In this video, we will discuss using datatype and displaycolumn attributes. [Please watch Part 40](http://csharp-video-tutorials.blogspot.com/2013/06/part-40-using-displayname-displayformat.html), before proceeding.

To understand **DataType**attribute, we will be using the same example, we used in [Part 40](http://csharp-video-tutorials.blogspot.com/2013/06/part-40-using-displayname-displayformat.html). Copy and paste the following code in **"Employee"** class, and navigate to localhost/MVCDemo/Home/Details/1.  
[MetadataType(typeof(EmployeeMetaData))]  
public partial class Employee  
{  
}  
  
public class EmployeeMetaData  
{  
    // Display mailto hyperlink  
    [DataType(DataType.EmailAddress)]  
    public string EmailAddress { get; set; }  
  
    // Display currency symbol. For country specific currency, set   
    // culture using globalization element in web.config.   
    // For Great Britain Pound symbol  
    // <globalization culture="en-gb"/>  
    [DataType(DataType.Currency)]  
    public int? Salary { get; set; }  
  
    // Generate a hyperlink  
    [DataType(DataType.Url)]  
    public string PersonalWebSite { get; set; }  
  
    // Display only Time Part  
    // [DataType(DataType.Time)]  
    // Display only Date Part  
    [DataType(DataType.Date)]  
    public DateTime? HireDate { get; set; }  
}

**DisplayColumn**attribute is useful, when a class has a property of complex type, and you want to pick one property of this complex object for display purpose. Let's understand this with an example.  
  
Right click on the **"Models"** folder and add **Company.cs** class file. Copy and paste the following code.   
public class Company  
{  
    public Employee CompanyDirector  
    {  
        get  
        {  
            SampleDBContext db = new SampleDBContext();  
            return db.Employees.Single(x => x.Id == 1);  
        }  
    }  
}  
  
Notice that, this class has **CompanyDirector**property which returns an **Employee**object. Employee is a complex type. Employee object has got several properties. If you want FullName to be used for display purpose, then make the following changes.  
  
Decorate **"Employee"** partial class in **"Models"** folder, with **DisplayColumn** attribute.  
[MetadataType(typeof(EmployeeMetaData))]  
[DisplayColumn("FullName")]  
public partial class Employee  
{  
}  
  
**Change "Details" action method in "Home" controller as shown below.**  
public ActionResult Details(int id)  
{  
    Company company = new Company();  
    return View(company);  
}  
  
**Copy and paste the following code in Details.cshtml view**  
@model MVCDemo.Models.Company  
  
@{  
    ViewBag.Title = "Details";  
}  
  
@Html.DisplayTextFor(x => x.CompanyDirector)  
  
Navigate to localhost/MVCDemo/Home/Details/1 and you should see the FullName of the employee.