# C# في Class و Struct و Record الفرق بين

## 1. Class (الكلاس)

✔ نوع مرجعي (Reference Type) → يتخزن في الـ Heap

✔ عند تمرير كائن من كلاس يتم تمرير المرجع (العنوان) وليس نسخة جديدة

✔ يدعم الوراثة (Inheritance) وتعدد الأشكال (Polymorphism)

✔ يُستخدم مع الكائنات الكبيرة والمعقدة

مثال:

class Person  
{  
 public string Name { get; set; }  
 public int Age { get; set; }  
}

## 2. Struct (الاستراكت)

✔ نوع قيمي (Value Type) → يتخزن في الـ Stack

✔ عند تمريره يتم نسخ البيانات وليس المرجع

✔ لا يدعم الوراثة (لكن ممكن يطبق Interfaces)

✔ أسرع في الأداء مع البيانات الصغيرة

✔ يُستخدم للبيانات البسيطة مثل (الإحداثيات)

مثال:

struct Point  
{  
 public int X { get; set; }  
 public int Y { get; set; }  
}

## 3. Record (الريكورد)

✔ نوع مرجعي (Reference Type) مثل الكلاس (ويدعم record struct أيضًا)

✔ مخصص للعمل مع البيانات الثابتة (Immutable)

✔ المقارنة تكون بالقيمة (Value-based Equality) افتراضياً

✔ يُستخدم غالباً في الـ DTOs ونقل البيانات

✔ يدعم الوراثة مثل الكلاس

مثال:

record Person(string Name, int Age);  
  
var p1 = new Person("Ali", 30);  
var p2 = new Person("Ali", 30);  
  
Console.WriteLine(p1 == p2); // True (مقارنة بالقيمة)

## جدول مقارنة

|  |  |  |  |
| --- | --- | --- | --- |
| الميزة | Class | Struct | Record |
| نوع التخزين | مرجعي (Heap) | قيمي (Stack) | مرجعي (Heap) |
| طريقة المساواة | مرجعية (Reference) | بالقيمة (Value) | بالقيمة (Value) |
| الوراثة | نعم | لا (إلا انترفيس) | نعم |
| الاستخدام | كائنات كبيرة ومعقدة | بيانات صغيرة | DTOs / Immutable |
| إمكانية التغيير | Mutable | Mutable | Immutable افتراضياً |

## مثال عملي يوضح الفرق

class C { public int X { get; set; } }  
struct S { public int X { get; set; } }  
record R(int X);  
  
var c1 = new C { X = 5 };  
var c2 = new C { X = 5 };  
Console.WriteLine(c1 == c2); // False (مقارنة مرجع)  
  
var s1 = new S { X = 5 };  
var s2 = new S { X = 5 };  
Console.WriteLine(s1.Equals(s2)); // True (مقارنة قيمة)  
  
var r1 = new R(5);  
var r2 = new R(5);  
Console.WriteLine(r1 == r2); // True (مقارنة قيمة)