**Important LINQ methods**

**0. Using Ienumerable**

Iterating over ienumerable is much faster than iterating over list

**1. SingleOrDefault**

Used to get a single matching element if one exists.

int[] array = { 1, 2, 3 };

int b = array.SingleOrDefault(element => element == 1); Console.WriteLine(b);

var customer = \_context.Customers. SingleOrDefault(c => c.Id == id);

**2. Where ()**

Return all elements that match a condition

List<string> fruits =

new List<string> { "apple", "passionfruit", "banana", "mango"}

IEnumerable<string> query = fruits.Where(fruit => fruit.Length < 6);

**3. IsNullOrWhiteSpace()**

It checks for string references that are null or have no data, and

string s = null;

if (string.IsNullOrEmpty(s) == true) for #, use hasValue property

**Select method**

Modfiy an element in an array or a collection

string[] array = { "cat", "dog", "mouse" };

var result = array.Select(element => element.ToUpper());

**@Html.editorFor**

Html.EditorFor renders input based on datatype, better than textbox. So for string, you would get a textbox, for numbers, you get sth else

**Logical Operators**

== Compares the string and numeric values between 2 varrialbes

is determine whether an object is a specific type

Length tells you the length of a variable

A property

4.RedirecToAction(‘method’, ‘controller’)

Direct to another action method in the controller

5. @Html. DisplayFor()

**Working with Lists**

Remove Remove (“dog”)

RemoveAt (index) Removes knowing the index

**String**

Length returns the length of string

stringName.Length

ToCharArray Converts a string into a bunch of alphabets

Equals()   
Comparing strings  
Str1.equals(str2)

Dates Initialization

DateTime dt = DateTime.Now

DateTime dt = new DateTime(2042, 12, 24);

ToShortDateString() Format: 8/23/2020