🥊 Boxing vs Unboxing في #C: شرح بسيط ومفيد للمطورين

لو بتتعلم C# أو شغال بيها، أكيد قابلت مصطلحات غريبة شوية زي **Boxing** و **Unboxing**.
الموضوع مش له علاقة بالملاكمة 😄، لكن له تأثير كبير على أداء برنامجك وفهمك لطريقة إدارة الذاكرة في .NET.

📦 ما هو الـ Boxing؟

الـ **Boxing** هو عملية **تحويل Value Type إلى Reference Type**.
يعني لو عندك متغير من نوع بسيط زي `int` أو `bool` وحطيته في متغير من النوع `object`، بيتم "تغليفه" داخل صندوق (Box) في الـ Heap بدل ما يفضل في الـ Stack.

🔹 مثال:

csharp
CopyEdit
```
int num = 5;
object obj = num; // Boxing
```

هنا `num` كان Value Type، لكن لما حطيناه في `obj` اتحول إلى Reference Type.

📤 ما هو الـ Unboxing؟

الـ **Unboxing** هو العكس تمامًا — **تحويل Reference Type مرة تانية إلى Value Type**.
يعني بتفتح الصندوق وتسترجع القيمة الأصلية.

🔹 مثال:

csharp
CopyEdit
```
object obj = 5;
int num = (int)obj; // Unboxing
```

⚡ ملاحظات مهمة:

1. **Boxing** و **Unboxing** **بيكلفوا أداء** لأنهم بيعملوا عمليات إضافية على الذاكرة (Heap ↔ Stack).
2. حاول تتجنبهم في الكود اللي بيشتغل في لوبات كبيرة أو عمليات حساسة للأداء.
3. استخدام الـ **Generics** غالبًا بيقلل الحاجة لعمليات Boxing/Unboxing.

💡 **خلاصة الفكرة**:
تخيل إن الـ Boxing زي ما تكون بتحط حاجة صغيرة في علبة كبيرة عشان تقدر تتعامل معاها بمرونة، لكن فتح العلبة (Unboxing) بياخد وقت ومجهود.
لو فهمت الفكرة دي، هتعرف تتجنب الأخطاء اللي بتبطّأ البرنامج وتكتب كود أنضف وأسرع.