**Part 65 - C# Tutorial - Indexers in c#**

In this video we will discuss about **creating indexers**.  Let us understand indexers with an example. Create an asp.net web application. Add a class file, with name = **Company.cs**. Copy and paste the following code.

using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
  
namespace Demo  
{  
    public class Employee  
    {  
        public int EmployeeId { get; set; }  
        public string Name { get; set; }  
        public string Gender { get; set; }  
    }  
  
    public class Company  
    {  
        private List<Employee> listEmployees;  
  
        public Company()  
        {  
            listEmployees = new List<Employee>();  
  
            listEmployees.Add(new Employee   
            { EmployeeId = 1, Name = "Mike", Gender = "Male" });  
            listEmployees.Add(new Employee   
            { EmployeeId = 2, Name = "Pam", Gender = "Female" });  
            listEmployees.Add(new Employee   
            { EmployeeId = 3, Name = "John", Gender = "Male" });  
            listEmployees.Add(new Employee   
            { EmployeeId = 4, Name = "Maxine", Gender = "Female" });  
            listEmployees.Add(new Employee   
            { EmployeeId = 5, Name = "Emiliy", Gender = "Female" });  
            listEmployees.Add(new Employee   
            { EmployeeId = 6, Name = "Scott", Gender = "Male" });  
            listEmployees.Add(new Employee   
            { EmployeeId = 7, Name = "Todd", Gender = "Male" });  
            listEmployees.Add(new Employee   
            { EmployeeId = 8, Name = "Ben", Gender = "Male" });  
        }  
  
        // Use "this" keyword to create an indexer  
        // This indexer takes employeeId as parameter  
        // and returns employee name  
        public string this[int employeeId]  
        {  
            // Just like properties indexers have get and set accessors  
            get  
            {  
                return listEmployees.  
                    FirstOrDefault(x => x.EmployeeId == employeeId).Name;  
            }  
            set  
            {  
                listEmployees.  
                    FirstOrDefault(x => x.EmployeeId == employeeId).Name = value;  
            }  
        }  
    }  
}  
  
**Points to remember:**  
**1.** In the Company class constructor, we are initializing variable **"listEmployees"** and adding employees to the list.  
**2.** We then created an indexer using **"this"** keyword. This indexer takes **employeeId** as parameter and returns **employee name**.  
public string this[int employeeId]  
**3.** Just like properties indexers have **get** and **set** accessors.  
**4.** Indexers can also be overloaded. We will discuss about indexer overloading in our next video.

Now let's discuss about,  **using the indexer**, that we just created. Copy and paste the following code in WebForm1.aspx.cs  
Company company = new Company();  
Response.Write("Name of Employee with Id = 2: " + company[2]);  
Response.Write("<br/>");  
Response.Write("Name of Employee with Id = 5: " + company[5]);  
Response.Write("<br/>");  
Response.Write("Name of Employee with Id = 8: " + company[8]);  
  
Response.Write("<br/>");  
Response.Write("<br/>");  
  
Response.Write("Changing names of employees with Id = 2,5,8");  
Response.Write("<br/>");  
company[2] = "Employee 2 Name Changed";  
company[5] = "Employee 5 Name Changed";  
company[8] = "Employee 8 Name Changed";  
  
Response.Write("Name of Employee with Id = 2: " + company[2]);  
Response.Write("<br/>");  
Response.Write("Name of Employee with Id = 5: " + company[5]);  
Response.Write("<br/>");  
Response.Write("Name of Employee with Id = 8: " + company[8]);  
  
**Points to remember:**  
**1.** EmployeeId's **2,5** and **8** are passed into the company object, to retrieve the respective **employee names**. To retrieve the names of the employees, the **"get"** accessor of the indexer is used.  
**2.** To change the names of employees, we are again using the integral indexer defined on Company class.  
company[2] = "Employee 2 Name Changed";  
  
Notice that, because of the **"employeeId"** indexer, I am able to use company object like an array.