Let's understand the difference between **TextBox**and **TextBoxFor** & **DropDownList**and **DropDownListFor** HTML helpers with an example. [Please watch Part 35](http://csharp-video-tutorials.blogspot.com/2013/06/part-35-setting-item-selected-when.html), before proceeding.

**Right click on the "Models" folder and add a class file with "name=Company.cs"**. Copy and paste the following code.  
public class Company  
{  
    private string \_name;  
    public Company(string name)  
    {  
        this.\_name = name;  
    }  
          
    public List<Department> Departments  
    {  
        get  
        {  
            SampleDBContext db = new SampleDBContext();  
            return db.Departments.ToList();  
        }  
    }  
  
    public string CompanyName  
    {  
        get  
        {  
            return \_name;  
        }  
        set  
        {  
            \_name = value;  
        }  
    }  
}

Copy and paste the following code in **HomeController**class. Notice that we are storing the **"Departments"** and **"CompanyName"** in the **ViewBag** object.  
public ActionResult Index()  
{  
    Company company = new Company("Pragim");  
  
    ViewBag.Departments = new SelectList(company.Departments, "Id", "Name");  
    ViewBag.CompanyName = company.CompanyName;  
  
    return View();  
}  
  
Right click on the **"Index"** action method in **"HomeController"** and add a view with **"name=Index"**. Copy and paste the following code. Notice that, here **the view is not strongly typed**, and we are hard-coding the name for **TextBox** and **DropDownList**HTML helpers.  
@{  
    ViewBag.Title = "Index";  
}  
  
<h2>Index</h2>  
@Html.TextBox("CompanyName", (string)ViewBag.CompanyName)  
<br />  
@Html.DropDownList("Departments", "Select Department")  
  
Add the following **"Index1"** action method to **"HomeController"**. Notice that we are passing **"Company"** object to the View, and hence the **view is strongly typed**. Since the view is strongly typed, we can use **TextBoxFor** and **DropDownListFor** HTML helpers.  
public ActionResult Index1()  
{  
    Company company = new Company("Pragim");  
    return View(company);  
}  
  
Right click on the **"Index1"** action method in **"HomeController"** and add a view with **"name=Index1"**. Copy and paste the following code.  
@model MVCDemo.Models.Company  
  
@{  
    ViewBag.Title = "Index1";  
}  
  
<h2>Index1</h2>  
@Html.TextBoxFor(m => m.CompanyName)  
<br />  
@Html.DropDownListFor(m => m.Departments, new SelectList(Model.Departments, "Id", "Name"), "Select Department")  
  
At this point, run the application and navigate to **"http://localhost/MVCDemo/home/index"**. A textbox and a dropdownlist will be rendered. Right click on the page and view it's source. The generated HTML is as shown below.  
<h2>Index</h2>  
<input id="CompanyName" name="CompanyName" type="text" value="Test" />  
<br />  
<select id="Departments" name="Departments"><option value="">Select Department</option>  
<option value="1">IT</option>  
<option value="2">HR</option>  
<option value="3">Payroll</option>  
</select>  
  
Now navigate to **"http://localhost/MVCDemo/home/index1"** and view page source. The HTML will be exactly the same as above.  
  
**So, in short, here are the differences**  
Html.TextBox amd Html.DropDownList are not strongly typed and hence they doesn't require a strongly typed view. This means that we can hardcode whatever name we want. On the other hand, Html.TextBoxFor and Html.DropDownListFor are strongly typed and requires a strongly typed view, and the name is inferred from the lambda expression.  
  
Strongly typed HTML helpers also provide compile time checking.

Since, in real time, we mostly use strongly typed views, prefer to use Html.TextBoxFor and Html.DropDownListFor over their counterparts.   
  
Whether, we use Html.TextBox & Html.DropDownList OR Html.TextBoxFor & Html.DropDownListFor, the end result is the same, that is they produce the same HTML.  
  
Strongly typed HTML helpers are added in MVC2.