة بلغة **C** في العادة.

دعونا نبدأ بالكشف عن المصدر الذي جاءت منه لغة البرمجة **C**، نبيّن كيف تطوّرت وما هو الدور الذي تلعبه في مجال تطوير البرمجيات اليوم.

## ما هي لغة البرمجة C

---

إنّ أبسط طريقة لتعريف لغة البرمجة **C** هي أن ندعوَها بلغة البرمجة الحاسوبيّة، ما يعني أنّه بإمكانك كتابة برامج يستطيع الحاسوب تنفيذها باستخدام هذه اللغة. قد تكون النتائج تطبيقات حاسوبيّة كبيرة من حيث الحجم والأداء، كمتصفح الويب الخاص بك، أو مجموعة صغيرة من التعليمات مضمنةً في معالج مصغّر أو مكونات حاسوبيّة أخرى.

تم العمل على إنتاج لغة **C** في سبعينيات القرن الماضي في مختبرات بيل، ويعود الفضل في المقام الأول لعمل كين تومسون **Ken Thompson** ودنيس ريتشي **Dennis Ritchie**. كان الهدف من تطويرها هو حاجة المبرمجين إلى مجموعة من التعليمات الأكثر سهولة في التعامل مع نظام التشغيل **UNIX**، والذي كان يتطلّب ـفي ذلك الوقتـ كتابة التعليمات بلغة التجميع **Assembly**. البرمجة بلغة التجميع ليست بالعملية السهلة والبسيطة، لأنّ تعليماتها تُخاطب بنية الحاسب بشكل مباشر، ما يجعل من الصعوبة بمكان كتابة البرنامج وتصحيحه، كما أنّها عملية مملة وتستهلك وقتًا كبيرًا من أجل إضافة وتعديل البرنامج ليقوم بمهمة معينة، حتى وإن كانت بسيطة. [المصدر: **King**]

كانت محاولة تومسون الأولى لعمل لغة برمجة عالية المستوى تُسمى B، تعتمد بشكل أساسي على لغة البرمجة **BCPL**.
وعندما احتاجت مختبرات **Bell** نظام التشغيل **Unix** نموذج **PDP-11**، قام تومسون بإعادة العمل على لغة **B** لتتوافق أكثر مع

المتطلبات الجديدة لنظام حاسوبي أجدد وأفضل. ومن هنا تمت ولادة لغة **C** وريثة لغة **B** في العام 1973. كانت لغة C مستقرة كفاية لدرجة يمكن فيها إعادة كتابة نظام **Unix** بحد ذاته باستخدام هذه اللغة الواعدة في تلك الآونة. [المصدر: **King**].

قبل استخدام مختبرات **Bell** للغة C بشكل فاعل، كانت هناك حاجة من قبل مبرمجين آخرين لدليل يشرح استخدام هذه اللغة. فخلال عام 1978 ومن خلال كتاب "البرمجة بلغة **C Programming Language** "C **The C** من قِبل بريان كيرنيغان **Brian Kernighan** وريتشي، والمعروف من قبل الهواة بـ **K&R** أو "الكتاب الأبيض"، حيث أصبح المصدر الرئيسي للبرمجة بلغة **C** حتى كتابة هذه السطور. والطبعة الثانية من **K&R** والتي نشرت أصلاً في عام 1988، لاتزال متاحة على نطاق واسع. ويطلق على الإصدار الأصلي **C R & K** بناء على هذا الكتاب.

لضمان عدم نشوء صيغ مختلفة للغة C من قبل المهتمين بها، وضع مطورو هذه اللغة في ثمانينيات القرن الماضي أساسيات ومعايير لها. فمعيار الولايات المتحدة للغة C، المعهد القومي الأمريكي للمعايير ANSI X3.159–1989، الذي أصبح كذلك رسميًا في عام 1989. وتبعه معيار المنظمة الدولية للمعيار رقم ISO/IEC 9899:1990. إصدارات C بعد K & R اعتمدت في مرجعيتها المعايير السابقة وكذلك الإصدارات اللاحقة أيضًا مثل C89, C90 and C99. يمكن ملاحظة أن C89 يطلق عليها أحيانًا **C ANSI** أو **ANSI/ISO** أو **C ISO**.

كانت لغة C واستخدامها في **UNIX** مجرد جزء واحد من الطفرة في تطوير نظام التشغيل خلال ثمانينيات القرن الماضي. على الرغم من جميع التحسينات التي تميزت بها لغة C عن سابقاتها من اللغات الأخرى، إلّا أنّها ظلت غير مناسبة لتطوير تطبيقات برمجية أكبر. ومع تطور صناعة الحواسيب زادت الحاجة لعملية إنتاج البرمجيات بشكلٍ أسهل، ما دفع المبرمجين لبناء مترجمات **compilers** خاصة بهم، والعمل، بالتالي، على إنتاج لغات برمجة أخرى باستخدام لغة البرمجة C. هذه اللغات الجديدة بسّطت عملية إنتاج البرامج المعقدة في ذلك الآن. على سبيل المثال، لغتي **Java** و **C++** كلاهما طورتا من لغة C وجعلتا تطوير البرامج المبنية على الكيانات (تقنية برمجية قائمة على إعادة استخدام نصوص البرامج) أمراً سهلاً.

الآن وبعد أن ألقينا نظرة على خلفية الأحداث، لنلقِ نظرة على آليات لغة C.

**كتابة وترجمة تعليمات لغة C**

---

تعرف C على أنّها لغة تعتمد على مترجم، أي أنك بحاجة لمترجم لتحويل البرنامج المصدر (الكود البرمجي) إلى برنامج قابل للتنفيذ على الحاسب. يكتب البرنامج المصدر بأحد برامج تحرير النصوص، والذي يتيح للمبرمج إمكانيّة فتح وقراءة البرنامج المكتوب بلغة C. ومن الأمثلة على برامج تحرير النصوص، برنامج المفكرة الذي يأتي مع نظام التشغيل **Windows**، وكذلك **TextEdit** في نظام **MAC**، وأيضا **gedit** في نظام **UNIX**.

الملف التنفيذي **executable file** هو ببساطة: البرنامج الذي يتم تنفيذه على الحاسب. يفحص مترجم لغة C البرنامج المصدر، وفي حالة كان خالياً من الأخطاء، تتم ترجمته إلى ملف تنفيذي.

قبل البدء بالحديث عن البرنامج المكتوب بلغة C، يجب أن نلقي نظرة على المترجمات المتوفرة. إذا كنت من مستخدمي أنظمة التشغيل **MAC OS X H** أو أحد منشورات نظام **UNIX** مثل **Ubuntu**، فإنه من الممكن أن تنصب مترجم لغة C لحاسوبك إذا قمت بتنصيب مستلزمات إنتاج برامج بلغة C لنظام التشغيل الذي تعتمده.

مترجمات لغة C المجانية تعتمد بالأساس على نظام تلقين الكمبيوتر الأوامر بشكل مكتوب، ما يعني أنّك ستشغّلهم من خلال موجه الأوامر الذي يمكن الوصول له من خلال نافذة الطرفية. أحد أوامر ترجمة برنامج مكتوب بلغة الآلة هو cc أو **gcc**، بالإضافة إلى بعض الخيارات والمعلومات الممرّرة والتي تمثل بعض الكلمات المستخدمة كخيارات تساعد في عملية الترجمة، وذلك قبل الضغط على **Enter** للبدء بتنفيذ عملية الترجمة.

اذا كنت من مستخدمي نظام التشغيل **Windows** أو كنت من محبي استخدام الواجهة الرسومية ففضلاً عن إعطاء أوامر مكتوبة، يمكنك أن تُثبّت بيئة التطوير المُتكاملة (**IDE**) لبرمجة C على جهاز الكمبيوتر. بيئة التطوير المتكاملة هي واجهة واحدة بحيث يمكنك كتابة التعليمات البرمجية، ترجمة البرنامج، اختباره والعثور بسرعة على الأخطاء وإصلاحها. بالنسبة لنظام التشغيل **Windows** يمكنك شراء برنامج **Microsoft Visual C++** وهو بيئة تطويرية متكاملة لكل من C و C++. كذلك برنامج **Eclipse** يعتبر من البرامج الجيدة للبرمجة بلغة C لأنّه مجاني ويعتمد على **Java** ويعمل على بيئات تشغيل مختلفة مثل **Windows** و **MAC** و **Linux** كما أنّ له برامج فرعية متاحة لعملية ترجمة برامج مكتوبة بلغة C والعديد من اللغات أخرى.

في لغة C كما في لغات البرمجة الأخرى، يكون رقم إصدار المترجم أمرًا غاية في الأهميّة. فالمبرمج يرغب دائمًا استخدام نسخة من مترجم لغة C أحدث من نسخة اللغة نفسها. إذا كنت تستخدم IDE، تأكد من ضبط الإعدادات الخاصة بك للتأكد من أن الـIDE يستخدم بالنسخة الصحيحة من لغة C. إذا كنت ممن يفضل ونتزويد الكمبيوتر بأوامر نصيّة، فيمكنك تزويده بالمعلومات اللازمة لترجمة البرنامج بشكل صحيح، مثل:

```
gcc –std c99 –o myprogram.exe myprogram.c
```

في الأمر أعلاه، يعد gcc الأمر الرئيسي الذي يقوم بالترجمة الفعليّة، وأمّا ما تبقى من الأمر فهي خيارات لضبط عملية الترجمة من البرنامج المصدر للبرنامج التنفيذي. تمت إضافة خيار ـstd متبوعاً بـ c99 لإعلام المترجم باستخدام النسخة القياسية للغة رقم c99 في عملية الترجمة. الخيار o– والذي يليه **myprogram.exe** يعنيان تحديد اسم الملف الناتج عن عمليّة الترجمة، وبدون هذا الخيار فإنّ المترجم يستخدم الاسم a.out بدلاً من الاسم المحدد في الأمر أعلاه بشكل تلقائي. والخيار الأخير **myprogram.c** يحدّد اسم البرنامج المصدر المراد ترجمته والمكتوب بلغة C. وباختصار، الأمر أعلاه يطلب من مترجم C تحويل البرنامج myprogram.c إلى ملف تنفيذي باسم **myprogram.exe** بواسطة المترجم gcc واعتماد المعيار c99 في عملية الترجمة.
يمكنك تصفح الويب للحصول على قائمة كاملة من الخيارات التي يمكنك استخدامها مع المترجم الخاص بك، سواء أكان ذلك gcc أو أي مترجم آخر.

بعد تثبيت المترجم على حاسوبك، أنت الآن جاهز للبرمجة بلغة C دعنا نبدأ بالتعرف على أبسط برنامج مكتوب بلغة C.

**أبسط برنامج مكتوب بلغة C**

---

دعنا نلقي نظرة على برنامج بسيط بلغة C ونستخدمه للتعرّف على أساسيات اللغة وعلى عملية ترجمته. إذا كان لديك جهاز الحاسوب الخاص بك مع مترجم C مثبتاً كما هو موضح سلفاً، يمكنك إنشاء ملف نصي يدعى **sample.c** واستخدامه للمتابعة معنا بتركيز بالخطوات التي سنقوم بشرحها. لاحظ أنّه يجب أن يكون امتداد الملف ".C" وأن لا يكون "**txt**." لأنّ ذلك سوف يؤدي إلى ظهور أخطاء أثناء الترجمة.
ها هنا برنامج لغة الـ C:

```
Sample program */ */
#include <stdio.h>
int main()
{
printf("This is output from my first program!\n");
;return 0
{
```

عندما تتم ترجمته وتنفيذه، فإن هذا البرنامج يرشد الكمبيوتر لطباعة الجملة " This is output from my first program!" ثم يتوقف.
لا يمكن كتابة برنامج أبسط من ذلك! الآن دعونا نلقي نظرة على ما يقوم به كل سطر:

السطر 1: هذه طريقة لكتابة التعليقات في البرنامج باستخدام /* و */ على سطر واحد أو أكثر.
السطر 2: يخبر المترجم بأنّ عليه النظر بمصادر أخرى لنصوص برمجيّة مكتوبة بلغة C، وبشكل خاص: المكتبات، وهي ملفات تتضمّن تعليمات شائعة قابلة لإعادة الاستخدام، وهنا يخبره بتضمين مكتبة بعنوان stdio.h والمستخدمة لعمليات القراءة من لوحة المفاتيح (تلقي مدخلات من المستخدم) والطباعة على شاشة الكمبيوتر. (هذه المكتبة تحتوي على توابع وتعليمات جاهزة للاستخدام من قبل المبرمج). سوف نتحدث عن المكتبات بشكل أوسع لاحقاً.
السطر 3: هو بداية التابع المُسمى **main**، حيث أنّ كل برنامج C يجب أن يحوي على الأقل تابعاً واحداً أو كتلة من التعليمات التي تمثل ما سيقوم الحاسوب بتنفيذه عند تشغيل البرنامج. بشكل عام ينفّذ التابع بعض المهام مع إمكانيّة إعادة قيمة تمثل الجواب النهائي لتنفيذ هذه المهام، والتي يمكن استخدامها من قبل التوابع الأخرى.
يجب أن يحوي برنامج C تابع وحيد على الأقل وهو **main** (التابع الرئيسي) كالموضّح في المثال أعلاه، مع قيمة معادة من النوع عدد صحيح **Integer**.
سنتعرّف على دور الأقواس الفارغة بعد أن نتفحص التوابع لاحقاً.

الأسطر 4 و7: تعليمات التابع تبدأ وتنتهي بهذه الأقواس. بعض المبرمجين يفضلون أن يضعوا هذه الأقواس على أسطر مستقلّة في البرنامج. والبعض الآخر يضع القوس الأول (قوس الفتح) في نهاية السطر الأول من تعريف التابع، وعلى الرغم من أنّ الأسطر في الكود البرمجي لا يُشترط أن تكون مفصولة، إلّا أنّ المبرمجين عادة ما يضعون كل تعليمة بسطر مستقل مسبوقة بفراغات مناسبة، بهدف جعل عملية قراءة وفهم وتعديل البرنامج في وقت لاحق أمراً سهلاً سواء من المبرمج نفسه أو من قبل مبرمجين آخرين.
السطر 5: هذه الجملة تمثل استدعاء تابع يدعى **printf** وهذا التابع معرّف في الملف **stdio.h** الذي تم تضمينه في البرنامج الحالي في السطر الأول، وبالتالي فليس هناك داعٍ لكتابته من قبل المبرمج، إنّ هذا الاستدعاء يخبر المترجم بما يجب عليه أن يطبع على شاشة الحاسب.
وجود /n في نهاية الجملة ضمن إشارتي الاقتباس لا تتم طباعتها على الشاشة، وانما تمثل أمرًا للحاسب بنقل مؤشر الطباعة إلى السطر التالي للسطر الحالي.
كما تلاحظ أيضاً أنّ كل سطر تعليمة بـ C ينتهي بفاصلة منقوطة.

السطر 6: كل تابع يعيد قيمة ما يجب أن ينتهي بالأمر **return**. في لغة C يتوجّب دائماً على التابع الرئيسي **main** أن يعيد قيمة من نوع عدد صحيح، حتّى وإن لم يتم استخدامها من قبل البرنامج.
لاحظ أنك عندما تقوم بتنفيذ برنامج ما بلغة الـ C، فإنك تنفذ التابع الرئيسي **main**، فعندما تنفذ البرنامج تستطيع فحص ما يقوم بإعادته.

يفضل عادة وضع قيمة ليعيدها التابع الرئيسي 0 حيث أنّ المبرمجين عادة ينظرون لتلك القيمة في اختبار التأكد من البرنامج يعمل بشكل ناجح.

عندما تكون على استعداد لاختبار برنامجك،لا تنسَ حفظ الملف والنقر على ترجمة وتشغيل البرنامج.
إذا كنت تستخدم مترجم **gcc** في سطر الأوامر، والبرنامج موجود في ملف يسمى **sample.c**، يمكنك ترجمته من خلال الأمر التالي:
**gcc -o sample.exe sample.c**
إذا لم يكن هنالك أخطاء في البرنامج، يتم إنتاج برنامج باسم **sample.exe** في نفس المجلد الموجود فيه البرنامج المصدري **sample.c**.
إنّ أكثر الاخطاء الشائعة هي الأخطاء الاملائية والتي تعني أنك قمت بكتابة تعليمة ما بشكل خاطئ، كنسيان طباعة الفاصلة المنقوطة في نهاية السطر أو عدم إغلاق قوس من الأقواس أو إشارات الاقتباس. إذا كنت بحاجة لإجراء بعض التعديلات على الأخطاء، فقم بفتح الملف المصدر من خلال أحد برامج تحرير النصوص، وأجرِ تعديلك على الخطأ،ومن ثم احفظ التغييرات التي أجريتها وحاول أن تقوم بالترجمة مرة أخرى.
لتشغيل البرنامج السابق **sample.exe** أدخل الامر التالي:
./sample.exe
/. نجبر الحاسوب على البحث في المجلد الحالي لإيجاد الملف التنفيذي ويقوم بتنفيذه.
تلك هي أساسيات الكتابة والترجمة بلغة C، على الرغم من أنّ هنالك الكثير لتتعلمه عن ترجمة البرامج من الكثير من المصادر المتعلقة بلغة C.
يتضمّن الجزء الثاني من السلسلة ما تتضمنّه لغة الـ C من بنى لبناء البرنامج.

- التاريخ: 2016-08-01
- التصنيف: تكنولوجيا

#برمجة #تكنولوجيا #لغات البرمجة

---

المصطلحات

- **الكسوف (Eclipse)**: عند مرور جسم سماوي أمام جسم آخر، فإنه يقوم بحجب ضوء الجسم الثاني (على سبيل المثال، حجب ضوء الشمس بوساطة القمر، أو قيام أحد النجوم الموجودة في نظام ثنائي بخسف النجم الثاني). قد يكون أيضا عبارة عن مرور كامل الجسم أو جزء منه ضمن ظل جسم آخر (على سبيل المثال، خسوف القمر و خلاله يقوم القمر بالمرور ضمن ظل الأرض).

**المساهمون**

- ترجمة
  - أمجد هواش

- مُراجعة
  - محمد اسماعيل باشا
- تحرير
  - أنس الهود
  - بنان محمود جوابره
- تصميم
  - Tareq Halaby
- نشر
  - أنس الهود