📘 الدليل الشامل لتعلم أساسيات لغة #C

(من الصفر إلى الاحتراف – النسخة المحسّنة)

📌 المقدمة

لغة #C (تُنطق "سي شارب") هي لغة برمجة حديثة، قوية، وموجهة للكائنات، طوّرتها شركة مايكروسوفت كجزء من منظومة .NET. تُستخدم على نطاق واسع في تطوير:

- تطبيقات سطح المكتب (Windows Forms, WPF)
- تطبيقات الويب (ASP.NET Core)
- الألعاب (باستخدام محرك Unity)
- الخدمات السحابية والـ APIs

هذا الدليل مصمم ليأخذك من المبتدئ الكامل إلى مستوى متقدم في الأساسيات، استعدادًا للانتقال إلى البرمجة الكائنية (OOP) والمشاريع الواقعية.

---

1. أنواع البيانات العددية الصحيحة (Integers)

تُستخدم لتمثيل الأعداد الصحيحة (بدون فاصلة عشرية).

`sbyte`	-128 إلى 127	1	عدد صحيح موقّع
`byte`	0 إلى 255	1	غير موقّع
`short`	-32,768 إلى 32,767	2	

`ushort`	0 إلى 65,535	2	
`int`	-2,147,483,648 إلى 2,147,483,647	4	الأكثر استخدامًا
`uint`	0 إلى 4,294,967,295	4	
`long`	-9,223,372,036,854,775,808 إلى 9,223,372,036,854,775,807	8	عند التعيين `L` يُضاف
`ulong`	0 إلى 18,446,744,073,709,551,615	8	

أمثلة:

csharp

```csharp
int age = 25;
long bigNumber = 9223372036854775807L; // ملاحظة: L إلزامية
short smallNumber = 32000;
byte smallByte = 255;
Console.WriteLine(age);
Console.WriteLine(bigNumber);
```

💡 نصيحة: استخدم `int` ما لم تكن بحاجة إلى نطاق أوسع.

---

2. الأنواع العددية العشرية (Floating-Point & Decimal)

`float`	~7 أرقام	يُضاف `f` (مثل: `3.14f`)
`double`	~15–16 رقم	الافتراضي للأعداد العشرية
`decimal`	28–29 رقم	الحسابات المالية (مثل الرواتب، الأسعار)

أمثلة:

csharp

```csharp
float price = 19.99f;
double pi = 3.141592653589793;
decimal salary = 5000.50m; // ملاحظة: m إلزامية
Console.WriteLine(price);
Console.WriteLine(pi);
Console.WriteLine(salary);
```

⚠️ تحذير: لا تستخدم `float` أو `double` للحسابات المالية بسبب أخطاء التقريب!

---

3. النصوص (Strings)

- النوع `string` يُستخدم لتمثيل سلسلة من المحارف.
- غير قابل للتغيير (Immutable): كل تعديل يُنشئ كائنًا جديدًا.

أمثلة:

csharp

```csharp
string name = "أحمد";
Console.WriteLine("مرحبًا، " + name);
```

تنسيق النصوص

أ. `string.Format`

csharp

```
string message = string.Format("30 ,"سارة" ,"{1} :العمر ،{0} :الاسم");
```

ب. Interpolation (الأفضل!)

csharp

```
string name = "ليلى";
int age = 22;
Console.WriteLine($"الاسم: {name}، العمر: {age}");
```

ج. Verbatim + Interpolation (لمسارات الملفات)

csharp

```
1
string path = $@"C:\Users\{name}\Documents";
```

---

4. القيم المنطقية (Booleans)

يمكن أن تكون إما `true` أو `false`.

csharp

```
bool isStudent = true;
bool isAdult = age >= 18;
Console.WriteLine(isAdult); // true 18 ≤ إذا كان العمر
```

---

5. التحويل بين الأنواع (Type Conversion)

أ. التحويل الضمني (Implicit)

من نوع أصغر إلى أكبر (آمن):

csharp

```csharp
int a = 100;
long b = a; // ✅ لا حاجة لـ casting
```

ب. التحويل الصريح (Explicit)

من نوع أكبر إلى أصغر (قد يفقد بيانات):

csharp

```csharp
double d = 123.45;
int c = (int)d; // c = 123
```

ج. باستخدام `Parse` و `TryParse`

- `("int.Parse("42`: يُطلق استثناء إذا فشل.
- `(int.TryParse("42", out int num`: آمن، يُرجع `true`/`false`.

csharp

```csharp
string input = "123";
if (int.TryParse(input, out int number))
    Console.WriteLine($"العدد: {number}");
else
    Console.WriteLine("إدخال غير صالح!");
```

✅ أفضل ممارسة: استخدم دائمًا `TryParse` مع إدخال المستخدم.

---

6. المتغيرات والثوابت

`var` – استنتاج النوع تلقائيًا

csharp

```csharp
var name = "علي";      // string
var count = 10;        // int
```

```
var price = 99.99m;    // decimal
```

⚠️ يجب تهيئة القيمة عند الإعلان.

`const` – ثابت لا يتغير

csharp

```
const double PI = 3.14159;
// PI = 3; // ❌ خطأ! لا يمكن تغييره
```

---

7. أنواع القيم مقابل أنواع المرجع (Value vs Reference Types)

`int`,`bool`,`char`,`struct`	`string`,`class`,`array`
تُخزن في Stack	تُخزن في Heap
النسخ = نسخ القيمة	النسخ = نسخ المرجع (نفس الكائن!)

csharp

```
int x = 5;            // Value
string s = "Hello";   // Reference
```

---

8. التواريخ والوقت

`DateTime` – تاريخ + وقت

csharp

```
1-DateTime now = DateTime.Now;
2-Console.WriteLine(now);
```

`DateOnly` و `TimeOnly` (.NET 6+)

csharp

```csharp
DateOnly birthDate = new DateOnly(1990, 5, 20); // تاريخ فقط TimeOnly
meetingTime = new TimeOnly(14, 30);     // وقت فقط
```

حساب العمر

csharp

```csharp
DateTime today = DateTime.Today;

DateTime birthday = new DateTime(2000, 1, 1);

int age = today.Year - birthday.Year;

if (today < birthday.AddYears(age)) age--;

Console.WriteLine($"العمر: {age}");
```

---

9. إدخال البيانات من المستخدم

csharp

```csharp
Console.Write("أدخل اسمك: ");

string name = Console.ReadLine();

Console.WriteLine($"مرحبًا، {name}!");
```

---

10. المعاملات الحسابية والمنطقية

المعاملات الحسابية

csharp

```csharp
int a = 10, b = 3;

Console.WriteLine(a + b); // 13

Console.WriteLine(a % b); // 1 (الباقي)
```

أولوية المعاملات

1. الأقواس `()`
2. الضرب `*`، القسمة `/`، الباقي `%`
3. الجمع `+`، الطرح `-`

csharp

```csharp
int result = 5 + 3 * 2; // = 11
```

معاملات الزيادة والنقصان

csharp

```csharp
int x = 5;
Console.WriteLine(x++); // يطبع 5، ثم x = 6
Console.WriteLine(++x); // يطبع 7، ثم x = 7
```

---

11. عبارات التحكم (Control Flow)

`if` / `else if` / `else`

csharp

```csharp
if (age >= 18)
    Console.WriteLine("أنت بالغ");
else
    Console.WriteLine("أنت قاصر");
```

العامل الثلاثي (Ternary)

csharp

```csharp
string status = (age >= 18) ? "بالغ" : "قاصر";
```

`switch` (كـ statement)

csharp

```csharp
switch (day)
{
    case "Saturday":
```

```
        Console.WriteLine("عطلة");
        break;
    default:
        Console.WriteLine("يوم عمل");
        break;
}
```

`switch` (كـ expression – حديث)

csharp

```
string result = day switch
{
    "Saturday" or "Sunday" => "عطلة",
    _ => "يوم عمل"
};
```

المعاملات المنطقية

csharp

```
if (age >= 18 && hasLicense)
    Console.WriteLine("يمكنك القيادة");
```

---

12. الحلقات التكرارية (Loops)

`for`

csharp

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine($"العدد: {i}");
}
```

حلقة متعددة

csharp

```csharp
for (int i = 0, j = 5; i < 3; i++, j--)
    Console.WriteLine($"{i}, {j}");
```

---

13. التعامل مع النصوص (String Manipulation)

csharp

```csharp
string text = "أحمد";
Console.WriteLine(text.Length);          // 4
Console.WriteLine(text.ToUpper());       // "أحمد"
Console.WriteLine(text.Substring(0, 3)); // "أحم"
Console.WriteLine(text.Replace("إحمد" // ;(("إ" ,"أ"
```

تسلسلات الهروب (Escape Sequences)

csharp

```csharp
Console.WriteLine("الاسم:\tأحمد\nالعمر:\t25");
// الناتج:
// الاسم:    أحمد
// العمر:    25
```

---

14. الأنواع القابلة للقيمة الفارغة (Nullable Types)

csharp

```csharp
int? age = null;
Console.WriteLine(age.HasValue); // false


int realAge = age ?? 0; // استخدم 0 إذا كانت null
```

🧠 تمارين مقترحة