

C++

مفهوم الكونستركتور

أهمية الكونستركتور في C++ <

طريقة استدعاء الكونستركتور في C++ <

مثال شامل حول التعامل مع الكونستركتور في C++ <

أهمية الكونستركتور في C++

من أهم الأشياء التي عليك التفكير بها عند إنشاء كلاس جديد، هي تسهيل طريقة إنشاء كائنات من هذا الكلاس لاحقاً.

من هنا جاءت فكرة الكونستركتور (**Constructor**) و الذي هو عبارة عن دالة، يتم إستدعائها أثناء إنشاء كائن من الكلاس لإعطاء قيم أولية للخصائص الموجودة فيه.

أيّ كلاس تقوم بتعريفه، يملك كونستركتور افتراضي حتى لو تقم بتعريفه له بنفسك و سبب ذلك أنه لا يمكن إنشاء كائن من كلاس إلا من خلال كونستركتور هذا الكلاس.

لهذا إن لم تقم بتعريف كونستركتور للكلاس بنفسك، فكن على يقين أن مترجم لغة C++ سيقوم بإنشاء كونستركتور افتراضي فارغ بدلاً عنك عندما تحاول إنشاء كائن من الكلاس.

نقاط مهمة حول الكونستركتور

- كل كلاس يتم إنشاؤه يحتوي على كونسرتكتور واحد على الأقل. و حتى إن لم تقم بتعريف أي كونسرتكتور، سيقوم المترجم بإنشاء واحد افتراضي.
- في كل مرة يتم إنشاء كائن جديد من كلاس، يجب استدعاء كونسرتكتور من الكلاس حتى يتم إنشاء هذا الكائن.
- القاعدة الأساسية عند تعريف كونسرتكتور هي أنه يجب أن يحمل نفس إسم الكلاس و يكون نوعه **public**.
- في حال قمت بتعريف كونسرتكتور، لن يقوم المترجم بإنشاء واحد افتراضي، أي لن يعود هناك كونسرتكتور افتراضي.
- يمكنك تعريف أكثر من كونسرتكتور. و يمكنك دائماً إنشاء كونسرتكتور فارغ، حتى تستخدمه إن كنت لا تريد إعطاء قيم أولية محددة للخصائص عند إنشاء كائن.

في المثال التالي قمنا بتعريف كلاس إسمه **Book** و لم نقم بتعريف كونسرتكتور له.

المثال الأول

```
class Book {
}
```



- سيقوم المترجم بإنشاء كونسرتكتور فارغ بشكل تلقائي عنا كالتالي.

```
class Book {
public:
    // هذا كونسرتكتور افتراضي أنشأه المترجم لأن الكلاس لا يحتوي على أي كونسرتكتور
    Book() {
    }
}
```



طريقة استدعاء الكونستركتور في C++

عند استدعاء كونستركتور موجود في كلاس, يجب أن تمرر قيمة لكل باراميتر موجود فيه من خلال وضع قوسين بعد إسم الكائن و تمرير القيم فيه.

فيما يلي وضعنا بعض الأمثلة التي توضح كيفية إستدعاء كونستركتور من الكلاس.

إذا كان إسم الكونستركتور هكذا `Book()` فإنك لا تحتاج فعل أي شيء حين تنشئ كائن من الكلاس حيث تنشئه كالتالي.

المثال الأول

```
Book book;
```



إذا كان إسم الكونستركتور هكذا `Book(string title)` فإنك مجبر على تمرير قيمة لكونستركتور الكائن كالتالي عند إنشائه كالتالي.

المثال الثاني

```
Book book("C++ for beginners");
```



إذا كان إسم الكونستركتور هكذا `Book(string title, string author)` فإنك مجبر على تمرير قيم لكونستركتور الكائن كالتالي عند إنشائه كالتالي.

المثال الثالث

```
Book book("C++ for beginners", "Walid Ben" );
```



إذا كان إسم الكلاس يحتوي على كونستركتور فارغ `Book()` و كونستركتور يأخذ قيمتين هكذا `Book(string title, string author)` فهنا تكون مخيّر على استدعاء الكونستركتور الذي تريده منهما عند إنشاء كائن من الكلاس.

المثال الرابع

```
// يمكنك أن تنشئ الكائن من خلال استدعاء الكونستركتور الأول كالتالي
```

```
Book book();
```

```
// يمكنك أن تنشئ الكائن من خلال استدعاء الكونستركتور الثاني كالتالي
```

```
Book book("C++ for beginners", "Walid Ben" );
```




مثال شامل حول التعامل مع الكونستركتور في C++

في المثال التالي, قمنا بتعريف كلاس إسمه `Book` يمثل المعلومات التي يمكن أن يتضمنها أي كتاب كعنوانه, إسم المؤلف, سعره و عدد صفحاته.

في الكلاس `Book` قمنا بتعريف كونستركتور يمكننا من خلاله إدخال قيم مباشرة للخصائص الموجودة في أي كائن ننشئه من الكلاس.

كما أننا قمنا بتعريف دالة إسمها `printInfo()` عند استدعاءها تقوم بطباعة قيم الخصائص التي يمكنها أي كائن ننشئه من الكلاس.

مثال

 main.cpp



```
#include <iostream>
```

```
using namespace std;
```

```
// هنا قمنا بتعريف كلاس اسمه Book يحتوي على 4 متغيرات
```

```
class Book {
```

```
public:
```

```
    // هنا قمنا بتعريف خصائص الكلاس
```

```
    string title;
```

```
    string author;
```

```
    double price;
```

```
    int numberOfPages;
```

```
// هنا قمنا بتعريف كونسرتكتور يجب تمرير 4 قيم له عندما يتم استدعاؤه
```

```
Book(string title, string author, double price, int numberOfPages)
```

```
{
```

```
    // هنا قمنا بوضع القيم التي نمررها للكونسرتكتور بالترتيب في خصائص الكائن
```

```
    this->title = title;
```

```
    this->author = author;
```

```
    this->price = price;
```

```
    this->numberOfPages = numberOfPages;
```

```
}
```

```
// ف دالة إسمها printInfo عند استدعائها من أي كائن ننشئه من Book تقوم بطباعة كل قيم المتغيرات الموجودة فيه
```

```
void printInfo()
```

```
{
```

```
    cout << "Title: " << title << "\n";
```

```
    cout << "Author: " << author << "\n";
```

```
    cout << "Price: " << price << "$\n";
```

```
    cout << "Number of pages: " << numberOfPages << "\n";
```

```
}
```

```
};
```

```
// هنا قمنا بتعريف الدالة main()
```

```
int main()
```

```
{
```

قمنا بتعريف كائن من Book إسمه book مع تمرير القيم التي نريد وضعها فيه بشكل مباشر من خلال الكونستركتور //

```
Book book("C++ for beginners", "Walid Ben", 9.99, 420);
```

هنا قمنا باستدعاء الدالة printInfo() من الكائن book حتى تقوم بطباعة القيم الموجودة فيه //

```
book.printInfo();
```

```
return 0;
```

```
}
```

• سنحصل على النتيجة التالية عند التشغيل.

Title: C++ for beginners

Author: Walid Ben

Price: 9.99\$

Number of pages: 420

