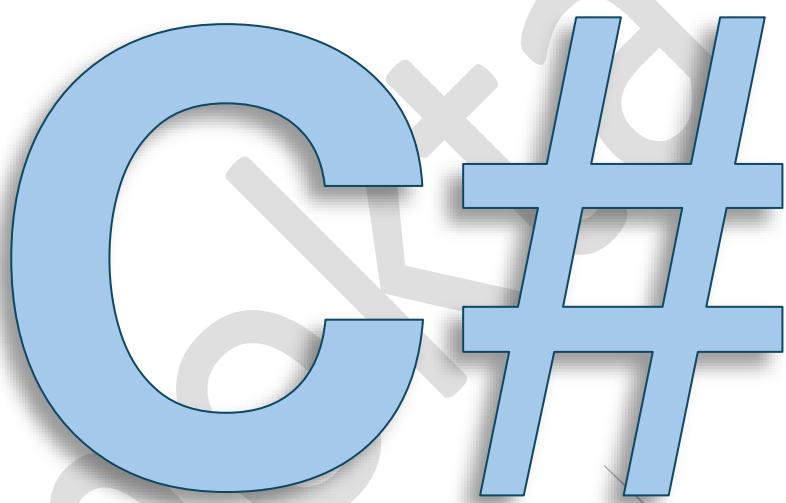


2025



اعداد و تنسيق  
صفاء حسن

## الفهرس

2 .....	مقدمة
3 .....	ما هي لغة C# ؟
3 .....	أين تستخدم لغة سي شارب؟
3 .....	لماذا تستخدم لغة سي شارب؟
3 .....	بيئة تطوير (C# IDE) كيف نبدأ ب برنامج على الكمبيوتر
6 .....	الملاحظات (العروض او comments) بـ C#
7 .....	المتغيرات بـ C#
8 .....	الرموز الخاصة المستخدمة في الطباعة
9 .....	العرض (فلن او output) بـ C#
9 .....	الادخال (كلى او input) بـ C#
9 .....	إدخال أعداد صحيحة (int)
10 .....	إدخال أعداد عشرية (double)
10 .....	الثوابت (קבועات او Constants) بـ C#
13 .....	تمارين

## مقدمة

عزيزي الطالب، في عصرنا الحديث، أصبح علم الحاسوب والبرمجة ركيزة أساسية لكل تطور تقني وابتكار علمي. إذا كنت تسعى لاستثمار قدراتك وإطلاق إبداعك في عالم التكنولوجيا المتقدم، فإن هذا الفرع يمثل نقطة الانطلاق نحو مستقبل مشرق مليء بالفرص والإمكانات.

إن علم الحاسوب ليس مجرد مجال دراسي، بل هو بوابة لاستكشاف عالم من الاحتمالات اللامحدودة. فهو يعلمك كيف تفكّر بأسلوب منطقي، وكيف تحلل البيانات، وكيف تصمم حلولاً مبتكرة للمشكلات. البرمجة ليست مجرد كتابة أكواد، بل هي أداة لخلق أنظمة تسهل حياة الناس، وتساهم في تطوير المجتمعات.

الانخراط في هذا المجال يمنحك فرصة للتميز في سوق العمل المتغير باستمرار، حيث تصبح لديك المهارات التي يبحث عنها العالم الرقمي اليوم. كن على استعداد لمواجهة التحديات، وصناعة التغيير، والارتقاء بقدراتك إلى أعلى المستويات. المستقبل ينتظرك، وعلم الحاسوب هو مفتاحك لتحقيق طموحاتك!

كما أن البرمجة توفر لك فرصة لفهم كيف تعمل التكنولوجيا الحديثة من حولك، بدءاً من التطبيقات اليومية على هاتفك الذكي إلى الأنظمة الكبيرة التي تدير الشركات والمصانع. فهي تمكّنك من تحويل الأفكار إلى مشاريع ملموسة وبرامج مبتكرة تخدم الإنسان والمجتمع.

في مجال البرمجة، ستتعلم لغات وأدوات متعددة مثل C#، التي تُعد واحدة من أبرز اللغات البرمجية المستخدمة في العالم لتطوير التطبيقات المتقدمة، بما في ذلك تطبيقات سطح المكتب، وتطبيقات الويب، وحتى ألعاب الفيديو.

ستكتشف أيضاً أهمية التفكير التحليلي والإبداعي لحل المشكلات، وكيف يمكنك بناء خوارزميات ذكية تعمل بكفاءة ودقة. علم الحاسوب ليس مجرد تعلم كيفية كتابة الأكواد، بل هو طريقة للتفكير والتخطيط، وتطوير الذات لمواكبة التطورات السريعة في عالم التقنية.

انضم إلى عالم البرمجة والحوسبة، وانطلق في رحلتك لبناء مستقبل مليء بالإبداع والتحدي. تذَكّر أن كل خطوة تخطوها في هذا المجال هي استثمار في نفسك وفي مستقبلك، حيث لا حدود لما يمكنك تحقيقه. فابداً الآن، واستعد لتحقيق أحلامك في عالم التقنية!

## ما هي لغة C#؟

C# تُنطق "سي شارب" هي لغة برمجة كائنية التوجه تم إنشاؤها بواسطة شركة مايكروسوف特 وتعمل على إطار عمل .NET.

C# مستوحة من عائلة لغة C، وهي قريبة من لغات شائعة أخرى مثل Java و C++.

تم إصدار النسخة الأولى من لغة سي شارب في عام 2002، وأحدث إصدار منها 12 C#، تم إصداره في نوفمبر 2023.

### لماذا تستخدم لغة سي شارب؟

- تطبيقات الهواتف المحمولة
- تطبيقات سطح المكتب
- تطبيقات الويب
- خدمات الويب
- موقع الإنترن特
- الألعاب
- الواقع الافتراضي (VR)
- تطبيقات قواعد البيانات
- والعديد من الاستخدامات الأخرى!

### لماذا تستخدم لغة سي شارب؟

- تُعد واحدة من أكثر لغات البرمجة شيوعاً في العالم.
- سهلة التعلم وبساطة الاستخدام، مما يجعلها مناسبة للمبتدئين والمحترفين.
- تتمتع بدعم مجتمعي واسع النطاق يوفر الموارد والحلول للمطورين.
- لغة كائنية التوجه، مما يتيح تنظيماً واضحاً للبرامج وإعادة استخدام الكود، وبالتالي تقليل تكلفة التطوير.
- قربها من لغات مثل C و C++ و Java يجعل من السهل على المبرمجين الانتقال بينها وبين C# بسهولة.

\* ليس من الضروري أن تكون لديك خبرة سابقة في البرمجة، لكنك تبدأ هذا الدليل التعليمي سيعمل لك أساسيات لغة سي شارب. لنبدأ

### بيئة تطوير (C# IDE) كيف نبدأ ببرنامج على الكمبيوتر

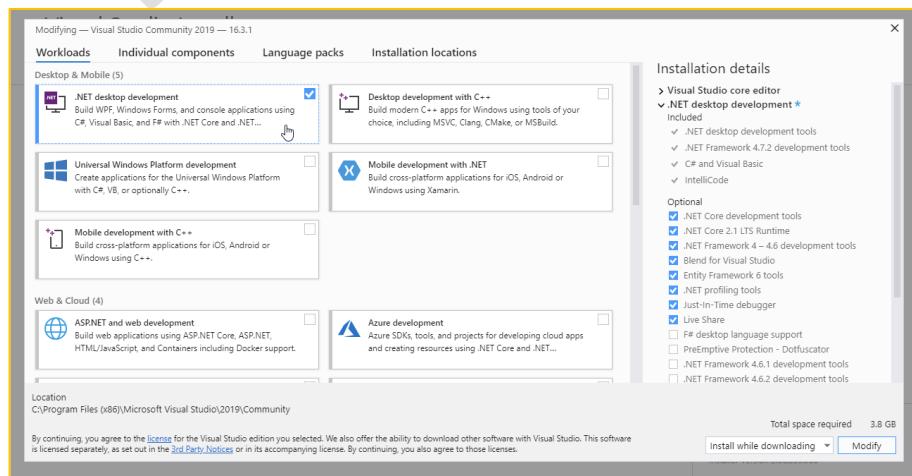
أسهل طريقة للبدء بتعلم لغة C# هي استخدام برنامج خاص يسمى بيئة تطوير متكاملة (IDE).  
هذا البرنامج يساعدك في كتابة الكود وتصحيحه وتشغيله بسهولة.

أفضل برنامج يمكن استخدامه هو Visual Studio Community.

هذا البرنامج مجاني تماماً ويمكن تحميله من الرابط التالي: <https://visualstudio.microsoft.com/vs/community/>

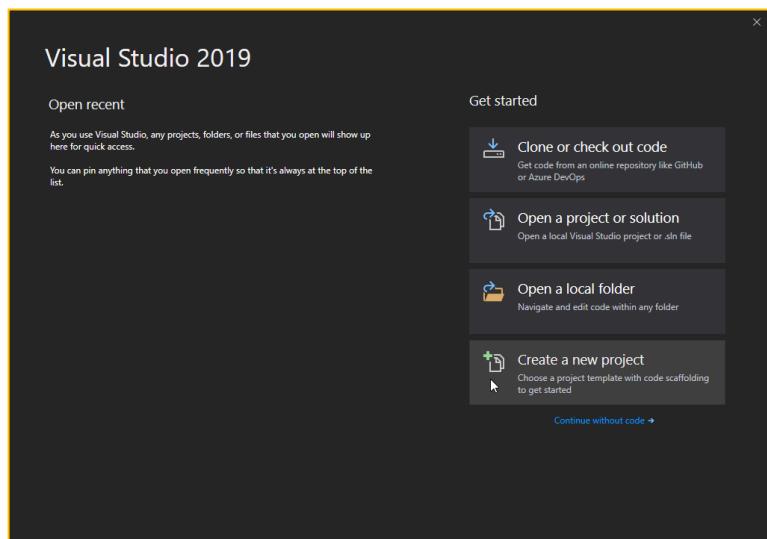
بعد التثبيت، افتح البرنامج واختر خيار .NET workload (الأدوات الخاصة بإطار عمل .NET).

اضغط على زر Install أو Modify لبدء عملية التثبيت:

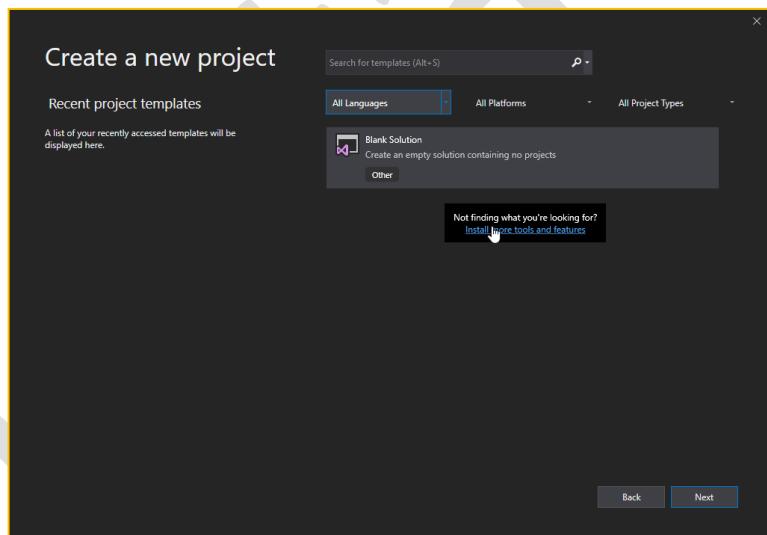


بعد اكتمال عملية التثبيت، اضغط على زر Launch لبدء تشغيل برنامج Visual Studio.

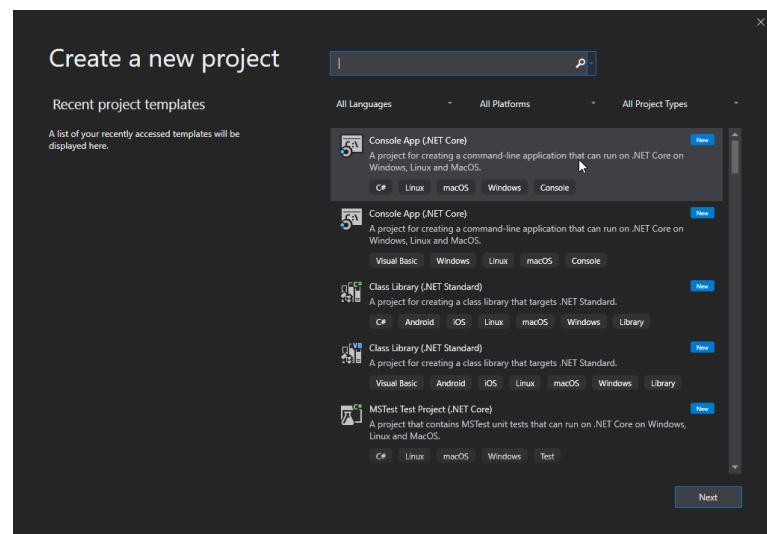
في نافذة البدء، اختر **Create a new project** (إنشاء مشروع جديد):



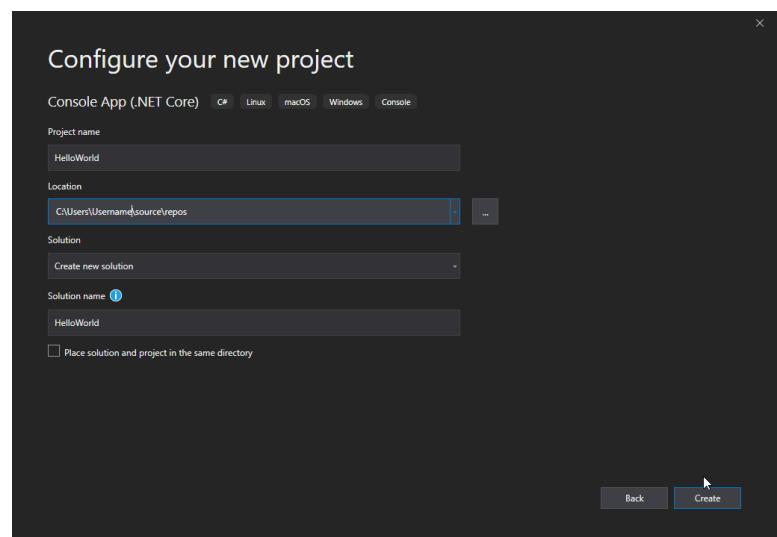
ثم اضغط على الرابط "Install more tools and features" (تثبيت المزيد من الأدوات والميزات):



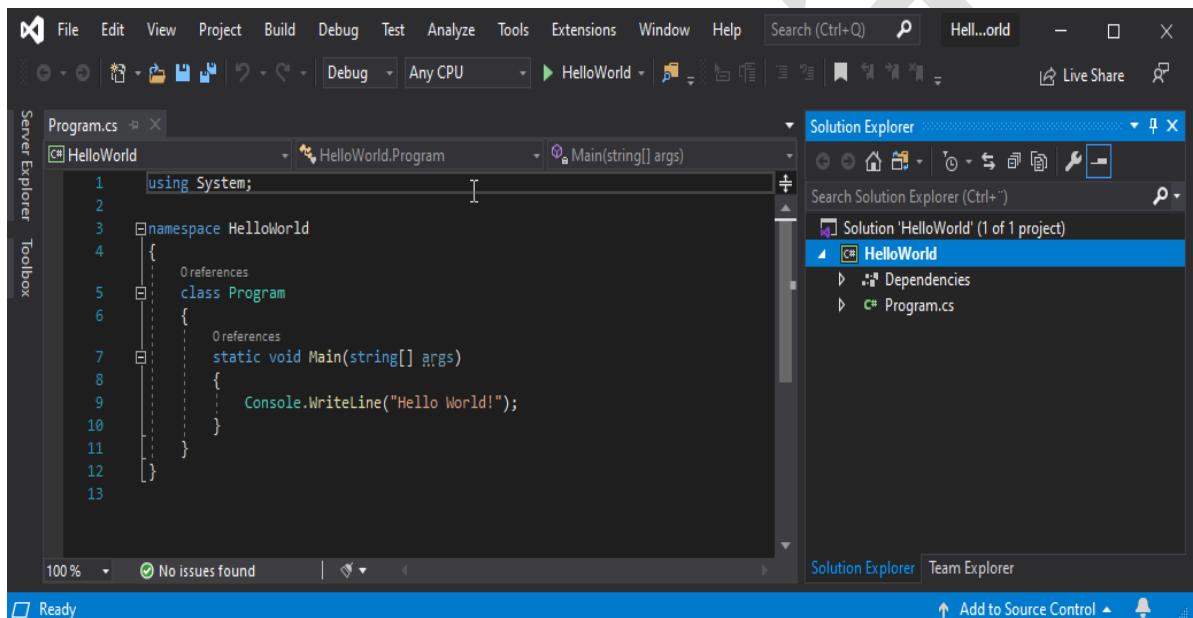
اختر "Console App (.NET Core)" (تطبيق وحدة التحكم باستخدام .NET Core) من القائمة، ثم اضغط على زر **Next** (التالي):



أدخل اسمًا لمشروعك، ثم اضغط على زر Create (إنشاء):



سيقوم Visual Studio تلقائيًا بإنشاء بعض الكود الأساسي لمشروعك:



لا تقلق إذا لم تفهم الكود المولد أعلاه – سنناقشه بالتفصيل في الفصول القادمة.

في الوقت الحالي، ركز فقط على كيفية تشغيل الكود.

لتشغيل البرنامج: اضغط على الزر F5 على لوحة المفاتيح (أو اختر "Start Debugging" من القائمة).

سيقوم هذا بـ**تجميع الكود وتنفيذها**. النتيجة ستبدو كالتالي:

```
Hello World!
C:\Users\Username\source\repos\HelloWorld\HelloWorld\bin\Debug\netcoreapp3.0\HelloWorld.exe
To automatically close the console when debugging stops, enable Tools->Options->Debugging->
Press any key to close this window . . .
```

مبروك! لقد قمت بكتابة وتنفيذ أول برنامج بلغة C# بنجاح. 😊

## نفس قليلاً ماذا فعلنا في الفصل السابق

قمنا بإنشاء ملف بلغة C# يسمى Program.cs، واستخدمنا الكود التالي لطباعة "Hello World" على الشاشة:

```
using System;
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

"using System;"  
يتيح استخدام الأدوات الجاهزة من مكتبة System (مثل الطباعة على الشاشة).

"namespace HelloWorld"  
يستخدم لتنظيم الكود وتجميع الأجزاء المرتبطة معاً.

static void Main(string[] args)"  
هذا هي نقطة البداية للبرنامج. أي كود بداخلاها سيتم تشغيله عند بدء البرنامج.

" Console.WriteLine("Hello World!"); "  
أمر لطباعة النص "Hello World" على الشاشة.

"class Program" هي  
حاوية تضم الكود الخاص بالبرنامج. هنا اسم الفئة هو Program.

الأقواس المعقونة ({}): تحدد بداية ونهاية الكود.

علامات الاقتباس المزدوجة (""): تحدد النصوص التي سيتم طباعتها.

الفاصلة المنقوطة (;): يجب أن تكون في نهاية كل أمر برمجي.

أي كود مكتوب داخل الأقواس المعقونة {} الخاصة بـ Main سيتم تنفيذه. لداعي لفهم الكلمات المفتاحية قبل وبعد Main بشكل عميق، ستتعرف عليها تدريجياً. كل ما عليك الان معرفته ان الكود الذي ستكتبه يجب ان يكون بين الأقواس المعقونة {}. الخاصية بـ Main. لكن يجب أن تكتب !!

## الملاحظات (الهערות او comments) بـ C# (comments)

الملاحظات تُستخدم لك اكتاب اكواط لتوضيح الكود المكتوب بلغة C# ولجعل الكود أكثر قابلية للقراءة. كما يمكن استخدامها لتعطيل سطور من الكود أثناء تجربة بدائل.

أي أن أي شيء يكتب في الكود بعد // / حتى نهاية السطر يتم تجاهله من قبل المترجم (لن يتم تنفيذه).

ملاحظة //  
هذا الامر يطبع مرحبا بكم //  
Console.WriteLine("مرحبا بكم");

يجب ان تنتهي الملاحظة في نفس السطر الذي بدأت به // .

لكتابة ملاحظة في اكثرا من سطر واحد، نضع الملاحظة بين /\* و \*/، هكذا:

/\*
ال코드 أدناه سيطبع الكلمات (مرحبا بكم) على الشاشة
وهو رائع جدا
\*/
Console.WriteLine("مرحبا بكم");

## C# المتغيرات بـ

المتغيرات هي أماكن مخصصة في ذاكرة الحاسوب تُستخدم لتخزين القيم والبيانات أثناء تشغيل البرنامج.

يتم تعريف المتغير بناءً على نوع البيانات (Type) المراد تخزينها فيه. لكل متغير يتم تعريفه، يجب تحديد نوع بياناته (مثل رقم صحيح، نص، إلخ...).

كل متغير يجب أن يتم إعطاؤه اسمًا مميزًا (Name) بحيث يمكن استدعاؤه لاحقًا في الكود. بعد ذلك يمكن إسناد قيمة إلى المتغير باستخدام علامة (=)، كما يمكن تعديل القيمة لاحقًا أثناء تنفيذ البرنامج.

أنواع المتغيرات:

- .1 - يستخدم لتخزين الأعداد الصحيحة (بدون كسور)، مثل 123 أو 123.
- .2 - يستخدم لتخزين الأعداد الحقيقية (الأعداد التي تشمل الأعداد الصحيحة مثل 1 و 3 والأعداد العشرية مثل 0.75).
- .3 - يستخدم لتخزين حرف واحد فقط، مثل 'a' أو 'B'.
- .4 - يستخدم لتخزين النصوص، مثل "Hello World".
- .5 - يستخدم لتخزين القيم المنطقية التي تكون إما true (صحيح) أو false (خطأ).

أمثلة:

```
تعريف متغير عددي بدون قيمة ابتدائية //  
int a;  
قيمة ابتدائية 5 بدون قيمة، و n1 //  
int n1, n2 = 5; //  
تعريف متغير يحمل رمزاً //  
char ch = '*' ;  
تعريف متغير عشري بقيمة 1.4 //  
double x = 1.4;  
تعريف متغير نصي بدون قيمة ابتدائية //  
string str;
```

## الرموز الخاصة المستخدمة في الطباعة

الرمز	الوصف	مثال
\n	إضافة سطر جديد	<pre>Console.WriteLine("hello\nworld"); hello world</pre> <p>C:\Users\safaa\OneDrive\Desktop\HelloWorld\cd path_to_your_file_14024) exited with code 0 (0x0). To automatically close the console when debugging stops, press the F5 key when debugging stops. Press any key to close this window . . . </p>
t\	إضافة مسافة (Tab)	<pre>Console.WriteLine("hello\tworld"); hello    world</pre> <p>C:\Users\safaa\OneDrive\Desktop\HelloWorld\cd path_to_your_file_26160) exited with code 0 (0x0). To automatically close the console when debugging stops, press the F5 key when debugging stops. Press any key to close this window . . . </p>
"\	طباعة علامة الاقتباس"	<pre>Console.WriteLine("""\hello world"""); "hello world"</pre> <p>C:\Users\safaa\OneDrive\Desktop\HelloWorld\cd path_to_your_file_6728) exited with code 0 (0x0). To automatically close the console when debugging stops, press the F5 key when debugging stops. Press any key to close this window . . . </p>
\\	طباعة شرطة مائلة \ للخلف	<pre>Console.WriteLine("hello\\world"); hello\world</pre> <p>C:\Users\safaa\OneDrive\Desktop\HelloWorld\cd path_to_your_file_20176) exited with code 0 (0x0). To automatically close the console when debugging stops, press the F5 key when debugging stops. Press any key to close this window . . . </p>

## العرض (فلات او output) بـ C#

العرض في لغة C# يشير إلى عملية إظهار النتائج أو البيانات للمستخدم.

يتم ذلك غالباً باستخدام نافذة الكونسول (Console)، التي تعتبر وسيلة أساسية لإخراج البيانات النصية أثناء تشغيل البرنامج.

بـ C# اذا كنت تريد طباعة قيمة او نصوص على الشاشة نستخدم الدالة WriteLine()

يكتب الامر هكذا:

```
Console.WriteLine("Hello World!");
```

المثال 1

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("مرحباً ");
        Console.Write(" بك ");
        Console.Write(" في C#! ");
    }
}
```

أي شيء نريد طباعته نضعه بين علامات الاقتباس المزدوجة " ".

المثال 2

```
static void Main()
{
    int age = 25;
    string name = "أحمد";
    Console.WriteLine("اسمي : " + name);
    Console.WriteLine("عمرى : " + age);
}
```

## الادخال (فلات او input) بـ C#

الإدخال في لغة C# يشير إلى عملية تلقي البيانات من المستخدم أثناء تشغيل البرنامج.

يتم تنفيذ ذلك باستخدام الأمر Console.ReadLine()

إدخال أعداد صحيحة (int)

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("أدخل عمرك: ");
        int age = int.Parse(Console.ReadLine()); // تحويل النص المدخل إلى عدد صحيح
        Console.WriteLine("عمرك هو: " + age);
    }
}
```

## إدخال أعداد عشرية (double)

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("أدخل رقمًا عشريًّا: ");
        double decimalNumber = double.Parse(Console.ReadLine());
        Console.WriteLine($"الرقم العشري المدخل هو: {decimalNumber}");
    }
}
```

## الثوابت (קבועים أو Constants) في C#

الثوابت هي متغيرات لا يمكن تغيير قيمتها بعد تعريفها. تُستخدم لتخزين القيم التي لا تتغير أثناء تشغيل البرنامج. يتم استخدام الكلمة المفتاحية `const` عند تعريف الثابت. يجب تعين قيمة للثابت عند تعريفه، ولا يمكن تغيير هذه القيمة لاحقًا.

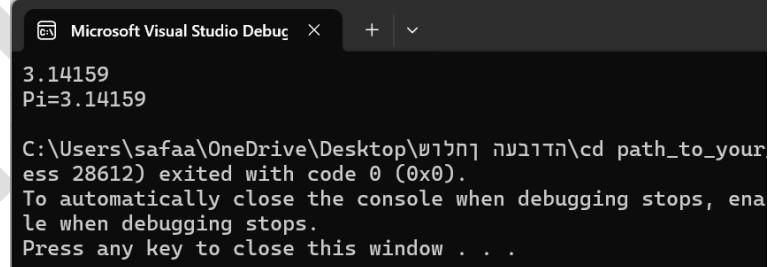
### مثال 1

```
تعريف ثابت // const double Pi = 3.14159;
تعريف ثابت // const int MaxUsers = 100;
Pi = 4;
Console.WriteLine("هي: " + Pi);
Console.WriteLine("العدد الأقصى للمستخدمين هو: " + MaxUsers);
```

في المثال أعلاه يوجد مشكلة وهو انه تم تغيير قيمة `Pi` بعد ان عُرفت على أنها متغير ثابت.

### مثال 2

```
تعريف ثابت // const double Pi = 3.14159;
Console.WriteLine(Pi);
Console.WriteLine("Pi={0}", Pi);
```



### مثال 3، تعريف واستخدام ثابت عددى

```
using System;

class Program
{
    static void Main()
    {
        const int MaxAge = 100; // تعريف ثابت عددى
        Console.WriteLine($"سنَة {MaxAge}: العمر الأقصى المتوقع هو");
    }
}
```

## C# (Operators in C#)

في لغة C#, المعاملات (Operators) هي رموز خاصة تُستخدم لإجراء عمليات على القيم والمتغيرات.

وتنقسم المعاملات إلى عدة أنواع رئيسية:

### (Arithmetic Operators)

تُستخدم لإجراء العمليات الحسابية على الأعداد.

مثال	الوصف	المعامل
int result = 5 + 3 ;	الجمع	+
int result = 5 - 3 ;	الطرح	-
int result = 5 * 3;	الضرب	*
int result = 10 / 2 ;	القسمة	/
int result = 10 % 3 ;	باقي القسمة	%

ملاحظة: في القسمة / بين عددين صحيحين int، يتم تقرير الناتج إلى أقرب عدد صحيح.

### (Comparison Operators)

تُستخدم لمقارنة قيمتين وتعيد إما true أو false.

مثال	الوصف	المعامل
5 == 5	يساوي	==
3 != 5	لا يساوي	!=
3 < 5	أكبر من	<
3 > 5	أصغر من	>
5 <= 5	أكبر من أو يساوي	<=

### (Logical Operators)

تُستخدم للعمليات المنطقية مثل AND، OR و NOT.

غالباً تُستخدم في العبارات الشرطية.

النتيجة	مثال	الوصف	المعامل
true	(5 > 3) && (10 > 5)	و AND - تعيد true إذا كانت كلتا العبارتين صحيحتين	&&
true	(5 > 3)    (10 < 5)	أو OR - تعيد true إذا كانت أي من العبارتين صحيحة	
false	!(5 > 3)	ليس (NOT) - تعكس النتيجة	!

### (Assignment Operators)

تُستخدم لإسناد القيم للمتغيرات.

مثال مكافئ	الوصف	المعامل
x = 5;	إسناد القيمة	=
x += 3; ⇔ x = x + 3;	إضافة ثم إسناد	+=
x -= 3; ⇔ x = x - 3;	طرح ثم إسناد	-=
x *= 3; ⇔ x = x * 3;	ضرب ثم إسناد	*=
x /= 3; ⇔ x = x / 3;	قسمة ثم إسناد	/=
x %= 3; ⇔ x = x % 3;	باقي القسمة ثم إسناد	%=

### (Unary Operators)

تُستخدم مع قيمة واحدة فقط.

المثال	الوصف	المعامل
+5	تحويل القيمة إلى موجبة	+
-5	تحويل القيمة إلى سالبة	-

<code>++x</code> أو <code>x++</code>	زيادة القيمة بمقدار 1 (قبل أو بعد المتغير)	<code>++</code>
<code>--x</code> أو <code>x--</code>	إنقصاص القيمة بمقدار 1 (قبل أو بعد المتغير)	<code>--</code>

ملاحظة:

- عند استخدام `x++` يتم الإرجاع أولاً ثم الزيادة.
- عند استخدام `++x` يتم الزيادة أولاً ثم الإرجاع.

الهيئات الشرطية (if-else, switch-case)

الحلقات التكرارية (for, while, do-while)

المصفوفات (Arrays) في C#

الدوال (Functions) في C#

## تمارين

1. اكتب برنامج يستقبل عدد صحيح على البرنامج ان يطبع العدد.
2. اكتب برنامج يستقبل عددين صحيحين ويطبعهما.
3. اكتب برنامج يستقبل 3 اعداد صحيحة ويطبع العدد الثاني.
4. اكتب برنامج يستقبل 3 اعداد صحيحة ويطبع سطرين في السطر الأول يطبع العدد الثاني والثالث والسطر الثاني يطبع العدد الثالث والأول.
5. اكتب برنامج يستقبل 3 اعداد صحيحة ويطبع 6 اعداد، الأول يطبعه مرة واحدة والعدد الثاني مرتين والثالث ثلاث مرات.
6. اكتب برنامج يستقبل عددين صحيحين ويطبع سطرين السطر الأول يطبع الأول والثاني السطر الثاني يطبع الأول ثلاث مرات.
7. اكتب برنامج يستقبل 3 اعداد صحيحة ويطبع العدد الأول خمس مرات والثاني والثالث في سطر واحد.
8. اكتب برنامج يستقبل عددين صحيحين يمثلان طول وعرض المستطيل على البرنامج ان يطبع مساحة المستطيل ومحيطه.
9. اكتب برنامج يستقبل طول ضلع مكعب على البرنامج ان يطبع حجم المكعب ومساحة وجوه المكعب.

\*ملاحظة؛ من المفضل استعمال متغيرات ذات معنى تدل على وظيفة المتغير في البرنامج.

10. معطى قطعة برنامج، تتبع البرنامج عن طريق جدول متابعة

```
//تعريف المتغيرات والقيم الابتدائية
int num1 = 14, num2 = 7, sum, sub, mult, div;

//العمليات الحسابية الأساسية
sum = num1 + num2;
sub = num1 - num2;
mult = num1 * num2;
div = num1 / num2;
```

11. معطى قطعة برنامج، تتبع البرنامج عن طريق جدول متابعة

```
//تعريف المتغيرات
int num1, num2, num3, diff1, diff2;

//طلب إدخال الأرقام الثلاثة من المستخدم
Console.WriteLine("Enter three numbers:");
num1 = int.Parse(Console.ReadLine());
num2 = int.Parse(Console.ReadLine());
num3 = int.Parse(Console.ReadLine());

//العمليات الحسابية
diff1 = num1 * num2 - num3;
diff2 = num1 * num3 - num1;
```

12. اكتب برنامج يستقبل عدد صحيح ويطبع العدد الثاني مساوٍ لضعفه والعدد الثالث مساوٍ لثلاث اضعاف العدد الثاني.
13. اكتب برنامج يستقبل أربع اعداد صحيحة على البرنامج ان يطبع معدلهم.
14. اكتب برنامج يستقبل عددين غير صحيحين ويطبع كل عدد بعد قسمته على 3.
15. اكتب برنامج يستقبل ثلاث أسعار لمنتجات الحليب ويطبع سعرهن بعد إضافة الضريبة 18%.
16. اكتب برنامج يستقبل نصف قطر الدائرة ويطبع مساحة الدائرة.
17. الدخول لبركة السباحة للكبار 20 شيكل وللصغار 10 شيكل، اكتب برنامج يستقبل عدد الكبار والصغار الذين دخلوا البركة ويطبع المجموع الكلي للدفع.
18. اكتب برنامج يستقبل سعر الكتاب وعدد الكتب ويطبع السعر الكلي للدفع.
19. اكتب برنامج يستقبل عدد الدولارات ويطبع قيمتها في الشيكل.
20. نفذ الأوامر:
- أ. اضرب  $2 * a$  واعرضها داخل  $A$ .
  - ب. اطرح قيمة  $B$  من  $a$  واعرضها في  $a$ .
  - ت. اضرب قيمة  $A$  بالمجموع  $B + C$  واعرضها في  $A$ .
21. معطى قطعة برنامج، ماذا تطبع قطعة البرنامج؟

```

9   Console.WriteLine("insert number");
10  int a = int.Parse(Console.ReadLine());
11  Console.WriteLine("insert number");
12  int b = int.Parse(Console.ReadLine());
13  a = a + b;
14  Console.WriteLine(a);
15  a = a - b;
16  Console.WriteLine(a);

```

22. معطى قطعة برنامج، ماذا تطبع قطعة البرنامج؟

```

9   int a, b, c, temp;
10  Console.WriteLine("insert three number");
11  a= int.Parse(Console.ReadLine());
12  b= int.Parse(Console.ReadLine());
13  c = int.Parse(Console.ReadLine());
14  temp = a;
15  a = b;
16  b = c;
17  c = temp;
18  Console.WriteLine(a);
19  Console.WriteLine(b);
20  Console.WriteLine(c);

```

23. اكتب برنامج يستقبل طول طالبين ويطبع الأطول بينهما.
24. اكتب برنامج يستقبل طول طالبين ويطبع الفرق بينهما.
25. اكتب برنامج يستقبل طول وعرض المستطيل على البرنامج ان يطبع المساحة ويطبع طول الوتر