$ g++ zero_to_builder.cpp -o builder

النسخة العربية



C++ · FROM ZERO

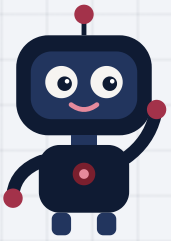
ZERO TO BUILDER

كتاب C++ الذي يشرح **كل سطر** ويعلّمك أن **تبني**، لا أن تحفظ.

ماذا يوجد بالداخل؟

- شرح كل سطر بالتفصيل
- 8 مشاريع حقيقية
- كويز في كل فصل
- تمارين بـ 3 مستويات
- الذاكرة والمؤشرات بالرسم
- امتحان نهائي + الإجابات
- ركن الأخطاء: رسائل حقيقية
- نصائح للتعلم بالذكاء الاصطناعي
- الأكواد والحلول كاملة

اللغة التي تجعل البرامج سريعة جداً.



بِت · مرشدك

مرحباً، أنا **بِت**. سأكون معك في كل صفحة. يقولون إن C++ صعبة، ومهمتي أن أجعل كل سطر فيها واضحاً. اتفقنا؟

C++ (تُنطق: "سي بلس بلس") لغة برمجة لبناء برامج يجب أن تكون **سريعة جداً** وتستخدم ذاكرة الكمبيوتر بعناية. عندما تعرض لعبةٌ 60 صورة في الثانية، أو يفتح المتصفح صفحة ثقيلة في لحظة، غالباً تكون C++ هي التي تعمل في الخلفية.

المعلومة	التفاصيل
من صنعها؟	بيارنه ستروستروب، عالم كمبيوتر دنماركي، في معامل Bell Labs.
متى؟	بدأ العمل عليها سنة 1979 باسم "C with Classes"، وتغيّر اسمها إلى C++ سنة 1983، وصدرت أول نسخة سنة 1985.
لماذا هذا الاسم؟	في لغة C، العلامة `++` معناها "زِد واحداً". فـ C++ معناها "C لكن أفضل بخطوة".
ما نوعها؟	لغة **تُترجَم مسبقاً** (compiled): الكود يتحول إلى ملف برنامج قبل أن يعمل. (الصفحة 4 تشرح ذلك.)

أفضل 4

ظلت C++ ضمن أفضل 4 لغات في مؤشر TIOBE للشعبية طوال 2026، بجانب Python وC وJava.

المصدر: مؤشر TIOBE، من يناير إلى أغسطس 2026

أنت تستخدم C++ كل يوم.

أنت لا ترى C++، بل ترى ما تشغّله. هذه أماكن حقيقية مكتوب فيها أجزاء كبيرة ومهمة بلغة C++:

محركات الألعاب
محرك **Unreal Engine** مكتوب أساساً بـ C++، وتُبنى به ألعاب كبيرة كثيرة.

متصفحات الإنترنت
Chrome و**Firefox** يستخدمان C++ في الجزء الذي يرسم صفحات الويب.

أنظمة التشغيل
أجزاء كبيرة من **Windows** مكتوبة بـ C و++C.

برامج التصميم
Adobe Photoshop يستخدم C++ في معالجة الصور الثقيلة.

قواعد البيانات
MySQL مكتوبة بـ C و++C لتتعامل مع كميات بيانات ضخمة.

الأجهزة الصغيرة
لوحات **Arduino** والروبوتات وأنظمة السيارات غالباً تعمل بـ C أو ++C.

لماذا يختارون C++؟

السرعة
تتحول إلى لغة الآلة، فتعمل قريبة جداً من الهاردوير.

التحكم
أنت تقرر كيف ومتى تُستخدم الذاكرة.

تعيش طويلاً
كود C++ مكتوب منذ سنوات ما زال يعمل اليوم.

أنظمة ضخمة
الـ Classes تنظّم برامج فيها ملايين الأسطر.

Elmasryx

ماذا تستطيع أن تفعل بـ C++، وماذا لا.

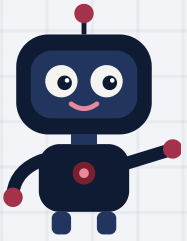
✅ بـ C++ تستطيع...

- بناء **ألعاب** ومحركات ألعاب (Unreal Engine يستخدم C++).
- كتابة برامج تحتاج **أقصى سرعة**: فيديو، 3D، محاكاة.
- برمجة **الهاردوير**: Arduino، روبوتات، حساسات.
- دخول **مسابقات البرمجة** (أغلب المتسابقين يستخدمون C++).
- فهم **كيف يعمل الكمبيوتر فعلاً**: الذاكرة والعناوين.

⚠️ بدون C++ (أو ليست الأداة الأفضل)...

- **المواقع**: المتصفح يشغّل JavaScript وليس C++.
- **السكربتات السريعة وتجارب الذكاء الاصطناعي**: Python أسرع في الكتابة.
- **تطبيقات الموبايل**: Android يستخدم Kotlin، وiPhone يستخدم Swift.
- **المهام المكتبية البسيطة**: C++ مجهود كبير لشغل صغير.
- تعلّمها أبطأ: أنت تتحكم في أشياء أكثر، فتكتب أكثر.

	Python	JavaScript	C++
السرعة وقت التشغيل	أبطأ	سريعة	**سريعة جداً**
سهلة للمبتدئين	**الأسهل**	سهلة	أصعب
طريقة التشغيل	تُقرأ أثناء التشغيل	تُقرأ أثناء التشغيل	**تُترجَم** لملف برنامج
تتحكم في الذاكرة؟	لا	لا	**نعم، تستطيع**
الأفضل في	الذكاء الاصطناعي والبيانات	المواقع وتطبيقات الويب	الألعاب والأنظمة والسرعة



نصيحة بِت

تعلّم C++ يجعل **كل لغة بعدها أسهل**. بعدها ستشعر أن Python وJavaScript إجازة!

كودك يتحوّل إلى برنامج حقيقي.

تقرآن كودك Python وJavaScript **سطراً بسطر أثناء التشغيل**. أما C++ فمختلفة: أداة اسمها **المترجِم (compiler)** تقرأ كودك **كله** أولاً، وتفحصه، ثم تحوّله إلى ملف برنامج يشغّله الكمبيوتر مباشرة.

1. أنت تكتب — `hello.cpp`
الكود في ملف نصي ينتهي بـ `.cpp`.

←

2. المترجِم — `g++`
يفحص كل سطر، ثم يحوّله للغة الآلة.

←

3. ملف البرنامج — `hello.exe`
تطبيق حقيقي. على Mac/Linux اسمه `hello` فقط.

←

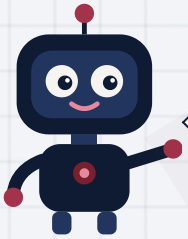
4. يعمل — `▶ run`
الكمبيوتر يشغّله مباشرة، وبسرعة كبيرة.

وقت الترجمة ووقت التشغيل

	وقت الترجمة (Compile time)	وقت التشغيل (Run time)
متى؟	عندما تبني البرنامج (الخطوة 2)	أثناء عمل البرنامج (الخطوة 4)
مثال على خطأ	نسيت `;` ← المترجِم يرفض	القسمة على صفر ← البرنامج ينهار
الخبر الجميل	أخطاء كثيرة تُكتشف **قبل** أن يستخدم أحد برنامجك	مفاجآت أقل، لأن المترجِم فحص أولاً

تذكّر

إذا غيّرت الكود يجب أن **تترجمه مرة أخرى**. ملف البرنامج القديم لا يتحدث وحده.



بِت

تخيّل مترجماً يترجم **الكتاب كله** قبل طباعته. يبدأ ببطء، لكن الكتاب المطبوع يُقرأ بسرعة. هذه هي C++.

جاهز خلال 15 دقيقة.

تحتاج شيئين مجانيين: **محرر** تكتب فيه الكود (VS Code)، و**مترجِم** يبنيه (g++). اختر نظامك:

1 نزّل VS Code

نزّله مجاناً من **code.visualstudio.com**. افتحه، وادخل على **Extensions**، وثبّت إضافة **C/C++** من Microsoft.

2 نزّل المترجِم

ويندوز: اتبع الدليل الرسمي من VS Code بعنوان "Using GCC with MinGW"، وهو يثبّت لك g++.

ماك: افتح Terminal واكتب `xcode-select --install`

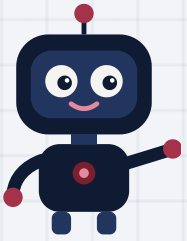
لينكس: اكتب `sudo apt install g++`

3 تأكد أنه يعمل

افتح Terminal جديداً في VS Code (**Terminal ← New Terminal**) واكتب:

```
الطرفية (Terminal)
$ g++ --version
g++ (GCC) 13.3.0
```

ظهر رقم إصدار؟ أنت جاهز. (رقمك قد يكون مختلفاً، وهذا طبيعي.)



نصيحة بِت

على ويندوز: بعد التثبيت **أغلق VS Code وافتحه من جديد**. ظهرت رسالة `g++ is not recognized`؟ المترجِم لم يُضَف إلى PATH بعد، والدليل يشرح الحل. لا يوجد لابتوب؟ ابحث عن "online C++ compiler" واستخدم هاتفك.

خمسة أسطر. وكل سطر مشروح.

أنشئ ملفاً اسمه `hello.cpp` واكتب هذا بالضبط:

HELLO.CPP

```cpp
#include <iostream>

int main() {
    std::cout << "Hello, world!" << "\n";
    return 0;
}
```

سطراً بسطر

أحضِر أدوات **الإدخال والإخراج**. بدونها لا يوجد `std::cout`.	`#include <iostream>`
كل برنامج C++ يبدأ من `main`. و`int` معناها أنها تُرجِع رقماً صحيحاً. و`{` تفتح جسمها.	`int main() {`
`std::cout` = الشاشة. و`<<` = "أرسل هذا إليها". و`std::` معناها "من المكتبة القياسية".	`std::cout << "Hello, world!"`
`\n` = سطر جديد. و`;` تُنهي الأمر، كما تنهي النقطة الجملة.	`<< "\n";`
إرجاع 0 = "انتهيت بدون مشاكل". و`}` تغلق `main`.	`return 0 ;` `}`

الطرفية (Terminal)

```
$ g++ hello.cpp -o hello
$ ./hello
Hello, world!
```

`-o hello` تحدد اسم ملف البرنامج. على ويندوز شغّل `hello.exe` أو `.\hello`.



ركن الأخطاء

نسيت `;`؟ المترجِم يتوقف ويخبرك بالسطر:

```
error: expected ';' before 'return'
```

الحل: أضف `;` في آخر السطر الذي قبله، ثم ترجِم مرة أخرى.

الفصل 01

01 المتغيرات والأنواع

المتغيّر **صندوق له اسم** يحمل قيمة. وفي C++ لكل صندوق أيضاً **نوع**: ما نوع القيمة التي يحملها. ويجب أن تذكر النوع عندما تصنع الصندوق.

ماذا يفعل؟

يخزّن القيم ليتذكرها برنامجك ويستخدمها.

لماذا تحتاجه؟

C++ تستخدم النوع لتعرف **كم ذاكرة** تحجز، وما المسموح أن تفعله بالقيمة.

TYPES.CPP

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {
    int age = 24;
    double price = 9.99;
    char grade = 'A';
    bool isMember = true;
    string name = "Sara";

    cout << name << " is " << age << ", grade " << grade << "\n";
    cout << "Price: " << price << ", member: " << isMember << "\n";
    return 0;
}
```

النتيجة

```
Sara is 24, grade A
Price: 9.99, member: 1
```

النوع	يحمل	مثال	ملاحظة
int	أرقام صحيحة	`24` ، `-5`	بدون كسور
double	أرقام عشرية	`9.99`	للفلوس والحسابات
char	حرف واحد	`'A'`	علامات تنصيص مفردة `' '`
bool	صح / خطأ	`true`	يُطبع 1 أو 0
string	نص	`"Sara"`	علامات مزدوجة، ويحتاج `<string>`