**C#:**

1. **What are generics in C#.NET?**

Generics are used to make reusable code classes to decrease the code redundancy, increase type safety and performance. Using generics, we can create collection classes. To create generic collection, System.Collections.Generic namespace should be used instead of classes such as ArrayList in the System.Collections namespace. Generics promote the usage of parameterized types.

1. **What are sealed classes in C#?**

We create sealed classes when we want to restrict the class to be inherited. Sealed modifier used to prevent derivation from a class. If we forcefully specify a sealed class as base class then a compile-time error occurs.

1. **What is method overloading?**

Method overloading is creating multiple methods with the same name with unique signatures in the same class. When we compile, the compiler uses overload resolution to determine the specific method to be invoke.

1. **MultiThreading:**  
   Multiple threads running at the same time and performing various tasks is referred as Multithreading. A thread is considered to be a lightweight process because it runs within the context of a program and takes advantage of resources allocated for that program.
2. **What is an object?**

An object is an instance of a class through which we access the methods of that class. “New” keyword is used to create an object. A class that creates an object in memory will contain the information about the methods, variables and behaviour of that class.

1. **Define Constructors?**

A constructor is a member function in a class that has the same name as its class. The constructor is automatically invoked whenever an object class is created. It constructs the values of data members while initializing the class.

1. **What’s the difference between an interface and abstract class?**

Interfaces have all the methods having only declaration but no definition. In an abstract class, we can have some concrete methods. In an interface class, all the methods are public. An abstract class may have private methods.

1. **What are delegates?**

Delegates are same are function pointers in C++ but the only difference is that they are type safe unlike function pointers. Delegates are required because they can be used to write much more generic type safe functions.

1. **What’s a multicast delegate?**

A delegate having multiple handlers assigned to it is called multicast delegate. Each handler is assigned to a method.

1. **What are indexers in C# .NET?**

Indexers are known as smart arrays in C#. It allows the instances of a class to be indexed in the same way as array.

1. **Data Abstraction:**

Data Abstraction is a concept in which the internal and superfluous details of the implementation of a logic is hidden from an end user(who is using the program) .A user can use any of the data and method from the class without knowing about how this is created or what is the complexity behind it. In terms of a real world example, when we drive a bike and change the gears we don’t have to care about how internally its working, like how lever is pulled or how chain is set.

1. **Inheritance**:

Inheritance is most popular Concept in OOP’s .This provides a developer an advantage called reusability of code. Suppose a class is written having functions with specific logic, then we can derive that class into our newly created class and we don’t have to write the logic again for derived class functions, we can use them as it is.

1. **Data Encapsulation:**

Wrapping up of member data and member functions of a class in a single unit is called encapsulation. The visibility of the member functions, data members is set via access modifiers used in class.

1. **Polymorphism**:

Poly means many and morphism means changing or alterable. The Concepts Introduces in the form of Many behaviours of an object.

**1. What is C#?**

C# is an object oriented, type safe and managed language that is compiled by .Net framework to generate [Microsoft](http://career.guru99.com/category/microsoft/) Intermediate Language.

|  |  |
| --- | --- |
|  |  |

**3. Can multiple catch blocks be executed?**

No, Multiple catch blocks can’t be executed. Once the proper catch code executed, the control is transferred to the finally block and then the code that follows the finally block gets executed.

**4. What is the difference between public, static and void?**

Public declared variables or methods are accessible anywhere in the application. Static declared variables or methods are globally accessible without creating an instance of the class. The compiler stores the address of the method as the entry point and uses this information to begin execution before any objects are created. And Void is a type modifier that states that the method or variable does not return any value.

**5. What is an object?**

An object is an instance of a class through which we access the methods of that class. “New” keyword is used to create an object. A class that creates an object in memory will contain the information about the methods, variables and behavior of that class.

**6. Define Constructors?**

A constructor is a member function in a class that has the same name as its class. The constructor is automatically invoked whenever an object class is created. It constructs the values of data members while initializing the class.

**7. What is Jagged Arrays?**

The array which has elements of type array is called jagged array. The elements can be of different dimensions and sizes. We can also call jagged array as Array of arrays.

**8. What is the difference between ref & out parameters?**

An argument passed as ref must be initialized before passing to the method whereas out parameter needs not to be initialized before passing to a method.

**9. What is the use of using statement in C#?**

The using block is used to obtain a resource and use it and then automatically dispose of when the execution of block completed.

**10. What is serialization?**

When we want to transport an object through network then we have to convert the object into a stream of bytes. The process of converting an object into a stream of bytes is called Serialization. For an object to be serializable, it should inherit ISerialize Interface.  
De-serialization is the reverse process of creating an object from a stream of bytes.

**11. Can “this” be used within a static method?**

We can’t use ‘This’ in a static method because we can only use static variables/methods in a static method.

**12. What is difference between constants and read-only?**

Constant variables are declared and initialized at compile time. The value can’t be changed after wards. Read-only variables will be initialized only from the Static constructor of the class. Read only is used only when we want to assign the value at run time.

**13. What is an interface class?**

Interface is an abstract class which has only public abstract methods and the methods only have the declaration and not the definition. These abstract methods must be implemented in the inherited classes.

**14. What are value types and reference types?**

Value types are stored in the Stack whereas reference types stored on heap.  
Value types:



|  |  |
| --- | --- |
| 1 | int, enum , byte, decimal, double, float, long |

Reference Types:



|  |  |
| --- | --- |
| 1 | string , class, interface, object |

**15. What are Custom Control and User Control?**

Custom Controls are controls generated as compiled code (Dlls), those are easier to use and can be added to toolbox. Developers can drag and drop controls to their web forms. Attributes can be set at design time. We can easily add custom controls to Multiple Applications (If Shared Dlls), If they are private then we can copy to dll to bin directory of web application and then add reference and can use them.  
User Controls are very much similar to ASP include files, and are easy to create. User controls can’t be placed in the toolbox and dragged – dropped from it. They have their design and code behind. The file extension for user controls is ascx.

**16. What are sealed classes in C#?**

We create sealed classes when we want to restrict the class to be inherited. Sealed modifier used to prevent derivation from a class. If we forcefully specify a sealed class as base class then a compile-time error occurs.

**17. What is method overloading?**

Method overloading is creating multiple methods with the same name with unique signatures in the same class. When we compile, the compiler uses overload resolution to determine the specific method to be invoke.

**18. What is the difference between Array and Arraylist?**

In an array, we can have items of the same type only. The size of the array is fixed. An arraylist is similar to an array but it doesn’t have a fixed size.

**19. Can a private virtual method be overridden?**

No, because they are not accessible outside the class.

**20. Describe the accessibility modifier “protected internal”.**

Protected Internal variables/methods are accessible within the same assembly and also from the classes that are derived from this parent class.

**21. What are the differences between System.String and System.Text.StringBuilder classes?**

System.String is immutable. When we modify the value of a string variable then a new memory is allocated to the new value and the previous memory allocation released. System.StringBuilder was designed to have concept of a mutable string where a variety of operations can be performed without allocation separate memory location for the modified string.

**22. What’s the difference between the System.Array.CopyTo() and System.Array.Clone() ?**

Using Clone() method, we creates a new array object containing all the elements in the original array and using CopyTo() method, all the elements of existing array copies into another existing array. Both the methods perform a shallow copy.

**23. How can we sort the elements of the array in descending order?**

Using Sort() methods followed by Reverse() method.

**24. Write down the C# syntax to catch exception?**

To catch an exception, we use try catch blocks. Catch block can have parameter of system.Exception type.

Eg:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | try  {  GetAllData();  }  catch(Exception ex)  {  } |

In the above example, we can omit the parameter from catch statement.

**25.   What’s the difference between an interface and abstract class?**

Interfaces have all the methods having only declaration but no definition. In an abstract class, we can have some concrete methods. In an interface class, all the methods are public. An abstract class may have private methods.

**26. What is the difference between Finalize() and Dispose() methods?**

Dispose() is called when we want for an object to release any unmanaged resources with them. On the other hand Finalize() is used for the same purpose but it doesn’t assure the garbage collection of an object.

**27. What are circular references?**

Circular reference is situation in which two or more resources are interdependent on each other causes the lock condition and make the resources unusable.

**28. What are generics in C#.NET?**

Generics are used to make reusable code classes to decrease the code redundancy, increase type safety and performance. Using generics, we can create collection classes. To create generic collection, System.Collections.Generic namespace should be used instead of classes such as ArrayList in the System.Collections namespace. Generics promotes the usage of parameterized types.

**29. What is an object pool in .NET?**

An object pool is a container having objects ready to be used. It tracks the object that is currently in use, total number of objects in the pool. This reduces the overhead of creating and re-creating objects.

**30. List down the commonly used types of exceptions in .Net?**

ArgumentException, ArgumentNullException , ArgumentOutOfRangeException, ArithmeticException, DivideByZeroException ,OverflowException , IndexOutOfRangeException ,InvalidCastException ,InvalidOperationException , IOEndOfStreamException , NullReferenceException , OutOfMemoryException , StackOverflowException etc.

**31. What are Custom Exceptions?**

Sometimes there are some errors that need to be handeled as per user requirements. Custom exceptions are used for them and are used defined exceptions.

**32. What are delegates?**

Delegates are same are function pointers in C++ but the only difference is that they are type safe unlike function pointers. Delegates are required because they can be used to write much more generic type safe functions.

**33. How do you inherit a class into other class in C#?**

Colon is used as inheritance operator in C#. Just place a colon and then the class name.



|  |  |
| --- | --- |
| 1 | public class DerivedClass : BaseClass |

**34. What is the base class in .net from which all the classes are derived from?**



|  |  |
| --- | --- |
| 1 | System.Object |

**35. What is the difference between method overriding and method overloading?**

In method overriding, we change the method definition in the derived class that changes the method behavior. Method overloading is creating a method with the same name within the same class having different signatures.

**36. What are the different ways a method can be overloaded?**

Methods can be overloaded using different data types for parameter, different order of parameters, and different number of parameters.

**37. Why can’t you specify the accessibility modifier for methods inside the interface?**

In an interface, we have virtual methods that do not have method definition. All the methods are there to be overridden in the derived class. That’s why they all are public.

**38. How can we set class to be inherited, but prevent the method from being over-ridden?**

Declare the class as public and make the method sealed to prevent it from being overridden.

**39. What happens if the inherited interfaces have conflicting method names?**

Implement is up to you as the method is inside your own class. There might be problem when the methods from different interfaces expect different data, but as far as compiler cares you’re okay.

**40. What is the difference between a Struct and a Class?**

Structs are value-type variables and classes are reference types. Structs stored on the stack, causes additional overhead but faster retrieval. Structs cannot be inherited.

**41. How to use nullable types in .Net?**

Value types can take either their normal values or a null value. Such types are called nullable types.

|  |  |
| --- | --- |
| 1  2  3  4 | Int? someID = null;  If(someID.HasVAlue)  {  } |

**42. How we can create an array with non-default values?**

We can create an array with non-default values using Enumerable.Repeat.

**43. What is difference between is and as operators in c#?**

“is” operator is used to check the compatibility of an object with a given type and it returns the result as Boolean.

“as” operator is used for casting of object to a type or a class.

**44. What’s a multicast delegate?**

A delegate having multiple handlers assigned to it is called multicast delegate. Each handler is assigned to a method.

**45. What are indexers in C# .NET?**

Indexers are known as smart arrays in C#. It allows the instances of a class to be indexed in the same way as array.

Eg:



|  |  |
| --- | --- |
| 1 | public int this[int index]    // Indexer declaration |

**46. What is difference between the “throw” and “throw ex” in .NET?**

“Throw” statement preserves original error stack whereas “throw ex” have the stack trace from their throw point. It is always advised to use “throw” because it provides more accurate error information.

**47. What are C# attributes and its significance?**

C# provides developers a way to define declarative tags on certain entities eg. Class, method etc. are called attributes. The attribute’s information can be retrieved at runtime using Reflection.

**48. How to implement singleton design pattern in C#?**

In singleton pattern, a class can only have one instance and provides access point to it globally.

Eg:

|  |  |
| --- | --- |
| 1  2  3  4 | Public sealed class Singleton  {  Private static readonly Singleton \_instance = new Singleton();  } |

**49. What is the difference between directcast and ctype?**

DirectCast is used to convert the type of an object that requires the run-time type to be the same as the specified type in DirectCast.

Ctype is used for conversion where the conversion is defined between the expression and the type.

**50. Is C# code is managed or unmanaged code?**

C# is managed code because Common language runtime can compile C# code to Intermediate language.

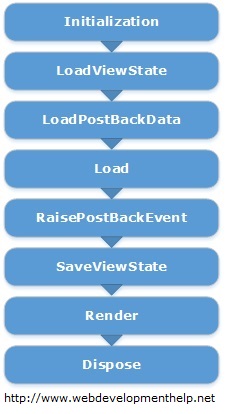
**ASP.Net:**

1. **Please briefly explain ASP.NET Page life Cycle?**

ASP.NET page passes through a series of steps during its life cycle. Following is the high-level explanation of life cycle stages/steps.

**Initialization:** Controls raise their Init event in this stage.Objects and variables are initializes for complete lifecyle of request.

**LoadViewState:** is a post back stage and loads the view state for the controls that enabled its view state property.



**LoadPostBackData:** is also a post back stage and loads the data posted for the controls and update them.

**Load:** In this stage page as well as all the controls raise their Load event. Till this stage all the controls are initialized and loaded. In most of the cases, we are coding this event handler.

**RaisePostBackEvent:** is again a postback stage. For example, it's raise against a button click event. We can easily put our code here to perform certain actions.

**SaveViewState:** Finally, controls state is saved in this stage before Rendering HTML.

**Render:** This is the stage where HTML is generated for the page.

**Dispose:** Lastly, all objects associated with the request are cleaned up.

For very detailed explanation of Page Life Cycle is explained [here](http://www.codeproject.com/Articles/73728/ASP-NET-Application-and-Page-Life-Cycle).

### What is the concept of view state in ASP.NET?

In order to maintain the state between post backs, ASP.NET provides a mechanism called view state. Hidden form fields are used to store the state of objects on client side and returned back to server in subsequent request (as postback occurs).

### Please briefly explain the usage of Global.asax?

Global.asax is basically ASP.NET Application file. It’s a place to write code for Application-level events such as Application start, Application end, Session start and end, Application error etc. raised by ASP.NET or by HTTP Modules.

There is a good list of events that are fired but following are few of the important events in Global.asax:

* Application\_Init occurs in case of application initialization for the very first time.
* Application\_Start fires on application start.
* Session\_Start fires when a new user session starts
* Application\_Error occurs in case of an unhandled exception generated from application.
* Session\_End fires when user session ends.
* Application\_End fires when application ends or time out.

### What are the types of Authentication in ASP.NET?

There are three types of authentication available in ASP.NET:

* Windows Authentication: This authentication method uses built-in windows security features to authenticate user.
* Forms Authentication: authenticate against a customized list of users or users in a database.
* Passport Authentication: validates against Microsoft Passport service which is basically a centralized authentication service.

### What are Session state modes in ASP.NET?

ASP.NET supports different session state storage options:

* **In-Process** is the default approach. It stores session state locally on same web server memory where the application is running.
* **StateServer** mode stores session state in a process other than the one where application is running. Naturally, it has added advantages that session state is accessible from multiple web servers in a Web Farm and also session state will remain preserved even web application is restarted.
* **SQLServer** mode stores session state in SQL Server database. It has the same advantages as that of StateServer.
* **Custom** modes allows to define our custom storage provider.
* **Off** mode disables session storage.

**ASP.Net MVC:**

### *MVC 4 INTERVIEW QUESTIONS AND ANSWERS*

**1. What is main objective of ASP.NET MVC 4 or What is new in MVC4 ?**

* Easy Mobile web applications (ASP.NET MVC 4 complete focus on **Mobile application development**)
* Full HTML5 support
* ASP.NET MVC web application with cloud support
* Working with different mobile and desktop web browsers

**Description:**

The main objective of ASP.NET MVC 4 is making to develop mobile web applications easily.Other than mobile web applications It’s focus is also on better HTML5 support and making ASP.NET MVC web application cloud ready.

By using new features of ASP.NET MVC 4 you can develop web applications that can work well across different desktop web browsers and mobile devices.

**2. What is WebAPI‘s in Asp.Net MVC4?**

* Web API is a new framework for consuming & building HTTP Services.
* Web API supports wide range of clients including different browsers and mobile devices.
* It is very good platform for developing RESTful services since it talk’s about HTTP.

**3. What is the use o f web API ? Why Web API needed, If you have already RESTful services using WCF ?**

Yes, we can still develop the RESTful services with WCF, but there are two main reasons that prompt users to use Web API instead of RESTful services.

* ASP.NET Web API is included in ASP.NET MVC which obviously increases TDD (Test Data Driven) approach in the development of RESTful services.
* For developing RESTful services in WCF you still needs lot of config settings, URI templates, contract’s & endpoints which developing RESTful services using web API is simple.

**4. What are the new enhancements done in default project template of ASP.NET MVC 4?**

* Adaptive rendering  for Nice Look & Feel
* Modern Looking for Mobile & Desktop browser

The new enhanced default project template came up with modern looking. Along with some cosmetic enhancements, it also employs new adaptive rendering to look nice in both desktop and mobile browsers without need of any kind of additional customization.

**5.  Why we need a separate mobile project template, while we can render our web application in mobile (What’s new in MVC 4 Mobile template) ?**

* Smart Phones & tablets touch got smart by using new jQuery.Mobile.MVC NuGet package.

The mobile project template touch optimized UI by using jQuery.Mobile.MVC NuGet Package for tablets and smart phones.

**6. What is the use of Display Modes?  
Ans.**

* **View** can be changed automatically based on browser(For mobile and desktop browser’s)

Display Modes is newly added feature in ASP.NET MVC 4. Views selected automatically by application depending on the browser. Example: If a desktop browser requests login page of an application it will return Views\Account\Login.cshtml view & if a mobile browser requests home page it will return Views\Account\Login.mobile.cshtml view.

**7. What are the main features of ASP.NET MVC 4 used by ASP.NET Web API?**

**Ans.**

* **Routing changes:** ASP.NET Web API uses same convention for config mapping that ASP.NET MVC provides.
* **Model Binding & Validation:** ASP.NET Web API uses same model binding functionality, but HTTP specific context related operations only.
* **Filters:** The ASP.NET Web API uses most of built-in filters from MVC.
* **Unit Testing:** Now Unit testing based on MVC, strongly unit testable.

**8. What are Bundling & Minification features in ASP.NET MVC 4?**

**Ans.**Bundling & Minification reduces number of HTTP requests. Bundling & Minification combines individual files into single. Bundled file for CSS & scripts and then it reduce’s overall size by minifying the contents of the bundle.

**9. What are the difference between asynchronous controller implementation b/w ASP.NET MVC 3 & ASP.NET MVC 4? Can you explain in detail?**

**Ans.**There is major difference is on implementation mechanism between ASP.NET MVC 3 and ASP.NET MVC 4.

In ASP.NET MVC 3, to implement async controller or methods we need to derive controller from AsyncController rather than from normal plain Controller class. We need to create 2 action methods rather than one. First with suffix  ‘Async’ keyword & second with ‘Completed’ suffix.

In  ASP.NET MVC 4 you need not to declare 2 action method. One can serve the purpouse. MVC 4 using .Net Framework 4.5 support for asynchronous communication.

**10. Is MVC 4 supporting Windows Azure SDK  (Software Development Kit) ?**

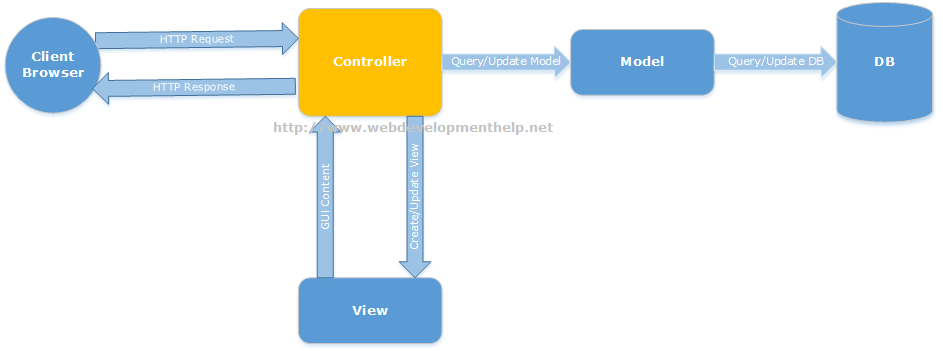
**Ans.**Yes, MVC 4 is supporting Windows Azure SDK version 1.6 or higher.

**MVC Frame Work:**

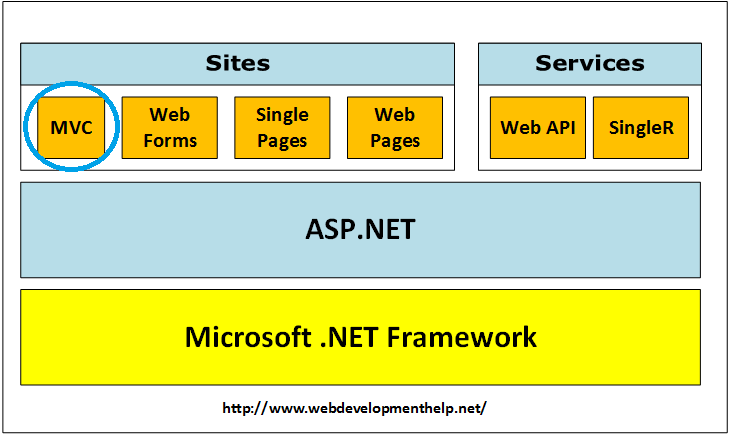
### 1. Explain MVC (Model-View-Controller) in general?

MVC (Model-View-Controller) is an architectural software pattern that basically decouples various components of a web application. By using MVC pattern, we can develop applications that are more flexible to changes without affecting the other components of our application.

* “Model”, is basically domain data.
* “View”, is user interface to render domain data.
* “Controller”, translates user actions into appropriate operations performed on model.

[](http://www.webdevelopmenthelp.net/wp-content/uploads/2013/09/MVC1.png)  
[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### 2. What is ASP.NET MVC?

ASP.NET MVC is a web development framework from Microsoft that is based on MVC (Model-View-Controller) architectural design pattern. Microsoft has streamlined the development of MVC based applications using ASP.NET MVC framework.  
[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### 3. Difference between ASP.NET MVC and ASP.NET WebForms?

ASP.NET Web Forms uses Page controller pattern approach for rendering layout, whereas ASP.NET MVC uses Front controller approach. In case of Page controller approach, every page has its own controller i.e. code-behind file that processes the request. On the other hand, in ASP.NET MVC, a common controller for all pages processes the requests.  
Follow the link for the [difference between the ASP.NET MVC and ASP.NET WebForms](http://www.webdevelopmenthelp.net/2013/10/difference-betweeen-asp-net-webforms-and-asp-net-mvc.html).  
[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### 4. What are the Core features of ASP.NET MVC?

Core features of ASP.NET MVC framework are:

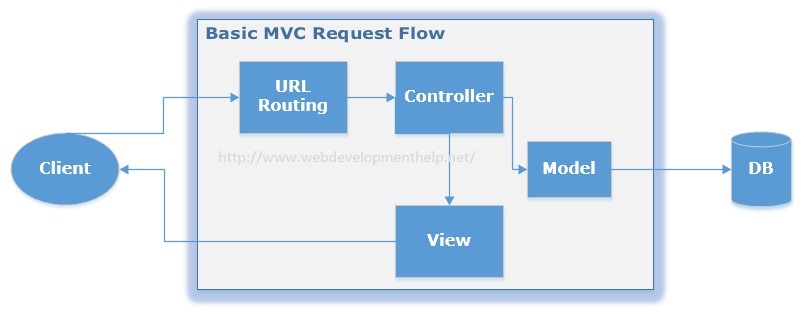
* ***Clear separation of application concerns*** (Presentation and Business Logic). It reduces complexity that makes it ideal for large scale applications where multiple teams are working.
* It’s an **extensible** as well as **pluggable framework**. We can plug components and further customize them easily.
* It provides extensive support for URL Routing that helps to make friendly URLs (means friendly for human as well as Search Engines).
* It supports for **Test Driven Development (TDD)** approach. In ASP.NET WebForms, testing support is dependent on Web Server but ASP.NET MVC makes it independent of Web Server, database or any other classes.
* Support for **existing ASP.NET features** like membership and roles, authentication and authorization, provider model and caching etc.

Follow for detailed understanding of above mentioned [core features](http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview).

[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### 5. Can you please explain the request flow in ASP.NET MVC framework?

Request flow for ASP.NET MVC framework is as follows:  
Request hits the controller coming from client. Controller plays its role and decides which model to use in order to serve the request. Further passing that model to view which then transforms the model and generate an appropriate response that is rendered to client.

You can follow the link, in order to understand the [Complete Application Life Cycle in ASP.NET MVC](http://www.webdevelopmenthelp.net/2015/04/understand-application-lifecycle-in-asp-net-mvc.html).  
[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### 6. What is Routing in ASP.NET MVC?

In case of a typical ASP.NET application, incoming requests are mapped to physical files such as .aspx file. On the other hand, ASP.NET MVC framework uses friendly URLs that more easily describe user’s action but not mapped to physical files. Let’s see below URLs for both ASP.NET and ASP.NET MVC.

**//ASP.NET  approach – Pointing to physical files (Student.aspx)**  
//Displaying all students  
http://locahost:XXXX/Student.aspx  
//Displaying a student by Id = 5  
http://locahost:XXXX/Student.aspx?Id=5

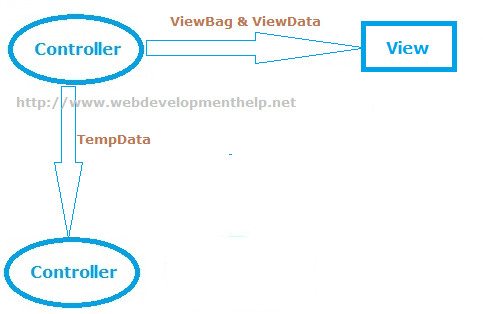
**//ASP.NET MVC approach – Pointing to Controller i.e. Student**  
//Displaying all students  
http://locahost:XXXX/***Student***  
//Displaying student by Id = 5  
http://locahost:XXXX/***Student***/**5**/

ASP.NET MVC framework uses a routing engine that maps URLs to controller classes. We can define routing rules for the engine, so that it can map incoming request URLs to appropriate controller. Practically, when a user types a URL in a browser window for an ASP.NET MVC application and presses “go” button, routing engine uses routing rules that are defined in Global.asax file in order to parse the URL and find out the path of corresponding controller. You can find complete details of [ASP.NET MVC Routing](http://www.webdevelopmenthelp.net/2014/03/understanding-routing-in-asp-net-mvc.html) here.  
[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### 7. What is the difference between ViewData, ViewBag and TempData?

In order to pass data from controller to view and in next subsequent request, ASP.NET MVC framework provides different options i.e. ViewData, ViewBag and TempData.

Both **ViewBag and ViewData** are used to to communicate between controller and corresponding view. But this communication is only for server call, it becomes null if redirect occurs. So, in short, it’s a mechanism to maintain state between controller and corresponding view. **ViewData** is a dictionary object while **ViewBag** is a dynamic property (a new C# 4.0 feature). ViewData being a dictionary object is accessible using strings as keys and also requires typecasting for complex types. On the other hand, ViewBag doesn’t have typecasting and null checks.

**TempData** is also a dictionary object that stays for the time of an HTTP Request. So, Tempdata can be used to maintain data between redirects i.e from one controller to the other controller.

You can easily find detailed examples for [implementation of ViewBag, ViewData and TempData](http://www.webdevelopmenthelp.net/2014/06/asp-net-mvc-pass-data-controller-view.html) here.  
[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### 8. What are Action Methods in ASP.NET MVC?

As I already explained about request flow in ASP.NET MVC framework that request coming from client hits controller first. Actually MVC application determines the corresponding controller by using routing rules defined in Global.asax. And controllers have specific methods for each user actions. Each request coming to controller is for a specific Action Method. The following code sample, “ShowBook” is an example of an Action Method.

  public ViewResult ShowBook(int id)  
  {  
           var computerBook = db.Books.Where(p => P.BookID == id).First();  
           return View(computerBook);  
  }

Action methods perform certain operation using Model and return result back to View. As in above example, ***ShowBook***is an action method that takes an Id as input, fetch specific book data and returns back to View as ViewResult. In ASP.NET MVC, we have many built-in ActionResults type:

* ViewResult
* PartialViewResult
* RedirectResult
* RedirectToRouteResult
* ContentResult
* JsonResult
* EmptyResult

For a complete list of available ActionResults types with Helper methods, please [click here](http://msdn.microsoft.com/en-us/library/system.web.mvc.actionresult(v=vs.118).aspx).

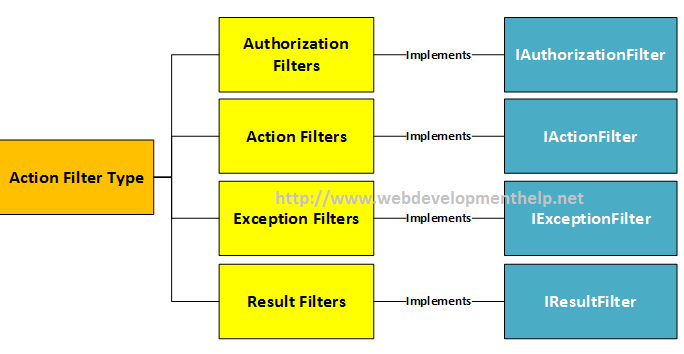
**Important Note:** All public methods of a Controller in ASP.NET MVC framework are considered to be Action Methods by default. If we want our controller to have a Non Action Method, we need to explicitly mark it with NonAction attribute as follows:

***[NonAction]***public void MyNonActionMethod() { ….. }

### 9. Explain the role of Model in ASP.NET MVC?

One of the core feature of ASP.NET MVC is that it separates the input and UI logic from business logic. Role of Model in ASP.NET MVC is to contain all application logic including validation, business and data access logic except view i.e. input and controller i.e UI logic.  
Model is normally responsible for accessing data from some persistent medium like database and manipulate it.

### 10. What are Action Filters in ASP.NET MVC?

If we need to apply some specific logic before or after action methods, we use action filters. We can apply these action filters to a controller or a specific controller action. Action filters are basically custom classes that provide a mean for adding pre-action or post-action behavior to controller actions.For example,

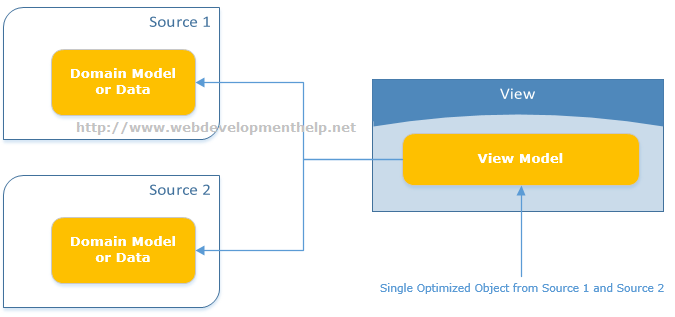
* Authorize filter can be used to restrict access to a specific user or a role.
* OutputCache filter can cache the output of a controller action for a specific duration.

### What is the difference between Razor View Engine and ASPX View Engine?

I have written a separate detailed blog post to understand the  [Difference between ASPX View Engine and Razor View Engine](http://www.webdevelopmenthelp.net/2014/10/aspx-view-engine-vs-razor-view-engine.html). You can follow the link to get detailed step by step description  [here](http://www.webdevelopmenthelp.net/2014/10/aspx-view-engine-vs-razor-view-engine.html). Most important differences are listed below:

|  |  |
| --- | --- |
| ASPX View Engine | Razor View Engine |
| WebForms View Engine uses namespace “***System.Web.Mvc.WebFormViewEngine”.*** | ***“System.Web.Razor”*** is the namespace for Razor View Engine. |
| Comparatively fast. | A little bit slower than ASPX View Engine. |
| Nothing like Test Driven Development | Good support for Test Driven Development. |
| Syntax for ASPX View Engine is inherited from Web Forms pages as: **<%= employee.FullName %>** | Syntax for Razor View Engine is comparatively less and clean. **@employee.FullName** |

### What is a ViewModel in ASP.NET MVC?

A ViewModel basically acts as a single model object for multiple domain models, facilitating to have only one optimized object for View to render. Below diagram clearly express the idea of ViewModel in ASP.NET MVC:

There are multiple scenarios where using ViewModel becomes obvious choice. For example:

* Parent-Child View Scenario
* Reports where often aggregated data required
* Model object having lookup data
* Dashboards displaying data from multiple sources

### What are ASP.NET MVC HtmlHelpers?

ASP.NET MVC HtmlHelpers fulfills almost the same purpose as that of ASP.NET Web From Controls. For imlementation point of view, HtmlHelper basically is a method that returns a string ( i.e. an HTML string to render HTML tags). So, in ASP.NET MVC we have HtmlHelpers for links, Images and for Html form elements etc. as follows:

@Html.ActionLink(“WebDev Consulting Company Profile”, “CompanyInfo”) will render:  
<a href=”/Site/CompanyInfo”>WebDev Consulting Company Profile</a>

and

@Html.TextBox(“strEmployeeName”) renders:  
<input id=”strEmployeeName” name=”strEmployeeName” type=”text” value=”” />

For a complete reference to Standard ASP.NET MVC Html Helpers, [Click Here](http://www.webdevelopmenthelp.net/2014/07/html-helpers-in-asp-net-mvc.html).

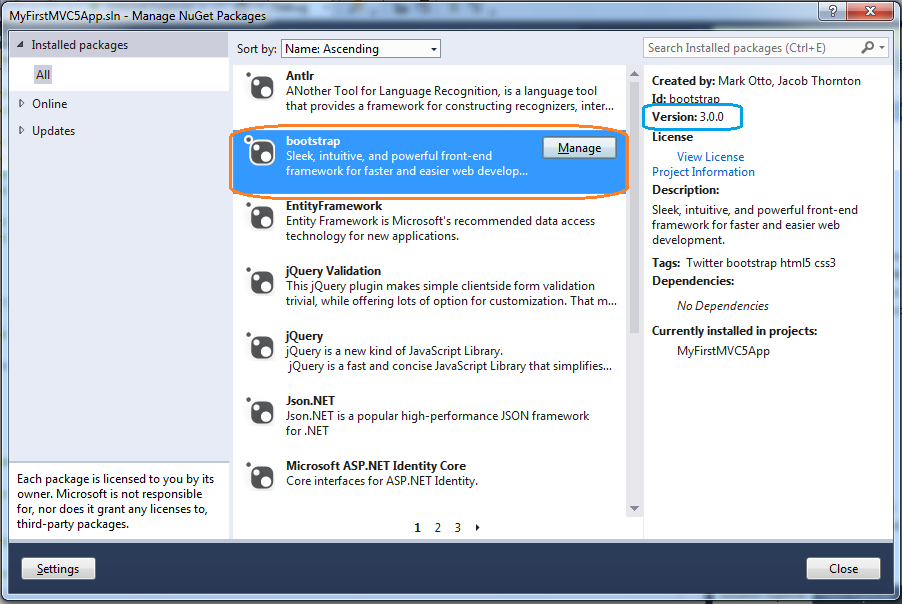
***Note:****Html Helpers are comparatively lightweight because these don’t have ViewState and event model as for ASP.NET Web Form Controls.*

We can also create our [Custom Html Helpers](http://www.webdevelopmenthelp.net/2014/07/custom-html-helper-in-aspnet-mvc.html) to fulfill specific application requirements.

There are 2 ways to create custom HtmlHelpers as follows:[](http://www.webdevelopmenthelp.net/2014/07/custom-html-helper-in-aspnet-mvc.html)  
[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

### What is Bootstrap in MVC5?

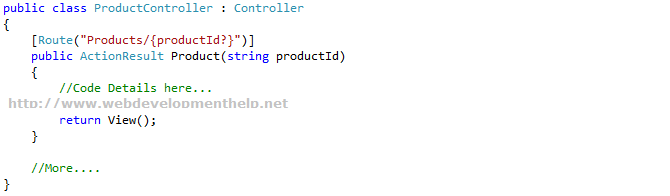
Bootstrap (a front-end framework) is an open source collection of tools that contains HTML and CSS-based design templates along with Javascript to create a responsive design for web applications. Bootstrap provides a base collection including layouts, base CSS, JavaScript widgets, customizable components and plugins.

Project Template in ASP.NET MVC5 is now using bootstrap that enhances look and feel with easy customization. Bootstrap version 3 is added to ASP.NET MVC5 template as shown below:[](http://www.webdevelopmenthelp.net/wp-content/uploads/2013/09/Bootstrap.png)

***Note****: You can*[*follow here*](http://www.webdevelopmenthelp.net/2015/07/bootstrap3-with-asp-net-mvc5.html)*to learn step by step how we can use any theme in Bootstraps to our ASP.NET MVC web application.*

### Kindly explain Attribute Routing in ASP.NET MVC5?

We already have discussed about Routing in [Question#6](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#Q6) that in ASP.NET MVC, we use friendly URLs that are mapped to controller’s actions instead of physical files as in case of ASP.NET WebForms. Now in ASP.NET MVC5, we can use attributes to define routes giving better control over the URIs.

We can easily define routes with Controller’s action methods as follows:

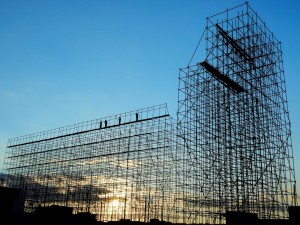
***Note****: Remember that conventional routing approach is not discarded, it’s still there and fully functional. Also, we can use both routing techniques in a same application.*[*Click here*](http://blogs.msdn.com/b/webdev/archive/2013/10/17/attribute-routing-in-asp-net-mvc-5.aspx)*for more details on Attribute Routing.*

### What is Scaffolding in ASP.NET MVC? and what are the advantages of using it?

We (developers) spent most of our time writing code for CRUD operations that is connecting to a database and performing operations like Create, Retrieve, Update and Delete. Microsoft introduces a very powerful feature called Scaffolding that does the job of writing CRUD operations code for us.

Scaffolding is basically a Code Generation framework. Scaffolding Engine generates basic controllers as well as views for the models using Micrsoft’s T4 template. Scaffolding blends with Entity Framework and creates the instance for the mapped entity model and generates code of all CRUD Operations. As a result we get the basic structure for a tedious and repeatative task.

You can find a detailed Web Development Tutorial with implementation on [ASP.NET MVC Scaffolding](http://www.webdevelopmenthelp.net/2015/04/scaffolding-in-asp-net-mvc.html) here.

[](http://www.webdevelopmenthelp.net/wp-content/uploads/2015/04/scaffolding_work_can_be_used_if_required.jpg)

Following are the few **advantages of Scaffolding**:

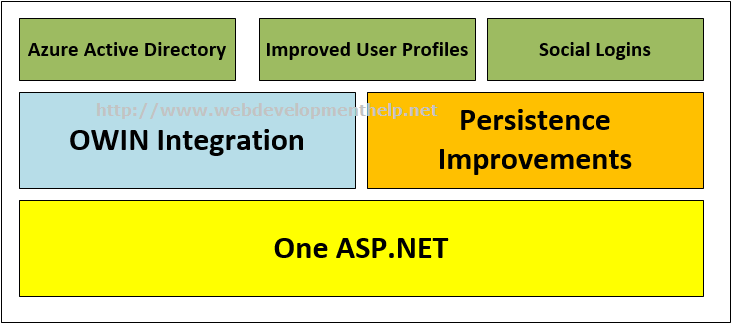
* RAD approach for data-driven web applications.
* Minimal effort to improve the Views.
* Data Validation based on database schema.
* Easily created filters for foreign key or boolean fields.

### Briefly explain ASP.NET Identity?

Microsoft introduces **ASP.NET Identity** as a system to manage access in ASP.NET application on premises and also in the cloud. There were issues with **Membership Provider Model** especially when we want to implement more advanced security features in our applications, so ASP.NET Identity gets away from the membership provider model.

If we look into the history of membership, its like follows:

* ASP.NET 2.0 Membership (VS 2005)
  + Forms Authentication
  + Sql Server Based Membership
* ASP.NET Simple Membership (VS 2010)
  + Easy to customize profile
  + ASP.NET Web Pages
* ASP.NET Universal Providers (VS2012)
  + Support Sql Azure

ASP.NET Identity is much improved system to manage access to our application and services.

### What are the new features introduced in ASP.NET MVC5?

ASP.NET MVC5 was introduced with following exciting features:

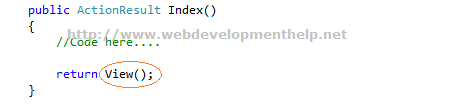
* [ASP.NET Identity](http://www.webdevelopmenthelp.net/2015/07/customize-profile-info-in-asp-net-identity.html)
* Authentication Filters
* Filter Overrides
* [Scaffolding](http://www.webdevelopmenthelp.net/2015/04/scaffolding-in-asp-net-mvc.html)
* [Bootstrap](http://www.webdevelopmenthelp.net/2015/07/bootstrap3-with-asp-net-mvc5.html)
* Attribute Routing

### What is a ViewEngine in ASP.NET MVC?

“View Engine in ASP.NET MVC is used to translate our views to HTML and then render to browser.”  
There are few View Engines available for ASP.NET MVC but commonly used View Engines are Razor, Web Forms/ASPX, NHaml and Spark etc. Most of the developers are familiar with Web Forms View Engine (ASPX) and Razor View Engine.

* Web Form View Engine was with ASP.NET MVC since beginning.
* Razor View Engine was introduced later in MVC3.
* NHaml is an open source view engine since 2007.
* Spark is also an open source since 2008.

**Note**: For a detailed comparison between Web Form View Engine and Razor View Engine, please follow[here](http://www.webdevelopmenthelp.net/2014/10/aspx-view-engine-vs-razor-view-engine.html).

[Back to top](http://www.webdevelopmenthelp.net/2013/09/Top-ASP-NET-MVC-Interview-Questions.html#top)

**Web API:**

**WCF:**

What is WSDL?

* WSDL stands for Web Services Description Language
* WSDL is written in XML
* WSDL is an XML document
* WSDL is used to describe Web services
* WSDL is also used to locate Web services
* WSDL is a W3C recommendation

**Asp.Net Web API VS Asp.Net MVC**

**Asp.Net MVC** is used to create web applications that return both views and data but **Asp.Net Web API** is used to create full blown **HTTP services** with easy and simple way that returns only data not view.

**Web API** helps to build **REST**-ful services over the **.NET Framework** and it also support content-negotiation (it's about deciding the best response format data that could be acceptable by the client.

It could be **JSON, XML, ATOM** or other formatted data), self hosting which are not in MVC.

**Web API** also takes care of returning data in particular format like **JSON, XML** or any other based upon the Accept header in the request and you don't worry about that.

**MVC** only return data in **JSON** format using **JsonResult**.

In **Web API** the request are mapped to the actions based on **HTTP** verbs but in **MVC** it is mapped to actions name.

**Asp.Net Web API** is new framework and part of the core **ASP.NET framework**.

The model binding, filters, routing and others MVC features exist in Web API are different from MVC and exists in the new **System.Web.Http** assembly.

In **MVC**, these features exist within **System.Web.Mvc**. Hence Web API can also be used with Asp.Net and as a standalone service layer.

You can mix **Web API** and **MVC controller** in a single project to handle advanced AJAX requests which may return data in **JSON, XML** or any others format and building a full blown HTTP service.

Typically, this will be called **Web API** self hosting.

When you have mixed **MVC** and **Web API** controller and you want to implement the authorization then you have to create two filters one for MVC and another for Web API since both are different.

Moreover, **Web API** is light weight architecture and except the web application it can also be used with smart phone apps.

Difference between WCF and Web API and WCF REST and Web Service

The .Net framework has a number of technologies that allow you to create HTTP services such as Web Service, WCF and now Web API.

Web Service

1. It is based on SOAP and return data in XML form.
2. It supports only HTTP protocol.
3. It is not open source but can be consumed by any client that understands xml.
4. It can be hosted only on IIS.

WCF

1. It is also based on SOAP and return data in XML form.
2. It is the evolution of the web service(ASMX) and support various protocols like TCP, HTTP, HTTPS, Named Pipes, MSMQ.
3. The main issue with WCF is, its tedious and extensive configuration.
4. It is not open source but can be consumed by any client that understands xml.
5. It can be hosted with in the applicaion or on IIS or using window service.

WCF Rest

1. To use WCF as WCF [Rest service](http://kellabyte.com/2011/09/04/clarifying-rest/) you have to enable webHttpBindings.
2. It support HTTP GET and POST verbs by [WebGet] and [WebInvoke] attributes respectively.
3. To enable other HTTP verbs you have to do some configuration in IIS to accept request of that particular verb on .svc files
4. Passing data through parameters using a WebGet needs configuration. The UriTemplate must be specified
5. It support XML, JSON and ATOM data format.

Web API

1. This is the new framework for building HTTP services with easy and simple way.
2. Web API is open source an ideal platform for building REST-ful services over the .NET Framework.
3. Unlike WCF Rest service, it use the full featues of HTTP (like URIs, request/response headers, caching, versioning, various content formats)
4. It also supports the MVC features such as routing, controllers, action results, filter, model binders, IOC container or dependency injection, unit testing that makes it more simple and robust.
5. It can be hosted with in the application or on IIS.
6. It is light weight architecture and good for devices which have limited bandwidth like smart phones.
7. Responses are formatted by Web API’s MediaTypeFormatter into JSON, XML or whatever format you want to add as a MediaTypeFormatter.

To whom choose between WCF or WEB API

1. Choose WCF when you want to create a service that should support special scenarios such as one way messaging, message queues, duplex communication etc.
2. Choose WCF when you want to create a service that can use fast transport channels when available, such as TCP, Named Pipes, or maybe even UDP (in WCF 4.5), and you also want to support HTTP when all other transport channels are unavailable.
3. Choose Web API when you want to create a resource-oriented services over HTTP that can use the full features of HTTP (like URIs, request/response headers, caching, versioning, various content formats).
4. Choose Web API when you want to expose your service to a broad range of clients including browsers, mobiles, iphone and tablets.

**WPF:**

**Entity Framework:**

**Entity Framework Interview Questions: Prepare for Data Modeling Questions**

Entity framework is a new Microsoft technology that allows programmers to connect to databases using data model objects. Instead of old data connections that required the developer to work with database connections, Entity framework allows the developer to create data model classes that do all of the backend work and leaves the developer with only data classes. These classes represent the tables in your database, so you can more easily understand your data and its relationship with other database objects.

**1) What is “code first” in relation to Entity framework?**

Entity framework uses the process of “code first” that maps database tables to data models. The data models in your .NET project are actually classes that represent the structure of your database tables. This code is used to work with your database, so you don’t need a direct connection to your database in each of your class methods.

**2) What is the difference between old ADO .NET and Entity framework coding techniques?**

When you used ADO, you connected to the database and had to define a stored procedure or query to retrieve data. With Entity framework, you don’t have to be “blind” when it comes to your tables. ADO did not allow you to get the table structure. With code first, you already have the table structure and Entity framework connects to the database and hides any connection processes. With Entity framework, you’re more aware of the database structure, which helps you avoid any coding mistakes. In addition, if the table structures change, Entity framework updates the data models for you during a refresh.

[**Understand how C# works with your data**](https://www.udemy.com/learn-c-make-a-databased-app-with-entity-framework/?tc=blog.entityframeworkinterviewquestions&utm_source=blog&utm_medium=udemyads&utm_content=post45588&utm_campaign=content-marketing-blog&xref=blog&couponCode=half-off-for-blog)

**3) What is LINQ?**

Language-Integrated Query (LINQ) is a way to query data without cumbersome stored procedures. Previously, programmers needed to create stored procedures and then call these stored procedures from their code. With Entity framework, you can pull data and query it using language similar to SQL.

[**Understand C# and Linq now**](https://www.udemy.com/object-oriented-programming-design-hitesh-choudhary/?tc=blog.entityframeworkinterviewquestions&utm_source=blog&utm_medium=udemyads&utm_content=post45588&utm_campaign=content-marketing-blog&xref=blog&couponCode=half-off-for-blog)

**4) How is data retrieved?**

The difference between older retrieval methods and current Entity framework retrieval methods is that you can now (with Entity) retrieve data as objects. The objects represent the tables (or linked tables) in your database. Instead of iterating through several columns and rows, you just use your class data models. For instance, if you have a table named “users,” you can use the “users” class instead of working through each data set after your query.

[**Understand C# and databases at Udemy.com**](https://www.udemy.com/csharpbasics/?tc=blog.entityframeworkinterviewquestions&couponCode=half-off-for-blog&utm_source=blog&utm_medium=udemyads&utm_content=post45588&utm_campaign=content-marketing-blog&xref=blog)

**5) Can you run SQL statements in an Entity framework environment?**

Yes, you can also run SQL query statements. You can use the “ExecuteStoreCommand” method to run SQL on your database. This is usually a secondary option from running simple LINQ on your Entity framework code. You can also run stored procedures from a database.

**6) How do you create a database model?**

Visual Studio has a database modeler available. You can create a database model from scratch, or you can query the database for the models. If you have a database already, you simply pull the database structures from your code and Entity framework will automatically set up the class data models.

**7) Does Entity framework support primary and foreign keys?**

Yes, Entity framework supports both types of primary and foreign keys. You can define these in your database tables and import them to your model classes. If you don’t already have a database set up, you can create these keys in your data model classes and their respective data modeling classes.

**8) How do you mark a data column as required by the database?**

You can “decorate” your data models. The “Required” decoration marks a field as required. Before a user can submit the data to the database, the user must have this field entered. You can also auto-generate required fields in your code, so the code automatically adds the required data.

**9) What is lazy loading?**

Lazy loading is a way to return only objects that are used. When you query the database model, lazy loading only returns the immediate tables needed by the user. All related tables are returned when they are used. This means you can reserve memory and storage when you work with large programs. It also means that objects are created until you need them, so it makes your program faster.

**10) What is eager loading?**

Eager loading is the opposite of lazy loading. When you query for objects, eager loading returns all of the objects including the related objects. For instance, when you query a list of customers and orders, eager loading loads all objects including the customers and the orders instead of just what you originally need (customers).

**11) What is a navigation property?**

A navigation property points to related tables in your database model. For instance, if you have a customer table that relates to the orders table, the navigation property points from the customers table to the orders table. While the primary and foreign keys are physical properties of the table, the navigation properties are a logical part of a data model. When you view the Entity framework model, you can view the navigation properties to better understand the structure of your tables.

**12) What is a scalar property in your data model?**

A scalar property is similar to a scalar property in the database. A scalar property points to one location in the table.

**13) What is a complex data type?**

A complex data type occurs when you need to point to another data object from one data object. For instance, if you have several tables that require an address, you can turn those addresses into a table of addresses. You then point your address columns to the new address table, which creates a complex data type. You will likely have complex data types in every .NET Entity framework project, because it makes it easier to relate one table and data model to another without creating cumbersome code.

**14) Overall, what is Entity framework?**

Entity framework is a type of ORM (object relationship model) that connects classes with database objects. It makes it easier to work with databases and tables without worrying about columns and rows. ORM makes it easier for you to query databases using LINQ, and you do not need to worry about your database connection. Additionally, you can create programs that are unaware of the database and its connection or type (Oracle, SQL Server or MySQL). Entity framework takes care of all of the back-end connection so you can worry about the queries and data.

**ADO.NET:**

ADO.Net Objects for Sql Data Provider:

Two Architecture’s for communicating to Sql Data source: **using System.Data.SqlClient;**

1. Connected Architechture:
   1. **SqlConnection:** Helps to identify the database server, the Database Name, Database username, Database Password. And other parameters that are required to connect to the Database.

**Ex: Sqlconnection** objConnection = new **SqlConnection(**” Data Source=Raviraj-PC; Initial Catalog=SQLTRAINING; User Id=sa; Password =12345; Integrated Security=true/SSPI”**);**

**(or)**

**objConnection.ConnectionString:** ” Data Source=Raviraj-PC; Initial Catalog=SQLTRAINING; User Id=sa; Password =12345; Integrated Security=true/SSPI”;

**SQLConnection Parameters:**

* + - 1. **Data Source:** This is used to recognize the DB Server
      2. Initial Catalog/Data Base: This is used to recognize the DB Name
      3. User Id: This is used to Authenticate a Particular user to That Database
      4. Password:
      5. Integrated Security: this specifies that user is secured :true
  1. **SqlCommand:** The Process of interacting with a database means that you must specify the actions you want to perform on DB. And this can be done by using SqlCommand.
     1. You use the Command Object to send a SQL Statements to the Database
     2. You can use the Command Object alone to execute a Command Directly, or assign a reference to a Command Object to a SqlDataAdapter, which holds a Set of Commands that work on a Group of Data.

Example: SqlCommand objCmd = new SqlCommand(“select \* from tbl”, objConnection)

(or)

objCmd.CommandText = “select \* from tbl”;

objCmd.Connection = objConnection;

**Note:** Before Executing a Command on a Database you have Open a Connection and Close the connection once you got the Data.

**Ex:** objConnection.Open (); for closing objConnection.Close ();

Execution of SQLCommand:

1. objCommand.ExecuteNonQuery() : for Executing Insert/Update/Delete Commands
   1. **Returns**: int i.e., no of Records Affected
2. objCommand.ExecuteReader(): for Select
   1. Returns a Reader which can have One Table.
3. objCommand.ExecuteScalar (): for Select and when we are getting scalar Output.
   1. Returns string value.

**SqlDataReader Object:** Thedata reader object allows you to obtain the result of SELECT statement from Command Object.

Note: This is good for only reading the Data but not manipulating it.

* SqlDataReader will have the data until the DB Connection is Open.

Example: SqlDataReader dr = objCommand.ExecuteReader();

objConnection.Close();

Connected Architechture: Connection🡪Command🡪 DataReader 🡪 UI

**Disconnected Architechture/Disconnected Model:**

1. **SqlConnection**:
2. **DataAdapter**:
   1. **SqlDataAdapter**: It is a Connection of Commands which acts like a Bridge between the Database and the **DataSet**.

NOTE: Data Adapter can always be binded to over single Table at a Time.

Data Set: It is a Collection of data tables.

DataAdapter Methods:

1. Fill(): Used to read the data from the Database based on the SELECT Statement of Select Command of Data Adapter and fills the Retrieved data into a DataSet.

SqlDataAdapter obj = new SqlDataAdapter(“select prod\_name from products”, cnn);

DataSet dsObj = new DataSet();

obj.Fill(dsObj, “products”);

1. Update():
2. Dataset:

**SQL:**

**What is normalization?**

**Definition :** Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored. There are several benefits for using Normalization in Database.

**Benefits :**

1. Eliminate data redundancy
2. Improve performance
3. Query optimization
4. Faster update due to less number of columns in one table
5. Index improvement

**HTML5:**

**Major goals of the**[**HTML specification**](http://www.w3.org/TR/html5/)**were to:**

* Deliver rich content (graphics, movies, etc.) without the need for additional plugins (e.g., Flash).
* Provide better semantic support for web page structure through the introduction of new structural element tags.
* Provide a stricter parsing standard to simplify error handling, ensure more consistent cross-browser behavior, and simplify backward compatibility with documents written to older standards.
* Provide better cross-platform support (i.e., to work well whether running on a PC, Tablet, or Smartphone).

**Key new features of HTML5 include:**

* Improved support for embedding graphics, audio, and video content via the new [<canvas>](http://www.w3schools.com/tags/tag_canvas.asp), [<audio>](http://www.w3schools.com/tags/tag_audio.asp), and [<video>](http://www.w3schools.com/tags/tag_video.asp) tags.
* Extensions to the JavaScript API such as [geolocation](http://www.w3schools.com/html/html5_geolocation.asp) and [drag-and-drop](http://www.w3schools.com/html/html5_draganddrop.asp) as well for [storage](http://www.w3schools.com/html/html5_webstorage.asp) and [caching](http://www.w3schools.com/html/html5_app_cache.asp).
* Introduction of [“web workers”](http://www.w3schools.com/html/html5_webworkers.asp).
* Several new semantic tags were also added to complement the structural logic of modern web applications. These include the<main>, <nav>, <article>, <section>, <header>, <footer>, and <aside> tags.
* New form controls, such

as <calendar>, <date>, <time>, <email>, <url>, and <search>.

### 1.What’s new HTML 5 DocType and Charset?

Normally for HTML files first line of code is DocType which basically tells browser about specific version of HTML. HTML5 is now not subset of SGML. As compared to previous version/standards of HTML,

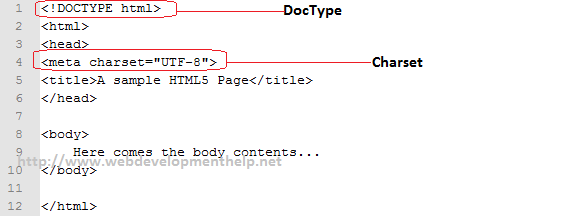
DocType is simplified as follows:

***<!doctype html>***

And HTML 5 uses UTF-8 encoding as follows:

***<meta charset=”UTF-8″>***

You can see a very simple HTML5 page below:

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### 2.How can we embed Audio in HTML5?

HTML 5 comes with a standard way of embedding audio files as previously we don’t have any such support on a web page. Supported audio formats are as follows:

* **MP3**
* **Wav**
* **Ogg**.

Below is the most simple way to embed an audio file on a web page.

*<audio controls>*  
*<source src=”jamshed.mp3″ type=”audio/mpeg”>*  
*Your browser does’nt support audio embedding feature.*  
*</audio>*

HTML5 Audio Tag

In above code, ***src*** value can be relative as well as absolute URL. We can also use multiple **<source>**elements pointing to different audio files. There are more new attributes for <audio> tag other than **src**as below:

* **controls** – it adds controls such as volume, play and pause.
* **autoplay** – it’s a boolean value which specifies that audio will start playing once it’s ready.
* **loop** – it’s also a boolean value which specifies looping (means it automatically start playing after it ends).
* **preload** – auto, metadata and none are the possible values for this attribute.
  + auto means plays as it loaded.
  + metadata displays audio file’s associated data
  + none means not pre-loaded.

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### 3.How can we embed Video in HTML 5?

Same like audio, HTML 5 defined standard way of embedding video files which was not supported in previous versions/standards. Supported video formats are as follows:

* **MP4**
* **WebM**
* **Ogg**

Please look into below sample code.

*<video width=”450″ height=”340″ controls>*  
*<source src=”Racing.mp4″ type=”video/mp4″>*  
*Your browser does’nt support video embedding feature.*  
*</video>*

Same like <audio>, <video> tag has associated optional attributes as controls, autoplay, preload, loop, poster, width, height and other global attributes etc. Controls, loop, preload and autoplay are already explained above. Others are explained below:

* **poster** – it’s basically a URL of the image that needs to display until video get started.
* **width** – video player width
* **height** – video player’s height

### 4. What are the new media element in HTML 5 other than audio and video?

HTML 5 has strong support for media. Other than audio and video tags, it comes with the following tags:

**<embed> Tag:**<embed> acts as a container for external application or some interactive content such as a plug-in. Special about <embed> is that it doesn’t have a closing tag as we can see below:

<embed type=”video/quicktime” src=”Fishing.mov”>

**<source> Tag:**<source> is helpful for multiple media sources for audio and video.

*<video width=”450″ height=”340″ controls>*  
*<source src=”jamshed.mp4″ type=”video/mp4″>*  
*<source src=”jamshed.ogg” type=”video/ogg”>*  
*</video>*

**<track> Tag:**<track> defines text track for media like subtitles as:

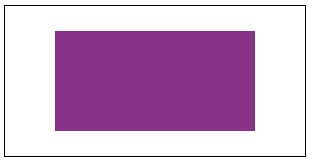
*<video width=”450″ height=”340″ controls>*  
*<source src=”jamshed.mp4″ type=”video/mp4″>*  
*<source src=”jamshed.ogg” type=”video/ogg”>*     <track kind=”subtitles” label=”English” src=”jamshed\_en.vtt” srclang=”en” default></track>      <track kind=”subtitles” label=”Arabic” src=”jamshed\_ar.vtt” srclang=”ar”></track>  
*</video>*

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### 5.What is the usage of canvas Element in HTML 5?

<canvas> is an element in HTML5 which we can use to draw graphics with the help of scripting (which is most probably JavaScript).  
This element behaves like a container for graphics and rest of things will be done by scripting. We can draw images, graphs and a bit of animations etc using <canvas> element.

<canvas id=”canvas1″ width=”300″ height=”100″>  
</canvas>

[](http://www.webdevelopmenthelp.net/wp-content/uploads/2014/09/Canvas.jpg)

As canvas is considered to be the most exciting feature in HTML5, following resources can be helpful to enhance one’s skill in this area. I have listed few good websites on HTML5 canvas as well as available tools and libraries:

* [**Canvas From Scratch**](http://code.tutsplus.com/series/canvas-from-scratch--net-19650)
* Starting from scratch and follow the step by step approach to take to advance level.
  + Introduction to Canvas in HTML5
  + Understanding advanced **Drawing** Topics
  + **Tranformation**, **Shadows** and **Gradients** in HTML5 Canvas
  + and finally **Pixel Manipulation** in Canvas
* [**Mozilla Developer Network – MDN**](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial)  
  MDN acts a complete reference guide for HTML5 with detailed examples and code snippets.
* [**HTML5 Canvas Tutorial**](http://www.html5canvastutorials.com/)  
  Canvas Tutorial is a good tutorial site for learning basics of HTML5 canvas topics including Lines, Curves, Paths, Shapes, Filling Styles, Images, Text Manipulation, Tranformation, Composites and Animation etc.
* [**PlayCanvas**](https://playcanvas.com/)  
  PlayCanvas is basically a **WebGL Game Engine** with a set of developer tools that can be used to develop **3D games** for browser as well as mobiles.

For a more detailed reference to **HTML5 Canvas** resource, you can go to Emily Heming’s tutorial on this blog  
“[Top 20 Resources You Need To Master HTML5 Canvas](http://www.webdevelopmenthelp.net/2015/02/mastering-html5-canvas.html)“.

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

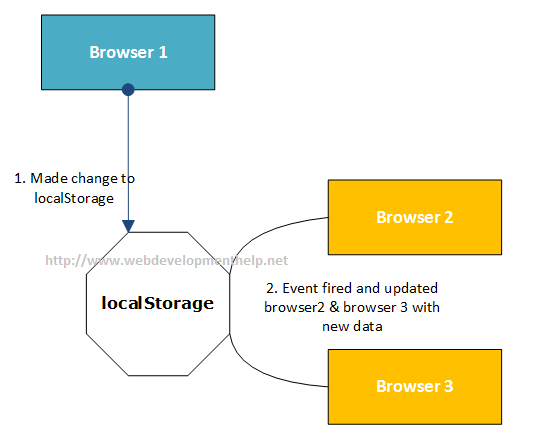
### 6.What are the different types of storage in HTML 5?

HTML 5 has the capability to store data locally. Previously it was done with the help of cookies.  
Exciting thing about this storage is that its fast as well as secure.

There are two different objects which can be used to store data.

* **localStorage**object stores data for a longer period of time even if the browser is closed.
* **sessionStorage**object stores data for a specific session.

|  |  |
| --- | --- |
| sessionStorage | localStorage |
| It persists data until we close the window or tab in which it was stored. | It persist data even if the window or tab is closed (but can be explicitly removed or expires). |
| Values stored in ***sessionStorage*** are not shared. These will be visible only to respective window or tab. | Values stored in ***localStorage*** are shared for all windows and tabs from same origin. |
| Maximum size is 5MB. | Maximum size for localStorage is more between 10-15MB. |

Working with localStorage is quite simple and having following methods:

* ***localStorage.getItem(key)*** -> fetch an item from storage against provided key.
* ***localStorage.setItem(key, value)*** -> add an item to storage.
* ***localStorage.removeItem(key)***-> removes an item from storage against provided key.
* ***localStorage.clear()*** -> clearing the storage removing all items from it.

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### 7.What are the new Form Elements introduced in HTML 5?

There are a number of new form elements has been introduced in HTML 5 as follows:

* **datalist** provides functionality for auto-complete feature.
* **datetime** facilitate selecting a datetime along with Time Zone.
* **output** represents the result of a calculation.
* **keygen** generates a key-pair field in a form to implement secure authentication.
* **date** is an input field for date and applies validation accordingly.
* **month** for selecting a month and year in a form input field.
* **week** for selecting a week and year in an input field.
* **time** is an input field for selecting time i.e. Hours:Minutes: AM/PM. For example, 10:30 AM.
* **color** is an input field for color.
* **number** that only allows numeric values.
* **range** is an input field for selecting value within a specified range.
* **email** is input field for email with standard email validations.
* **url** is for an URL(Uniform Resource Locator) and validated accordingly.

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### 8.What are the deprecated Elements in HTML5 from HTML4?

Elements that are deprecated from HTML 4 to HTML 5 are:

* frame
* frameset
* noframe
* applet
* big
* center
* basefront

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### 9.What are the new APIs provided by HTML 5 standard?

HTML 5 standard comes with a number of new APIs. Few of it are as follows:

* Media API
* Text Track API
* Application Cache API
* User Interaction
* Data Transfer API
* Command API
* Constraint Validation API
* History API
* and many more….

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### 10.What is the difference between HTML 5 Application Cache and regular HTML Browser Cache?

One of the key feature of HTML 5 is “Application Cache” that enables us to make an offline version of a web application. It allows to fetch few or all of website contents such as HTML files, CSS, images, javascript etc locally. This feature speeds up the site performance. This is achieved with the help of a manifest file defined as follows:

*<!doctype html>*  
*<html manifest=”example.appcache”>*  
*…..*  
*</html>*

As compared with traditional browser caching, Its not compulsory for the user to visit website contents to be cached.  
[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

#### [Recommended Online Training Courses on HTML5 and Related](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#acc-2)

* [**Complete HTML5 and CSS3 Course +1 start to finish project**](http://click.linksynergy.com/link?id=s8waqo9RNjU&offerid=323058.595728&type=2&murl=https%3A%2F%2Fwww.udemy.com%2Fcomplete-html5-and-css3-course-1-start-to-finish-project%2F) [Usman Raoof and Fahad Chaudhry] 94 Lectures, 8.5 Hours Video, All Levels  
  Every little thing in html5 and css3 is covered and +1 complete project will help you to create an amazing website.
* [**Build Responsive Real World Websites with HTML5 and CSS3**](http://click.linksynergy.com/link?id=s8waqo9RNjU&offerid=323058.437398&type=2&murl=https%3A%2F%2Fwww.udemy.com%2Fdesign-and-develop-a-killer-website-with-html5-and-css3%2F) [Jonas Schmedtmann] 77 Lectures, 11.5 hour video, All Level  
  The easiest way to learn modern web design, HTML5 and CSS3 step-by-step from scratch. Design AND code a huge project.
* [**Learn Bootstrap Development By Building 10 Projects**](http://click.linksynergy.com/link?id=s8waqo9RNjU&offerid=323058.427530&type=2&murl=https%3A%2F%2Fwww.udemy.com%2Flearn-bootstrap-development-by-building-10-projects%2F)  
  [Eduonix Learning Solutions] 74 Lectures, 13.5 hour video, All Level  
  Latest approach of web application development

### Briefly explain cache manifest file in HTML5 with an example?

Cache manifest file is simply a text file that dictates the browser, what to store for offline access? It basically list down the required resources for offline access.

#### [Following is an example of a simple manifest file:](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#acc-3)

CACHE MANIFEST  
/decorate.css  
/work.js  
/amazing.jpg

So, the resources mentioned in above manifest file (decorate.css, work.js, and amazing.jpg) will be downloaded and cached locally for offline access.

**Note:** Remember to add manifest attribute for each page of our website we want to be cached.

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### Is it possible to get the geographical location of a user using HTML5?

Yes. It’s quiet possible to get the geographical location of a user using HTML5. As it has user’s privacy concerns so user’s approval needed. As we discussed above about new HTML5 API including Media API, Text Track API, Application Cache, User Interaction, Data Transfer API, Command API etc: HTML5 introduced new Geolocation API for locating a user’s position.

In order to get user position, getCurrentPosition() method is used as:

                  navigator.geolocation.getCurrentPosition(show\_map);

Above code simply provides an idea how we can retrieve geographical location of a user. But while providing professional implementation, we will get user approval and also handle errors accordingly.

#### [Important Note](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#acc-4)

If you want to integrate **Google Maps in a Website or Mobile application** like Android, iOS and don’t know how to integrate and create programs in Google Maps by using Google Maps JavaScript APIs? you can find this complete Video Course really helpful:

* [**Google Maps JavaScript API for beginners**](http://click.linksynergy.com/link?id=s8waqo9RNjU&offerid=323058.506938&type=2&murl=https%3A%2F%2Fwww.udemy.com%2Fgoogle-maps-javascript-api-for-beginners2%2F) by Girish Shakya having **46 lectures**, **2 hours videos**, beginner level.

Using HTML5 and JavaScript, you will be:

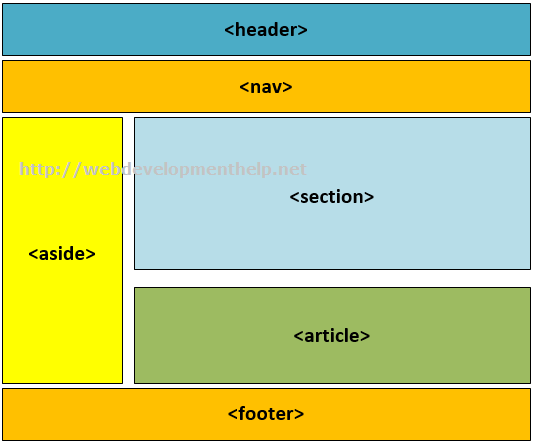
* able to setup Google Maps APIs
* able to create Google Maps for devices
* able to create Google Maps in a website
* able to control the Maps UI events

For detailed course outline, [follow here](http://click.linksynergy.com/link?id=s8waqo9RNjU&offerid=323058.506938&type=2&murl=https%3A%2F%2Fwww.udemy.com%2Fgoogle-maps-javascript-api-for-beginners2%2F).

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### What are HTML5 Semantic Elements? Explain with Example.

Semantic elements are those elements that clearly explains the purpose or meaning of the element to user  
(developer). For example, <div> and <span> elements in HTML doesn’t explain what they will contain as contents. On the other hand, <img> and <form> elements clearly explains the contents it can contain.

HTML5 introduces many new semantic elements with few are demonstrated in below diagram:

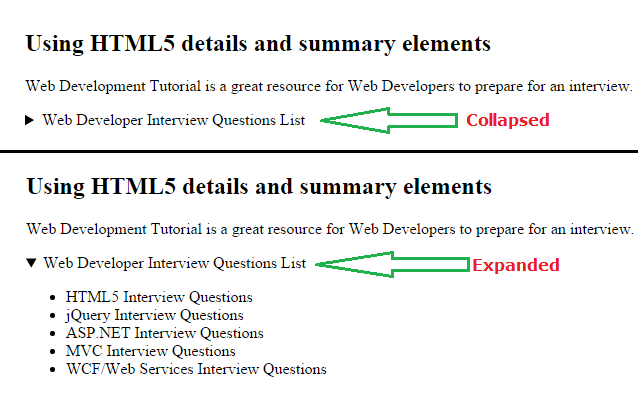
<header>, <nav>, <aside>, <section>, <article> and <footer> elements clearly explains the meaning of each element.

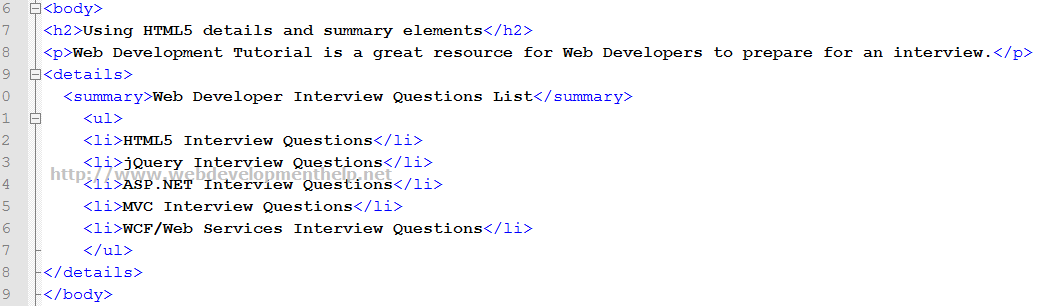
There are more semantic elements as listed below:

* <details>
* <figure>
* <figcaption>
* <main>
* <mark>
* <nav>
* <summary>
* <time>

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

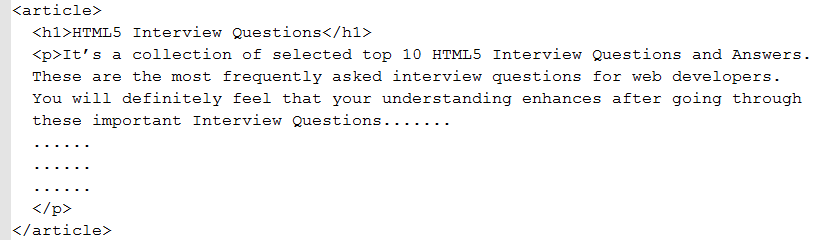
### How to use HTML5 details and summary elements?

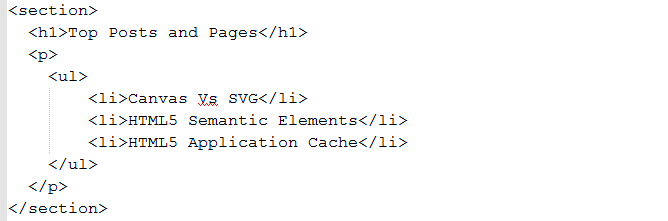
HTML5 details and summary element can be used collectively to produce the below collapse/Expanse affect as follows:

Following is the source code of HTML5 details and summary element to produce above output.

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### Article Vs Section tags in HTML5?

An <article> tag is a complete and independent piece of content of a document or page. For Example, this article about HTML5 interview Question is a complete and independent piece of content on this page that covers a specific topic.

On the other hand, <section> tag refers to specific section of a document or page for grouping purpose. For Example, at the end of this page, we have a specific section for related list of “Top Interview Questions And Answers Series”. Or in right nav, we have a section for “Top Posts and Pages” of this website.

An <article> can wrap a <section> to represents a sub-topic as follows:

<article>  
<h1>Interview Questions Series</h1>  
<p>List of Web Development Interview Question with detailed answers for web developers.</p>  
<section>  
<h1>HTML5 Interview Questions</h1>  
<p>List of Interview Questions on HTML5</p>  
</section>  
<section>  
<h1>JavaScript Interview Questions</p>  
<p>List of Interview Questions on JavaScript</p>  
</section>  
<section>  
<h1>ASP.NET MVC Interview Questions</p>  
<p>List of Interview Questions on ASP.NET MVC</p>  
</section>  
</article>  
[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### What is an HTML5 Web Worker?

Normally if some script is executing in an HTML page, the page remains unresponsive until the scripts execution stops. But an **HTML5 web worker** is a script (i.e. JavaScript) that keeps executing in background. At the same time user can interact with the page and will not feel any performance degradation.

HTML5 web worker normally exists in external files and used for long-running CPU intensive tasks but without affecting the User Interface or other scripts.  
[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### What are the limitations of HTML5 Web Worker?

HTML5 Web worker seems to be very handy in many scenarios (especially for CPU intensive tasks) but it has certain limitations. Few JavaScript objects are not accessible to HTML5 web worker as:

* parent object
* window object
* document object

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### Canvas Vs SVG?

Following table clearly explains the difference between Canvas and SVG (Scalable Vector Graphics):

|  |  |
| --- | --- |
| Canvas | SVG |
| Only ONE HTML Element for rendering graphics i.e. canvas element | Multiple Graphics Elements including (Circle, Rect, Boxes, Path, Line, Polygon etc.) |
| Draws Graphics on the fly using only Script i.e. JavaScript. | Support Script as well as CSS. |
| Primarily based on Pixels. | Based on Graphics elements as discussed above. |
| No manipulation using Event Handling due to pixel based interaction. | Can manipulate by attaching event handlers to SVG elements. |
| Better in Performance. | Slow in rendering when manipulating complex scenarios |

[Back to top](http://www.webdevelopmenthelp.net/2013/04/HTML5-Interview-Questions.html#top)

### WHAT IS ANGULARJS ?

“AngularJS is a JavaScript framework which simplifies binding JavaScript objects with HTML UI elements.”  
Let us try to understand the above definition with simple sample code.  
Below is a simple “Customer” function with “CustomerName” property. We have also created an object called as “Cust” which is of “Customer” class type.  
Hide Copy Code  
function Customer()  
{  
this.CustomerName = “AngularInterview”;  
}  
var Cust = new Customer();  
Now let us say the above customer object we want to bind to a HTML text box called as “TxtCustomerName”. In other words when we change something in the HTML text box the customer object should get updated and when something is changed internally in the customer object the UI should get updated.  
Hide Copy Code  
So in order to achieve this communication between UI to object developers end up writing functions as shown below. “UitoObject” function takes data from UI and sets it to the object while the other function “ObjecttoUI” takes data from the object and sets it to UI.  
Hide Copy Code  
function UitoObject()  
{  
Cust.CustomerName = $(“#TxtCustomerName”).val();  
}  
function ObjecttoUi()  
{  
$(“#TxtCustomerName”).val(Cust.CustomerName);  
}  
So if we analyze the above code visually it looks something as shown below. Your both functions are nothing but binding code logic which transfers data from UI to object and vice versa.  
Now the same above code can be written in Angular as shown below. The javascript class is attached to a HTML parent div tag using “ng-controller” directive and the properties are binded directly to the text box using “ng-model” declarative.  
So now whatever you type in the textbox updates the “Customer” object and when the “Customer” object gets updated it also updates the UI.  
Hide Copy Code  
In short if you now analyze the above code visually you end up with something as shown in the below figure.You have the VIEW which is in HTML, your MODEL objects which are javascript functions and the binding code in Angular.  
Now that binding code have different vocabularies.  
Some developers called it “ViewModel” because it connects the “Model” and the “View”.  
Some call it “Presenter” because this logic is nothing but presentation logic.  
Some term it has “Controller” because it controls how the view and the model will communicate.  
To avoid this vocabulary confusion Angular team has termed this code as “Whatever”. It’s that “Whatever” code which binds the UI and the Model. That’s why you will hear lot of developers saying Angular implements “MVW” architecture.  
Explain Directives in Angular?  
Directives are attributes decorated on the HTML elements. All directives start with the word “ng”. As the name says directive it directs Angular what to do.  
For example below is a simple “ng-model” directive which tells angular that the HTML textbox “txtCustomerName” has to be binded with the “CustomerName” property.  
Hide Copy Code  
Some of the most commonly used directives are ng-app,ng-controller and ng-repeat.

### WHAT ARE CONTROLLERS AND NEED OF NG-CONTROLLER AND NG-MODEL IN ANGULAR?

“Controllers” are simple javascript function which provides data and logic to HTML UI. As the name says controller they control how data flows from the server to HTML UI.  
For example below is simple “Customer” controller which provides data via “CustomerName” and “CustomerCode” property and Add/ Update logic to save the data to database.  
Note: – Do not worry too much about the $scope , we will discuss the same in the next question.  
Hide Copy Code  
function Customer($scope)  
{  
$scope.CustomerName = “Shiv”;  
$scope.CustomerCode = “1001”;  
$scope.Add = function () {  
}  
$scope.Update = function () {  
}  
}  
“ng-controller” is a directive.Controllers are attached to the HTML UI by using the “ng-controller” directive tag and the properties of the controller are attached by using “ng-model” directive. For example below is a simple HTML UI which is attached to the “Customer” controller via the “ng-controller” directive and the properties are binded using “ng-model” directive.  
Hide Copy Code

### WHAT ARE EXPRESSIONS IN ANGULAR?

Angular expressionsare unit of code which resolves to value. This code is written inside curly braces “{“.  
Below are some examples of angular expressions:-  
The below expression adds two constant values.  
Hide Copy Code  
{{1+1}}  
The below expression multiplies quantity and cost to get the total value.  
Hide Copy Code  
The value total cost is {{ quantity \* cost }}  
The below expression displays a controller scoped variable.  
Hide Copy Code  
The value of Customer code is {{CustomerCode}}  
The value of Customer code is {{CustomerCode}}  
How can we initialize Angular application data?  
We can use “ng-init” directive to achieve the same. You can see in the below example we have used “ng-init” directive to initialize the “pi” value.  
Hide Copy Code  
The value of pi is {{pi}}

### EXPLAIN $SCOPE IN ANGULAR?

“$scope” is an object instance of a controller. “$scope” object instance get’s created when “ng-controller” directive is encountered.  
For example in the below code snippet we have two controllers “Function1” and “Function2”. In both the controllers we have a “ControllerName” variable.  
Hide Copy Code  
function Function1($scope)  
{  
$scope.ControllerName = “Function1”;  
}  
function Function2($scope)  
{  
$scope.ControllerName = “Function2”;  
}  
Now to attach the above controllers to HTML UI we need to use “ng-controller” directive. For instance you can see in the below code snippet how “ng-controller” directive attaches “function1” with “div1” tag and “function2” with “div2” tag.  
Hide Copy Code  
Instance of {{ControllerName}} created  
Instance of {{ControllerName}} created  
So this is what happens internally. Once the HTML DOM is created Angular parser starts running on the DOM and following are the sequence of events:-  
The parser first finds “ng-controller” directive which is pointing to “Function1”. He creates a new instance of “$scope” object and connects to the “div1” UI.  
The parser then starts moving ahead and encounters one more “ng-controller” directive which is pointing to “Function2”. He creates a new instance of “$scope” object and connects to the “div2” UI.  
Now once the instances are created, below is a graphical representation of the same. So the “DIV1” HTML UI is binded with “function1” $scope instance and the “DIV2” HTML UI is binded with “function2” $scope instance. In other words now anything changes in the $scope object the UI will be updated and any change in the UI will update the respective $scope object.

### WHAT IS “$ROOTSCOPE” AND HOW IS IT RELATED WITH “$SCOPE”?

“$rootScope” is a parent object of all “$scope” angular objects created in a web page.  
Let us understand how Angular does the same internally.

Below is a simple Angular code which has multiple “DIV” tags and every tag is attached to a controller. So let us understand step by step how angular will parse this and how the “$rootScope” and “$scope” hierarchy is created.  
The Browser first loads the above HTML page and creates a DOM (Document object model) and Angular runs over the DOM.Below are the steps how Angular creates the rootscope and scope objects.  
Step 1:- Angular parser first encounters the “ng-app” directive and creates a “$rootScope” object in memory.  
Step 2:- Angular parser moves ahead and finds the expression {{SomeValue}}. It creates a variable  
Step 3:- Parser then finds the first “DIV” tag with “ng-controller” directive which is pointing to “Function1” controller. Looking at the “ng-controller” directive it creates a “$scope” object instance for “Function1” controller. This object it then attaches to “$rootScope” object.  
Step 4:- Step 3 is then repeated by the parser every time it finds a “ng-controller” directive tag. Step 5 and Step 6 is the repetition of Step 3.  
If you want to test the above fundamentals you can run the below sample Angular code. In the below sample code we have created controllers “Function1” and “Function2”. We have two counter variables one at the root scope level and other at the local controller level.  
Hide Copy Code  
function Function1($scope, $rootScope)  
{  
$rootScope.Counter = (($rootScope.Counter || 0) + 1);  
$scope.Counter = $rootScope.Counter;  
$scope.ControllerName = “Function1”;  
}  
function Function2($scope, $rootScope)  
{  
$rootScope.Counter = (($rootScope.Counter || 0) + 1);  
$scope.ControllerName = “Function2”;  
}  
var app = angular.module(“myApp”, []); // creating a APP  
app.controller(“Function1”, Function1); // Registering the VM  
app.controller(“Function2”, Function2);  
</script  
Below is the HTML code for the same. You can we have attached “Function1” and “Function2” two times with “ng-controller” which means four instances will be created.  
Hide Copy Code  
Global value is {{Counter}}  
Child Instance of {{ControllerName}} created :- {{Counter}}  
Child Instance of {{ControllerName}} created :- {{Counter}}  
Child Instance of {{ControllerName}} created :- {{Counter}}  
Child Instance of {{ControllerName}} created :- {{Counter}}  
Above is the output of the code you can see the global variable of root scope has be incremented four times because four instances of $scope have been created inside “$rootScope” object.

### DO I NEED JQUERY FOR ANGULAR?

No , you do not need Jquery for Angular. It’s independent of Jquery.

### HOW IS THE DATA BINDING IN ANGULAR ?

Its two way binding. So whenever you make changes in one entity the other entity also gets updated.

### EXPLAIN COMPILE AND LINK PHASE?

At the heart Angular framework is a parser. A parser which parses the Angular directives and render’s HTML output.  
Angular parser works in 3 steps:-  
Step 1:- HTML browser parses the HTML and creates a DOM (Document Object Model).  
Step 2:- Angular framework runs over this DOM looks at the Angular directives and manipulates the DOM accordingly.  
Step 3:- This manipulated is then rendered as HTML in the browser.  
Now the above angular parsing is not so simple as it looks to be. It occurs in two phases “Compile” and “Link”. Firs the compile phase occurs then the link phase.  
In compile phase the angular parser starts parsing the DOM and whenever the parser encounters a directive it create a function. These functions are termed as template or compiled functions. In this phase we do not have access to the $scope data.  
In the link phase the data i.e. ($scope) is attached to the template function and executed to get the final HTML output.

### HOW DO WE MAKE HTTP GET AND POST CALLS IN ANGULAR?

To make HTTP calls we need to use the “$http” service of Angular. In order to use the http services you need to make provide the “$http” as a input in your function parameters as shown in the below code.  
Hide Copy Code  
function CustomerController($scope,$http)  
{  
$scope.Add = function()  
{  
$http({ method: “GET”, url: “http://localhost:8438/SomeMethod” }).success(function (data, status, headers, config)  
{  
// Here goes code after success  
}  
}  
}  
“$http” service API needs atleast three things:-  
First what is the kind of call “POST” or “GET”.  
Second the resource URL on which the action should happen.  
Third we need to define the “success” function which will be executed once we get the response from the server.  
Hide Copy Code  
$http({ method: “GET”, url: “http://localhost:8438/SomeMethod” }).success(function (data, status, headers, config)  
{  
// Here goes code after success  
}  
**How do we pass data using HTTP POST in Angular ?**You need to pass data using the “data” keyword in the “$http” service API function. In the below code you can see we have created a javascript object “myData” with “CustomerName” property. This object is passed in the “$http” function using HTTP POST method.  
Hide Copy Code  
Var myData = {};  
myData.CustomerName = “Test”;  
$http({ method: “POST”,  
data: myData,  
url: “http://www.xyz.com”})  
.success(function (data, status, headers, config)  
{  
// Here goes code after success  
}

### WHAT IS DEPENDENCY INJECTION AND HOW DOES IT WORK IN ANGULAR?

Dependency injection is a process where we inject the dependent objects rather than consumer creating the objects. DI is everywhere in Angular or we can go one step ahead and say Angular cannot work without DI.  
For example in the below code “$scope” and “$http” objects are created and injected by the angular framework. The consumer i.e. “CustomerController” does not create these objects himself rather Angular injects these objects.  
Hide Copy Code  
function CustomerController($scope,$http)  
{  
// your consumer would be using the scope and http objects  
}

### HOW DOES DI BENEFIT IN ANGULAR?

There are two big benefits of DI: – Decoupling and Testing.  
Let’s first start with Decoupling. Consider your application has a logger functionality which helps to log errors , warning etc in some central place. This central place can be a file, event viewer, database etc.  
Hide Copy Code  
function FileLogger()  
{  
this.Log = function () {  
alert(“File logger”);  
};  
}  
function EventLogger()  
{  
this.Log = function () {  
alert(“Event viewer logger”);  
};  
}  
Now let’s say you have a “Customer” class who wants to use the “Logger” classes. Now which “Logger” class to use depends on configuration.  
So the code of “Customer” is something as shown below. So depending on the configuration “Customer” class either creates “FileLogger” or it creates “EventLogger” object.  
Hide Copy Code  
function Customer($scope, Logger)  
{  
$scope.Logger = {};  
if (config.Loggertype = “File”)  
{  
$scope.Logger = new FileLogger();  
}  
else  
{  
$scope.Logger = new EventLogger();  
}  
}  
But with DI our code becomes something as shown below. The “Customer” class says he is not worried from where the “Logger” object comes and which type of “Logger” objects are needed .He just wants to use the “Logger” object.  
Hide Copy Code  
function Customer($scope,$http, Logger)  
{  
$scope.Logger = Logger;  
}  
With this approach when a new “Logger” object gets added the “Customer” class does not have to worry about the new changes because the dependent objects are injected by some other system.  
The second benefit of DI is testing. Let’s say you want to test the “Customer” class and you do not have internet connection. So your “$http” object method calls can throw errors. But now you can mock a fake “$http” object and run your customer class offline without errors.The fake object is injected using DI.

### WHAT ARE SERVICES IN ANGULAR?

Service helps to implement dependency injection. For instance let’s say we have the below “Customer” class who needs “Logger” object. Now “Logger” object can be of “FileLogger” type or “EventLogger” type.  
Hide Copy Code  
function Customer($scope,$http, Logger)  
{  
$scope.Logger = Logger;  
}  
So you can use the “service” method of the application and tie up the “EventLogger” object with the “Logger” input parameter of the “Customer” class.  
Hide Copy Code  
var app = angular.module(“myApp”, []); // creating a APP  
app.controller(“Customer”, Customer); // Registering the VM  
app.service(“Logger”, EventLogger); // Injects a global Event logger object  
So when the controller object is created the “EventLogger” object is injected automatically in the controller class.

### ARE SERVICE OBJECT INSTANCES GLOBAL OR LOCAL?

Angular Services create and inject global instances. For example below is a simple “HitCounter” class which has a “Hit” function and this function increments the variable count internally every time you call hit the button.  
Hide Copy Code  
function HitCounter()  
{  
var i = 0;  
this.Hit = function ()  
{  
i++;  
alert(i);  
};  
}  
This “HitCounter” class object is injected in “MyClass” class as shown in the below code.  
Hide Copy Code  
function MyClass($scope, HitCounter)  
{  
$scope.HitCounter = HitCounter;  
}  
Below code advises the Angular framework to inject “HitCounter” class instance in the “MyClass” class. Read the last line of the below code specially which says to inject the inject the “HitCounter” instance.  
Hide Copy Code  
var app = angular.module(“myApp”, []); // creating a APP  
app.controller(“MyClass”, MyClass); // Registering the VM  
app.service(“HitCounter”, HitCounter); // Injects the object  
Now let’s say that the “Controller” “MyClass” is attached to twodiv tag’s as shown in the below figure.  
So two instances of “MyClass” will be created. When the first instance of “MyClass” is created a “HitCounter” object instance is created and injected in to “MyClass” first instance.  
When the second instance of “MyClass” is created the same “HitCounter” object instance is injected in to second instance of “MyClass”.  
Again I repeat the same instance is injected in to the second instance, new instances are not created.  
If you execute the above code you will see counter values getting incremented even if you are coming through different controller instances.

### WHAT IS A FACTORY IN ANGULAR?

“Factory” in real world means a premise where products are manufactured. Let’s take an example of a computer manufacturing firm. Now the company produces different kinds and sizes of computers likelaptops,desktops, tablets etc.  
Now the process of manufacturing the computer products are same with slight variation. To manufacture any computer we need processor, RAM and hard disk. But depending on what kind of final case packing is the final product shapes.  
That’s what the use of Factory in Angular.  
For example see the below code we have a “Customer”, “Phone” and “Address” class.  
Hide Copy Code  
function Customer()  
{  
this.CustomerCode = “1001”;  
this.CustomerName = “Shiv”;  
}  
function Phone()  
{  
this.PhoneNumber = “”;  
}  
function Address()  
{  
this.Address1 = “”;  
this.Address2 = “”;  
}  
So now we would create different types of “Customer” object types using the combination of “Address” and “Phones” object.  
We would like to combine “Customer” with “Address” and create a “Customer” object which has “Address” collection inside it.  
Or must be we would like to create “Customer” object with “Phone” objects inside it.  
Or must be “Customer” object with both “Phone” and “Address” objects.  
In other words we would like to have different permutation and combination to create different types of “Customer” objects.  
So let’s start from bottom. Let’s create two factory function’s one which creates “Address” object and the other which creates “Phone” objects.  
Hide Copy Code  
functionCreateAddress()  
{  
var add = new Address();  
return add;  
}  
functionCreatePhone()  
{  
var phone = new Phone();  
return phone;  
}  
Now let’s create a main factory function which uses the above two small factory functions and gives us all the necessary permutation and combination.  
In the below factory you can see we have three functions:-  
“CreateWithAddress” which creates “Customer” with “Address” objects inside it.  
“CreateWithPhone” which creates “Customer” object with “Phone” objects inside it.  
“CreateWithPhoneAddress” which creates “Customer” object with aggregated “Phone” and “Address” objects.  
Hide Copy Code  
function CreateCustomer() {  
return {  
CreateWithAddress: function () {  
varcust = new Customer();  
cust.Address = CreateAddress();  
returncust;  
},  
CreateWithPhone: function () {  
varcust = new Customer();  
cust.Phone = {};  
cust.Phone = CreatePhone();  
returncust;  
}  
,  
CreateWithPhoneAddress: function () {  
debugger;  
varcust = new Customer();  
cust.Phone = CreatePhone();  
cust.Address = CreateAddress();  
returncust;  
}  
}  
}  
Below is a simple “CustomerController” which takes “CustomerFactory” as the input. Depending on “TypeOfCustomer” it creates with “Address” , “Phones” or both of them.  
Hide Copy Code  
functionCustomerController($scope, Customerfactory)  
{  
$scope.Customer = {};  
$scope.Init = function(TypeofCustomer)  
{  
if (TypeofCustomer == “1”)  
{  
$scope.Customer = Customerfactory.CreateWithAddress();  
}  
if (TypeofCustomer == “2”)  
{  
$scope.Customer = Customerfactory.CreateWithPhone();  
}  
if (TypeofCustomer == “3”) {  
$scope.Customer = Customerfactory.CreateWithPhoneAddress();  
}  
}  
}  
You also need to tell Angular that the “CreateCustomer” method needs to be passed in the input. For that we need to call the “Factory” method and map the “CreateCustomer” method with the input parameter “CustomerFactory” for dependency injection.  
Hide Copy Code  
var app = angular.module(“myApp”, []); // creating a APP  
app.controller(“CustomerController”, CustomerController); // Register the VM  
app.factory(“Customerfactory”, CreateCustomer);  
So if we consume the “CustomerController” in UI , depending on situation it creates different flavors of “Customer” object. You can in the below code we have three different “DIV” tags and depending on the “TypeofCustomer” we are displaying data.

### WHAT IS THE DIFFERENCE BETWEEN FACTORY AND SERVICE?

“Factory” and “Service” are different ways of doing DI (Dependency injection) in angular. Please read the previous question to understand what is DI.  
So when we define DI using “service” as shown in the code below. This creates a new GLOBAL instance of the “Logger” object and injects it in to the function.  
Hide Copy Code  
app.service(“Logger”, Logger); // Injects a global object  
When you define DI using a “factory” it does not create a instance. It just passes the method and later the consumer internally has to make calls to the factory for object instances.  
Hide Copy Code  
app.factory(“Customerfactory”, CreateCustomer);  
Below is a simple image which shows visually how DI process for “Service” is different than “Factory”.  
Factory Service  
Usage When we want to create different types of objects depending on scenarios. For example depending on scenario we want to create a simple “Customer” object , or “Customer” with “Address” object or “Customer” with “Phone” object. See the previous question for more detailed understanding. When we have utility or shared functions to be injected like Utility , Logger , Error handler etc.  
Instance No Instance created. A method pointer is passed. Global and Shared instance is created.

### HOW ARE VALIDATIONS IMPLEMENTED IN ANGULAR?

Angular leverages HTML 5 validations and new form element types to implement validation.  
For instance below is a simple form which has two text boxes. We have used HTML 5 “required” validation attribute and a form element of type “email”.  
Hide Copy Code  
Name :- Email :-  
Below are some example of new form elements introduced in HTML 5 and Angular works with almost all of them :-  
Color.  
Date  
Datetime-local  
Email  
Time  
Url  
Range  
Telephone  
Number  
Search  
When you run the above HTML inside a browser which understands HTML 5 , you will see your validations and form types in actions as shown in the below browser screen shot.  
Angular leverages HTML 5 validation attributes and new HTML 5 form elements. Now if we want Angular to handle validation we need first stop HTML 5 to do validation. So for that the first step is to specify “novalidate” attribute on the form tag.  
Hide Copy Code  
—–  
So now the HTML will not fire those validations it will be routed to the Angular engine to further take actions.  
In other words when end user fills data in the HTML UI , validation events are routed to Angular framework and depending on scenario Angular sets a field called as “$Valid”. So if the validations are fine it sets it to “True” or else its sets it to “False”.  
So you can see in the below code we have attached the angular controller and models to the text boxes. Watch the code of the button it has “ng-disabled” attribute which is set via the “$Valid” property in a NEGATED fashion.  
Negated fashion means when there is no error it should enable the button and when there are errors that means it’s false it should disable the button.  
Hide Copy Code  
Name:-  
Email :-  
Note :- “Name” is needed for the validations to work.  
How to check error validation for a specific field?  
To check for a specific field you need to use the below DOM code.  
Hide Copy Code  
!frm1.CustomerName.$valid

### WHAT DOES SPA (SINGLE PAGE APPLICATION) MEAN?

SPA is a concept where rather loading pages from the server by doing post backs we create a single shell page or master page and load the webpages inside that master page.

### HOW CAN WE IMPLEMENT SPA WITH ANGULAR?

By using Angular routes.

### HOW TO IMPLEMENT ROUTING IN ANGULAR?

Implementing Angular route is a five step process: –  
Step 1: – Add the “Angular-route.js” file to your view.  
Hide Copy Code  
Step 2: – Inject “ngroute” functionality while creating Angular app object.  
Hide Copy Code  
var app = angular.module(“myApp”, [‘ngRoute’]);  
Step 3: – Configure the route provider.  
In route provider we need to define which URL pattern will load which view. For instance in the below code we are saying “Home” loads “Yoursite/Home” view and “Search” loads “YourSite/Search” view.  
Hide Copy Code  
app.config([‘$routeProvider’,  
function ($routeProvider) {;  
$routeProvider.  
when(‘/Home, {  
templateUrl: ‘Yoursite/Home’,  
controller: ‘HomeController’  
}).  
when(‘/Search’, {  
templateUrl: YourSite/Search’,  
controller: ‘SearchController’  
}).  
otherwise({  
redirectTo: ‘/’  
});  
}]);  
Step 4: – Define hyperlinks.  
Define hyper link with the “#” structure as shown below. So now when user clicks on the below anchor hyperlinks, these actions are forwarded to route provider and router provider loads the view accordingly.  
Hide Copy Code  
Home  
Search

### STEP 5: – DEFINE SECTIONS WHERE TO LOAD THE VIEW.

Once the action comes to the router provider it needs a place holder to load views. That’s defined by using the “ng-view” tag on a HTML element. You can see in the below code we have created a “DIV” tag with a place holder. So the view will load in this section.  
Hide Copy Code  
So if we summarize angular routing is a three step process (Below is a visual diagram for the same): –  
Step 1: – End user clicks on a hyperlink or button and generates action.  
Step 2: – This action is routed to the route provider.  
Step 3: – Router provider scans the URL and loads the view in the place holder defined by “ng-view” attribute.  
How can we create a custom directive in Angular?  
Till now we have looked in to predefined Angular directives like “ng-controller”,”ng-model” and so on. But what if we want to create our own custom Angular directive and attach it with HTML elements as shown in the below code.  
Hide Copy Code  
To create a custom directive we need to use the “directive” function to register the directive with angular application. When we call the “register” method of “directive” we need to specify the function which will provide the logic for that directive.  
For example in the below code we have created a copy right directive and it returns a copy right text.  
Please note “app” is an angular application object which has been explained in the previous sections.  
Hide Copy Code  
app.directive(‘companyCopyRight’, function ()  
{  
return  
{  
template: ‘@CopyRight questpond.com ‘  
};  
});  
The above custom directive can be later used in elements as shown in below code.  
Hide Copy Code

### WHAT KIND OF NAMING CONVENTIONS IS USED FOR CUSTOM DIRECTIVES?

For angular custom directive the best practice is to follow camel casing and that also with atleast two letter’s. In camel case naming convention we start with a small letter, followed by a capital letter for every word.  
Some example of camel cases are “loopCounter” , “isValid” and so on.  
So when you register a custom directive it should be with camel case format as shown in the below code “companyCopyRight”.  
Hide Copy Code  
app.directive(‘companyCopyRight’, function ()  
{  
return  
{  
template: ‘@CopyRight questpond.com ‘  
};  
});  
Later when this directive is consumed inside HTML before each capital letter of camel case we need to insert a “-“ as specified in the below code.  
Hide Copy Code  
If you are making a one letter prefix like “copyright” it’s very much possible that tomorrow if HTML team creates a tag with the same name, it will clash with your custom directive. That’s why angular team recommends camel case which inserts a “-“ in between to avoid further collision with future HTML tag’s.

### WHAT ARE THE DIFFERENT CUSTOM DIRECTIVE TYPES IN ANGULARJS?

There are different flavors of Angular directives depending till what level you want to restrict your custom directive.  
In other words do you want your custom directive to be applied only on HTML element or only on an attribute or just to CSS etc.  
So in all there are four different kinds of custom directives:-  
Element directives (E)  
Attribute directives (A)  
CSS class directives (C)  
Comment directives (M)  
Below is a simple custom directive implementation at the element level.  
Hide Copy Code  
myapp.directive(‘userinfo’, function()  
{  
var directive = {};  
directive.restrict = ‘E’;  
directive.template = “User : {{user.firstName}} {{user.lastName}}”;  
return directie;  
});  
The “restrict” property is set to “E” which means that this directive can only be used at element level as shown in the code snippet below.  
Hide Copy Code  
If you try to use it at an attribute level as shown in the below code it will not work.  
Hide Copy Code  
So “E” for element, “A” for attribute, “C” for CSS and “M” for comments.

### WHAT IF I WANT CUSTOM DIRECTIVES TO BE APPLIED ON ELEMENT AS WELL AS ATTRIBUTES ?

Hide Copy Code  
directive.restrict = ‘EA’;

### CAN I SET AN ANGULAR DIRECTIVE TEMPLATE TO A HTML WEB PAGE?

Yes, you can set template to page directly by using “templateUrl” property of the directive as shown in the code snippet below.  
Hide Copy Code  
directive.templateUrl = “/templates/footer.html”;

### EXPLAIN $Q SERVICE, DEFERRED AND PROMISES?

Promises are POST PROCESSING LOGICS which you want to execute after some operation / action is completed. While deferred helps to control how and when those promise logics will execute.  
We can think about promises as “WHAT” we want to fire after an operation is completed while deferred controls “WHEN” and “HOW” those promises will execute.  
For example after an operation is complete you want to a send a mail, log in to log file and so on. So these operations you will define using promise. And these promise logics will be controlled by deferred.  
We are thankful to [http://www.stepbystepschools.net](http://www.stepbystepschools.net/) for the above image.  
So once some action completes deferred gives a signal “Resolve”, “Reject” or “Notify” and depending on what kind of signal is sent the appropriate promise logic chain fires.  
“$q” is the angular service which provides promises and deferred functionality.  
Using promises, deferred and “q” service is a 4 step process:-  
Step 1:- Get the “q” service injected from Angular.  
Step 2 :- Get deferred object from “q” service object.  
Step 3 :- Get Promise object from deferred object.  
Step 4 :- Add logics to the promise object.  
Below is the angular code for the above 4 steps.  
Hide Copy Code  
// Step 1 :- Get the “q” service  
function SomeClass($scope,$q) {  
// Step 2 :- get deferred from “q” service  
var defer = $q.defer();  
// step 3:- get promise from defer  
var promise = defer.promise;  
// step 4 :- add success and failure logics to promise object  
promise.then(function () {  
alert(“Logic1 success”);  
}, function () {  
alert(“Logic 1 failure”);  
});  
promise.then(function () {  
alert(“Logic 2 success”);  
}, function () {  
alert(“Logic 2 failure”);  
});  
}  
So now depending on situations you can signal your promise logics via deferred to either fire the success events or the failure events.  
Hide Copy Code  
// This will execute success logics of promise  
defer.resolve();  
Hide Copy Code  
// This will execute failure logics of promise  
defer.reject();

|  |
| --- |
| 1. **In Angular, why would you want to use the “controller as” syntax vs $scope?**   The $scope object is used to pass data and behaviour to the view. It glues the view and controller together.  Essentially the $scope object performs the following tasks:   1. Pass data from the controller to the view 2. Pass behaviour from the controller to the view 3. Glues the controller and view together 4. The $scope object gets modified when a view changes and a view gets modified when the properties of the $scope object change   We attach properties to a $scope object to pass data and behaviour to the view. Before using the $scope object in the controller, we need to pass it in the controller function as dependencies.  By using the controller as syntax, the code is very clean at the controller and only the alias name is visible on the view.  Here are some steps to use the controller as syntax:   1. Create a controller without $scope object. 2. Assign this to a local variable. I preferred variable name as vm, you can choose any name of your choice. 3. Attach data and behaviour to the vm variable. 4. On the view, give an alias to the controller using the controller as syntax. 5. You can give any name to the alias. I prefer to use vm unless I’m not working with nested controllers.   In creating the controller, there are no direct advantages or disadvantages of using the $scope object approach or the controller as syntax. It is purely a matter of choice, however, using the controller as syntax makes the controller’s JavaScript code more readable and prevents any issues related to this context.  Nested controllers in $scope object approach.  In the controller as syntax, we have more readable code and the parent property can be accessed using the alias name of the parent controller instead of using the $parent syntax.   1. **Explain how have you performed asynchronous calls in C#?**  How To Make Asynchronous Calls Asynchronous calls are made by using delegates. A delegate is an object that wraps a function. Delegates provide a synchronous function and also provide methods for calling the wrapped function asynchronously. Those methods are **BeginInvoke()** and **EndInvoke()**. The parameter lists of these methods vary depending on the signature of the function that the delegate wraps. Note that the Visual Studio .NET IntelliSense feature does not display **BeginInvoke()** and **EndInvoke()**, so you do not see them appear in the function lists as you type.   **BeginInvoke()** is used to initiate the asynchronous call. It has the same parameters as the wrapped function, plus two additional parameters that will be described later in this article. **BeginInvoke()** returns immediately and does not wait for the asynchronous call to complete. **BeginInvoke()** returns an **IAsyncResult** object.  The **EndInvoke()** function is used to retrieve the results of the asynchronous call. It can be called anytime after **BeginInvoke()**. If the asynchronous call has not completed yet, **EndInvoke()** blocks until it completes. The parameters of the **EndInvoke()** function includes the out and ref parameters that the wrapped function has, plus the **IAsyncResult**object that is returned by **BeginInvoke()**.   1. **In Javascript, when would you use prototypal inheritance over classical inheritance? Is classical inheritance ever an appropriate choice?**   Hence we only need two things to make prototypal inheritance work:   1. A way to create a new object (e.g. object literals). 2. A way to extend an existing object (e.g. Object.create).   In contrast classical inheritance is much more complicated. In classical inheritance you have:   1. Classes. 2. Object. 3. Interfaces. 4. Abstract Classes. 5. Final Classes. 6. Virtual Base Classes. 7. Constructors. 8. Destructors.   The point is that prototypal inheritance is easier to understand, easier to implement, and easier to reason about. Prototypal Inheritance is Powerful Prototypal inheritance simply means one object inheriting from another object. There are actually [two ways to implement prototypal inheritance](http://aaditmshah.github.io/why-prototypal-inheritance-matters/#two_methods_of_prototypal_inheritance):   1. Delegation or Differential Inheritance 2. Cloning or Concatenative Inheritance  Prototypal Inheritance is Less Redundant This point is a little more difficult to explain because classical inheritance doesn't necessarily lead to more redundant code. In fact inheritance, whether classical or prototypal, is used to reduce the redundancy in code.  One argument could be that most programming languages with classical inheritance are statically typed and require the user to explicitly declare types (unlike Haskell which has implicit static typing). Hence this leads to more verbose code. Prototypal Inheritance is Dynamic One of the most important advantages of prototypal inheritance is that you can add new properties to prototypes after they are created. This allows you to add new methods to a prototype which will be automatically made available to all the objects which delegate to that prototype.  This is not possible in classical inheritance because once a class is created you can't modify it at runtime. This is probably the single biggest advantage of prototypal inheritance over classical inheritance, and it should have been at the top. However I like saving the best for the end.   1. **What unit testing frameworks do you prefer to use, and what do you consider an acceptable level of unit test coverage (in percentage)?** |
| [NUnit vs. MbUnit vs. MSTest vs. xUnit.net](http://stackoverflow.com/questions/261139/nunit-vs-mbunit-vs-mstest-vs-xunit-net) [xUnit.NET](http://xunit.github.io/). While most of the other testing frameworks mentioned are all pretty much the same, xUnit.NET has taken a pretty unique, modern, and flexible approach to unit testing. It changes terminology, so you no longer define TestFixtures and Tests...you specify Facts and Theories about your code, which integrates better with the concept of what a test is from a TDD/BDD perspective.  xUnit.NET is also EXTREMELY extensible. Its FactAttribute and TraitAttribute attribute classes are not sealed, and provide overridable base methods that give you a lot of control over how the methods those attributes decorate should be executed. While xUnit.NET in its default form allows you to write test classes that are similar to NUnit test fixtures with their test methods, you are not confined to this form of unit testing at all. You are free to extend the framework to support BDD-style Concern/Context/Observation specifications, as depicted [here](http://iridescence.no/post/Extending-xUnit-with-a-Custom-ObservationAttribute-for-BDD-Style-Testing.aspx).  xUnit.NET also supports fit-style testing directly out of the box with its Theory attribute and corresponding data attributes. Fit input data may be loaded from excel, database, or even a custom data source such as a Word document (by extending the base data attribute.) This allows you to capitalize on a single testing platform for both unit tests and integration tests, which can be huge in reducing product dependencies and required training. |
|  |
| 1. **In Javascript, when would you choose to one-way data bind a model?** |
| 1. **How do you unit test an AngularJS filter?**   You can inject $filter and load the filter that you want to test. Then you pass the parameter to be filtered through the filter you have injected and you 'expect' what you needed. Here is an example:  describe('Filter test', function(){  var filter;  beforeEach(function(){  module.apply(moduleName);  inject(function($injector){  filter = $injector.get('$filter')('nameOfTheFilter');  });  });  it('should filter the parameters passed', function(){  expect(filter(parameterToBeFiltered)).toBe(Result);  });  }); |
| 1. **Can you describe a situation where using $scope or $rootscope (or both) is unavoidable even with “controller as”?**   A property assigned with $scope cannot be used outside the controller in which it is defined whereas the a property assigned with $rootScope can be used anywhere.  "$rootScope” is a parent object of all “$scope” angular objects created in a web page.  $scope is created with ng-controller while $rootscope is created with ng-app. |
| 1. **How would you optimize the speed of a slow-running SQL query?**   Slow-running queries can be caused by performance problems related to your network or the computer where SQL Server is running. Slow-running queries can also be caused by problems with your physical database design.  common reasons for slow-running queries and updates:   * Slow network communication. * Inadequate memory in the server computer, or not enough memory available for SQL Server. * Lack of useful statistics * Lack of useful indexes. * Lack of useful indexed views. * Lack of useful data striping. * Lack of useful partitioning. |
| 1. **Explain the 4 principles of object oriented programming.**   **Data Abstraction:** Data Abstraction is a concept in which the internal and superfluous details of the implementation of a logic is hidden from an end user(who is using the program) .A user can use any of the data and method from the class without knowing about how this is created or what is the complexity behind it. In terms of a real world example, when we drive a bike and change the gears we don’t have to care about how internally its working, like how lever is pulled or how chain is set. **Inheritance:** Inheritance is most popular Concept in OOP’s .This provides a developer an advantage called reusability of code. Suppose a class is written having functions with specific logic, then we can derive that class into our newly created class and we don’t have to write the logic again for derived class functions, we can use them as it is. **Data Encapsulation:** Wrapping up of member data and member functions of a class in a single unit is called encapsulation. The visibility of the member functions,data members is set via access modifiers used in class. **Polymorphism:** Poly means many and morphism means changing or alterable. The Concepts Introduces in the form of Many behaviours of an object. **Message Communication:** Message Communication means when an object passes the call to method of class for execution. |
| 1. **How does dependency injection work, and how are you using it in your current project?**   Dependency injection (DI) is a technique for achieving loose coupling between objects and their collaborators, or dependencies. Rather than directly instantiating collaborators, or using static references, the objects a class needs in order to perform its actions are provided to the class in some fashion. Most often, classes will declare their dependencies via their constructor, allowing them to follow the [Explicit Dependencies Principle](http://deviq.com/explicit-dependencies-principle/). This approach is known as “constructor injection”.  ASP.NET Core includes a simple built-in container (represented by the IServiceProvider interface) that supports constructor injection by default, and ASP.NET makes certain services available through DI. ASP.NET’s container refers to the types it manages as services. Throughout the rest of this article, services will refer to types that are managed by ASP.NET Core’s IoC container. You configure the built-in container’s services in the ConfigureServices method in your application’s Startup class. |
| 1. **In AngularJS, how would you use the template of a directive before it is executed? Is this even possible?**   Beware of the use of templateUrl, as “ Because the template loading is asynchronous the compilation/linking is suspended until the template is loaded ” [[source]](http://docs.angularjs.org/api/ng/service/$compile#-templateurl-). As a result, the order of execution will be disrupted. You can remedy this by including the template inlined in the template property instead. |

|  |  |
| --- | --- |
| 1. **In Angular, what is the difference between $routeProvider and $stateProvider?**  $stateProvider $stateProvider allows us to give names for routes. Having a name we can duplicate the route with another name assign different controller, view, well.. we can do whatever we want! It means… with this approach we have different states of the route, thats why its called **$stateProvider**.  **$stateProvider**  **.state("contact", {**  **url: "/contact/",**  **templateUrl: '/app/Aisel/Contact/views/contact.html',**  **controller: 'ContactCtrl'**  **});**  **$routeProvider** solution for a routing. What $routeProvider is doing, is linking specific URL to a controller and assigning a template.  $routeProvider  .when('/contact/', {  templateUrl**:** 'app/views/core/contact/contact.html',  controller**:** 'ContactCtrl'  })  }); | |
| 1. **What is the difference between a controller in ASP.NET MVC and ASP.NET Web API?**   Web API framework is to generate HTTP services that reaches more clients by generating data in raw format, for example, plain XML or JSON string. So, ASP.NETWeb API creates simple HTTP services that renders raw data. On the other hand, ASP.NET MVC framework is used to develop web applications that generates Views as well as data. ASP.NET MVC facilitates in rendering HTML easy. | |
| 1. **How do lambda expressions work in C#?  Explain how anonymous methods and delegates relate to lambda expressions.** | |
| 1. **What is your favorite design pattern?  Give an example of how you last implemented that design pattern.** | |
| 1. **In Angular, what is the difference between $q and $http.post().then()?**  * A $q service that helps you run functions asynchronously, and use their return values (or exceptions) when they are done processing. * $q is integrated with the [$rootScope.Scope](https://docs.angularjs.org/api/ng/type/$rootScope.Scope) Scope model observation mechanism in angular, which means faster propagation of resolution or rejection into your models and avoiding unnecessary browser repaints, which would result in flickering UI. * The $http service is a core Angular service that facilitates communication with the remote HTTP servers via the browser's [XMLHttpRequest](https://developer.mozilla.org/en/xmlhttprequest) object or via [JSONP](http://en.wikipedia.org/wiki/JSONP). * Shortcut method to perform POST request. | |
| 1. **Describe the difference between OLAP and OLTP.**   ***OLTP (On-line Transaction Processing)*** is characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3NF).   - ***OLAP (On-line Analytical Processing)*** is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema).    The following table summarizes the major differences between OLTP and OLAP system design.  **OLTP System - Online Transaction Processing (Operational System)** **OLAP System - Online Analytical Processing (Data Warehouse)**  **Source of data** OLTP: Operational data; OLTPs are the original source of the data. OLAP: Consolidation data; OLAP data comes from the various OLTP Databases  **Purpose of data** OLTP: To control and run fundamental business tasks OLAP: To help with planning, problem solving, and decision support  **What the data** OLTP: Reveals a snapshot of ongoing business processes OLAP: Multi-dimensional views of various kinds of business activities  **Inserts and Updates** OLTP: Short and fast inserts and updates initiated by end users OLAP: Periodic long-running batch jobs refresh the data  **Queries** OLTP: Relatively standardized and simple queries Returning relatively few records OLAP: Often complex queries involving aggregations  **Processing Speed** OLTP: Typically very fast OLAP: Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes  **Space Requirements** OLTP: Can be relatively small if historical data is archived OLAP: Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP  **DatabaseDesign** OLTP: Highly normalized with many tables OLAP: Typically de-normalized with fewer tables; use of star and/or snowflake schemas  **Backup and Recovery** OLTP: Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability OLAP: Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method*source:* | |
| 1. **Have you used Telerik controls before (Kendo UI controls or RadControls for ASP.NET AJAX)?** | |
| 1. **What guidance would you provide to help a team choose between ASP.NET MVC and ASP.NET Web API for a solution?** | |
| 1. **What is a table scan vs a table seek in SQL?**   **Index Scan:** Since a scan touches every row in the table, whether or not it qualifies, the cost is proportional to the total number of rows in the table. Thus, a scan is an efficient strategy if the table is small or if most of the rows qualify for the predicate.  **Index Seek:** Since a seek only touches rows that qualify and pages that contain these qualifying rows, the cost is proportional to the number of qualifying rows and pages rather than to the total number of rows in the table.  Index Scan is nothing but scanning on the data pages from the first page to the last page. If there is an index on a table, and if the query is touching a larger amount of data, which means the query is retrieving more than 50 percent or 90 percent of the data, and then the optimizer would just scan all the data pages to retrieve the data rows. If there is no index, then you might see a Table Scan (Index Scan) in the execution plan.  Index seeks are generally preferred for the highly selective queries. What that means is that the query is just requesting a fewer number of rows or just retrieving the other 10 (some documents says 15 percent) of the rows of the table. | |
| 1. **Can you use dynamic SQL stored procedures with Entity Framework? If so, how?** 2. **Difference between an abstract & virtual method.  He stated that virtual had to be overridden.** 3. **I asked him how to define the routes in a WebAPI project on a per-method basis.  He didn’t know.** 4. **I asked him how to change a synchronous WebAPI method to an asynchronous one.  He didn’t know.** 5. **Madhu even gave him the AWAIT keyword, and even after that he didn’t still state anything about the ‘async’ modifier or wrapping the result type in a Task<>** | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Interview Coding Practices:**

**Convert a Binary Tree to Double Linked list?**

**Fibonacci Series?**

public void FibonacciSeries()

{

int Prev = 1;

int Next = -1;

int Sum = 0;

Console.WriteLine("How many fibonacci numbers you want to print? ");

int numbers = Convert.ToInt16(Console.ReadLine());

for (int i = 0; i < numbers; i++)

{

Sum = Prev + Next;

Console.WriteLine(Sum);

Next = Prev;

Prev = Sum;

}

Console.ReadLine();

}

**Palindrome of a Number?**

**Print Square of a stars ”\*” with text inside?**

namespace PrintRectStars

{

class Helper

{

public static int maxLengthOfString(List<string> array)

{

int maxStringLength = 0;

for (var i = 0; i < array.Count; i++)

{

if (array[i].Length > maxStringLength)

{

maxStringLength = array[i].Length;

}

}

maxStringLength += 4; //To add additional space

return maxStringLength;

}

public static void printAstricInLine(int maxStringLength)

{

for (var index = 0; index < maxStringLength; index++)

{

Console.Write("\*");

}

Console.WriteLine();

}

public static void printAstricAndString(string stringName, int maxStringLength)

{

Console.Write("\* ");

var needtoBeAddedSpace = maxStringLength - stringName.Length;

needtoBeAddedSpace -= 3;

for (var index = 0; index < needtoBeAddedSpace; index++)

{

stringName = stringName + " ";

}

stringName = stringName + "\*";

Console.WriteLine(stringName);

}

}

class Program : Helper

{

static void Main(string[] args)

{

List<string> array = new List<string>() { "Hello", "World", "in", "a", "frame", "My First Name", "My Last Name" }; ;

int maxStringLength = maxLengthOfString(array);

//Print first line

printAstricInLine(maxStringLength);

//Print astrinc and text combination

for (var index = 0; index < array.Count; index++)

{

printAstricAndString(array[index], maxStringLength);

}

//Print last line

printAstricInLine(maxStringLength);

Console.ReadKey();

}

}

}

**Select Sort Algorithm?**

**//Output: SelectSortAlgorithm.AlgSort();**

public class SelectSortAlgorithm

{

public static int[] a = new int[5] { 5, 7, 0, 8, 1 };

public static void AlgSort()

{

Sort();

foreach (int temp in a)

{

Console.WriteLine(temp);

}

Console.ReadLine();

}

public static void Sort()

{

int i, j;

int min, temp;

for (i = 0; i < a.Count() - 1; i++)

{

min = i;

for (j = i + 1; j < a.Count(); j++)

{

if (a[j] < a[min])

{

min = j;

}

}

temp = a[i]; // Store the current value in temp variuable

a[i] = a[min]; // Swap the minimum value to the current position

a[min] = temp; // Swap the current value to the minimum value position

}

}

}

1. Angular JS

*“AngularJS is a JavaScript framework which simplifies binding JavaScript objects with HTML UI elements.”*

1. Promises & Deferred

$q” is the angular service which provides promises and deferred functionality.

Using promises, deferred and “q” service is a 4 step process:-

* Step 1:- Get the “q” service injected from Angular.
* Step 2 :- Get deferred object from “q” service object.
* Step 3 :- Get Promise object from deferred object.
* Step 4 :- Add logics to the promise object.

1. Share data b/w controller
2. using services
3. using $state.go services
4. using stateparams
5. using rootscope

## Explanation of each method:

using $state.go

$state.go('book.name', {Name: 'XYZ'});

// then get parameter out of URL

$state.params.Name;

$stateparam works in a similar way to $state.go, you pass it as object from sender controller and collect in receiver controller using stateparam

using $rootscope

(a) sending data from child to parent controller

$scope.Save(Obj,function(data) {

$scope.$emit('savedata',data);

//pass the data as the second parameter

});

$scope.$on('savedata',function(event,data) {

//receive the data as second parameter

});

(b) sending data from parent to child controller

$scope.SaveDB(Obj,function(data){

$scope.$broadcast('savedata',data);

});

$scope.SaveDB(Obj,function(data){`enter code here`

$rootScope.$broadcast('saveCallback',data);

});

1. What is a Service & Factory? Diff between Factory and Service in AngularJS

Factory” and “Service” are different ways of doing DI (Dependency injection) in angular.

Dependency injection is a process where we inject the dependent objects rather than consumer creating the objects. DI is everywhere in Angular or we can go one step ahead and say Angular cannot work without DI.

1. Digest Cycle:

Digest cycle: - It is a simple loop which updates the model and view.

Watchers :- They are listeners which are attached to expression and angular directives and fire when the model data changes.

Dirty check :- This is a extra digest loop which runs to check any cascading left over updates due to the first digest cycle.

1. How do you call an API from angular JS

$HTTP

1. Given a URL (return type is HTML content), write a program to fetch all the urls in that page. Function should be recursive
2. In Angular JS, write a Service that returns no of milliseconds since 1970?
3. How to center an element in the dashboard?
4. Swap Two Numbers, without Addl variable.
5. Check if the string is unique?
6. C#, MVC, LINQ,SQL Server

**IEnumerable VS IQueryable:**

|  | **IEnumerable** | **IQueryable** |
| --- | --- | --- |
| *Namespace* | System.Collections Namespace | System.Linq Namespace |
| *Derives from* | No base interface | Derives from IEnumerable |
| [*Deferred Execution*](http://synvistech.com/blogs/deferred-execution-vs-lazy-loading-vs-eager-loading-vs-explicitly-loading/) | Supported | Supported |
| [*Lazy Loading*](http://synvistech.com/blogs/deferred-execution-vs-lazy-loading-vs-eager-loading-vs-explicitly-loading/) | Not Supported | Supported |
| *How does it work* | While querying data from database, IEnumerable executes select query on server side, load data in-memory on client side and then filter data. Hence does more work and becomes slow. | While querying data from database, IQueryable executes select query on server side with all filters. Hence does less work and becomes fast. |
| *Suitable for* | LINQ to Object and LINQ to XML queries | LINQ to SQL queries |
| *Custom Query* | Doesn’t support | Supports using CreateQuery and Executemethods |
| *Extension method* *parameter* | Extension methods supported in IEnumerable takes functional objects. | Extension methods supported in IEnumerable takes expression objects, i.e., expression tree. |
| *When to use* | When querying data from in-memory collections like List, Array, etc. | When querying data from out-memory (like remote database, service) collections. |
| *Best Uses* | In-memory traversal | Paging |

1. MVC Page life cycle

**Step 1 Fill route: -** MVC requests are mapped to route tables which in turn specify which controller and action to be invoked. So if the request is the first request the first thing is to fill the route table with routes collection. This filling of route table happens in the global.asax file.

**Step 2 Fetch route: -** Depending on the URL sent “UrlRoutingModule” searches the route table to create “RouteData” object which has the details of which controller and action to invoke.

**Step 3 Request context created: -** The “RouteData” object is used to create the “RequestContext” object.

**Step 4 Controller instance created: -** This request object is sent to “MvcHandler” instance to create the controller class instance. Once the controller class object is created it calls the “Execute” method of the controller class.

**Creating Response object: -** This phase has two steps executing the action and finally sending the response as a result to the view.

1. View data/view bag and differences between them

**Temp data** - Helps to maintain data when you move from one controller to another controller or from one action to another action. In other words when you redirect, tempdata helps to maintain data between those redirects. It internally uses session variables.

**View data** - Helps to maintain data when you move from controller to view.

**View Bag** - It’s a dynamic wrapper around view data. When you use Viewbag type, casting is not required. It uses the dynamic keyword internally.

1. Partial view and its advantages

Partial view is a reusable view (like a user control) which can be embedded inside other view.

1. How do you transfer data between MVC action methods?

RedirectToRoute, Tempdata

1. LINQ
2. Array Programming - find elements in an array that gives total of n.
3. Array Programming - Quick Sort
4. Needs to write code in front of the interviewer (Either in a notepad or PC)
5. Is a String Palindrom
6. Find 2nd highest number in an array
7. Find duplicate in a string array
8. Write simple code for finding the second largest number in a given array.
9. Find the latest number in array
10. Reverse the number
11. How to create Angular.js in .net
12. What is the benefits of using Angular.js in .net
13. C# Datastructure --> GroupA, GroupB, GroupC

GroupA can be child to GroupB

and GroupB can be child to GroupC

how can we write simply Class for Group and data structure for this.

1. Write a helper function that can be called multiple times to return the  input into small chunks by replacing \n with \r\n where the output array is emptied later.

Char[] input = new Char[1000];

Char[] output = new char[10];

1. Problem solving question asked : - Merge two sorted array into one without using extra space.

Did not get the solution neither the brute force approach.

Angular :- Deferred object, Api Calls, injection, share data between controllers, scope.

C# :- Delegate, Async programming.

Sql :- basic sql.

Problem solving

Spiral 2D array printing

X 0 0

0 X 0

X 0 X

3\*3 array

Write a method which takes this array as input and evaluate the winner from the board

The method should be capable of solving N\*N array as well.

  Tic tac toe check winner

When I saked for 3 X 3 he was hardcoding the indexes like 0,1, 2.. asked him whether he can do better if the code needs to be expandable to 4x4.

  Given an int array move all zeroes to one end..

Reverse a linked list

  Sum of triplet in array matching sum:  wanted to sort array first using any algo.. asked to write login if sorted..didn’t solve in code, but explained logic which is half right.

Write a definition to declare an element and define a component to use that.

               He wrote the a sample component which returns the div by taking a local variable.

              Where should we do the AJAX request and why ?

               ComponentDidMount

* 1. **.Net framework and .Net core difference**

.NET Core is a free open source, a general-purpose development platform for developing modern cloud-based software applications on Windows, Linux, and macOS operating systems. It operates across several platforms and has been revamped to make .NET fast, scalable, and modern. .NET Core is one of Microsoft’s big contributions and released under the MIT License. It offers the following features:

* + - * Cross-Platform
      * Open Source
      * High Performance
      * Multiple environments and development mode etc.

.NET is a software framework that is designed and developed by Microsoft. The first version of the .Net framework was 1.0 which came in the year 2002. In easy words, it is a virtual machine for compiling and executing programs written in different languages like C#, VB.Net, etc. It is used to develop form-based applications, web-based applications, and web services. There is a variety of programming languages available on the .Net platform, VB.Net, and C# being the most common ones. It is used to build applications for Windows, mobile, web, etc. It provides a lot of functionalities and supports industry standards.

* 1. .Net core - DI concept
     + **Dependency Injection(DI)** is one of the good design patterns in which a technique, “an object achieves other objects that it depends on, called dependencies”. The “injection” does mean the passing of a dependency “a service” into the client that uses it. The service is made part of the client's state. It actually helps us to create a loosely coupled application.
     + DI is one of the most common practices that assist you to create better maintainable code. The main advantage is the application which is going to be created is loosely coupled and has provided superb maintainability, reusability, and testability as well. It is loosely coupled due to dependency required by the class being injected from the external world rather than made themselves directly win in code.
  2. Services.addtransient, Addscoped - whats the difference.

**Transient**

* Makes an instance each time.
* Never shared.
* Used for lightweight stateless services.

**Singleton**

* Creates only single instance.
* Shared among all components that demand it.

**Scoped**

* Creates an instance once per scope.
* Created on every request to the application.
  1. Middleware components
  2. Security on in .net core.
  3. const and read-only difference
     + Constant variables are declared and initialized at compile time. The value can’t be changed after wards. Read-only variables will be initialized only from the Static constructor of the class. Read only is used only when we want to assign the value at run time.
  4. Linq and Stored Proc - which is better

## A brief comparison of LINQ and Stored Procedure

1. Stored procedures are faster as compared to LINQ query since they have a predictable execution plan and can take the full advantage of SQL features. Hence, when a stored procedure is being executed next time, the database used the cached execution plan to execute that stored procedure.
2. LINQ has full type checking at compile-time and Intellisense support in Visual Studio as compared to a stored procedure. This powerful feature helps you to avoid run-time errors.
3. LINQ allows debugging through .NET debugger as compared to a stored procedure.
4. LINQ also supports various .NET framework features like multithreading as compared to stored procedures.
5. LINQ provides the uniform programming model (means common query syntax) to query the multiple databases while you need to re-write the stored procedure for different databases.
6. A stored procedure is the best way for writing complex queries as compared to LINQ.
7. Deploying a LINQ based application is much easy and simple as compared to stored procedures based. Since in case of stored procedures, you need to provide a SQL script for deployment but in case of LINQ, everything gets compiled into the DLLs. Hence you need to deploy only DLLs.

## Limitation of LINQ over Stored Procedures

1. LINQ query is compiled each time while stored procedures re-used the cached execution plan to execute. Hence, a LINQ query takes more time in execution as compared to stored procedures.
2. LINQ is not good for writing complex queries as compared to stored procedures.
3. LINQ is not a good way for bulk insert and update operations.
4. Performance is degraded if you don't write the LINQ query correctly.
5. If you have done some changes in your query, you have to recompile it and redeploy its DLLs to the server.
   1. Generic and collections – explain
      * Generics are used to make reusable code classes to decrease the code redundancy, increase type safety and performance. Using generics, we can create collection classes. To create generic collection, System.Collections.Generic namespace should be used instead of classes such as ArrayList in the System.Collections namespace. Generics promotes the usage of parameterized types
   2. string and stringbuilder difference
      * StringBuilder is used to represent a mutable string of characters. Mutable means the string which can be changed. So, String objects are immutable, but StringBuilder is the mutable string type. It will not create a new modified instance of the current string object but do the modifications in the existing string object. The complete functionality of StringBuilder is provided by StringBuilder class which is present in System.Text namespace.
      * Need of the StringBuilder: As stated above that the String class objects are immutable which means that if the user modifies any string object it will result into the creation of a new string object. It makes the use of string costly. So, when the user needs the repetitive operations on the string then the need of StringBuilder come into existence. It provides the optimized way to deal with the repetitive and multiple string manipulation operations
   3. EF - what is db context - what it does.

DbContext is an important class in Entity Framework API. It is a bridge between your domain or entity classes and the database.

* The primary class that is responsible for interacting with data as objects DbContext.
* The DbContext APIs is not released as part of the .NET Framework, Entity Framework team distributes EntityFramework.dll through NuGet to be more flexible and frequent with releasing new features to Code First and the DbContext APIs.
* DbContext APIs simplify your application interaction with the database.
* It also reduces the number of methods and properties you need to access commonly used tasks.
* In previous versions of Entity Framework, these tasks were often complicated to discover and code.
  1. Function and SP - difference - in sql

### Function Vs. Stored Procedure in SQL Server

|  |  |
| --- | --- |
| **Function** | **Stored Procedure** |
| It returns single value. | It returns single, multiple values, and even zero. |
| It takes only input values. | It takes both input and output values. |
| It cannot modify database. | It can modify database. |
| It can be called from procedure. | It cannot be called from function. |
| It allows SELECT statements only. | It allows both SELECT and DML statements. |
| Function needs to be compiled and executed every time. | Procedures reuse the execution plan and do not need to be compiled repeatedly. |
| Transaction management is not possible in function. | Transaction management is not possible in stored procedures. |

* 1. What is View, Indexes, Query performance?
* **Views** are virtual tables that hold data from one or more tables. It is stored in the database. A view does not contain any data itself, it is a set of queries that are applied to one or more tables that are stored within the database as an object. Views are used for security purposes in databases. Views restrict the user from viewing certain columns and rows. In other words, using a view we can apply the restriction on accessing specific rows and columns for a specific user. A view can be created using the tables of the same database or different databases. It is used to implement the security mechanism in the SQL Server.

**The uses of views In SQL**

* Views are used to implement the security mechanism in SQL Server. Views are generally used to restrict the user from viewing certain columns and rows. Views display only the data specified in the query, so it shows only the data that is returned by the query defined during the creation of the view. The rest of the data is totally abstract from the end user.

## Types of Indexes in SQL?

 Below are few important Indexes available in SQL,

1. Clustered
2. Non-clustered
3. Unique
4. Column store
5. Index with included columns
6. Full text
7. XML
8. Spatial etc

The Scope of this article is limited to Cluster and Non-Cluster indexes.

**Cluster Index:**

A Cluster Index determined the order of physical data of table hence table can have only one cluster index in SQL. In the Cluster index, data can be reordered or sorted in one way.

## Non-Cluster Index:

The non-Cluster index doesn’t sort physical data like the Cluster index. It means that the sequence of data will not be change and data will be returning on the same sequence in which its inserted. Like Book, Non-Cluster index will be store in one place and data will be store in another place. This will allow creating more than one non-cluster index on the table.

|  |  |
| --- | --- |
| **Cluster Index** | **Non-Cluster Index** |
| Can have only one Cluster Index per table. | Can have more than one Non-Cluster Index per table. |
| Cannot consume extra disk storage as that only sort data row in the table | Can consume more disk storage space as data and index will be store in different places. |
| It is faster than Non-Cluster Index. | It is slower than Cluster index as it needs extra look up to search record. |

* Use GROUP BY, ORDER BY, or DISTINCT in your query only when absolutely necessary.

I’m highlighting some of them below,

1. Use EXISTS instead of IN to check the existence of data.
2. Avoid \* in a SELECT statement. Give the name of the columns which you require.
3. Choose appropriate Data Type. E.g. To store strings use varchar in place of the text data type. Use text data type, whenever you need to store large data (more than 8000 characters).
4. Avoid nchar and nvarchar if possible since both the data types takes just double memory as char and varchar.
5. Avoid NULL in a fixed-length field. In case of requirement of NULL, use variable-length (varchar) field that takes less space for NULL.
6. Avoid Having Clause. Having clause is required if you further wish to filter the result of aggregation.
7. Create Clustered and Non-Clustered Indexes.
8. Keep clustered index small since the fields used in a clustered index may also be used in a non-clustered index.
9. Most selective columns should be placed leftmost in the key of a non-clustered index.
10. Drop unused Indexes.
11. Better to create indexes on columns that have integer values instead of characters. Integer values use less overhead than character values.
12. Use joins instead of sub-queries.
13. Use WHERE expressions to limit the size of result tables that are created with joins.
14. Use TABLOCKX while inserting into a table and TABLOCK while merging.
15. Use WITH (NOLOCK) while querying the data from any table.
16. Use SET NOCOUNT ON and use TRY- CATCH to avoid deadlock condition.
17. Avoid Cursors since cursors are very slow in performance.
18. Use Table variable in place of Temp table. Use of Temp tables required interaction with the TempDb database which is a time-taking task.
19. Use UNION ALL in place of UNION if possible.
20. Use Schema name before SQL objects name.
21. Use Stored Procedure for frequently used data and more complex queries.
22. Keep transaction as small as possible since transaction lock the processing tables data and may result into deadlocks.
23. Avoid prefix “sp” with user-defined stored procedure name because SQL server first searches the user-defined procedure in the master database and after that in the current session database.
24. Avoid use of Non-correlated Scalar Sub Query. Use this query as a separate query instead of part of the main query and store the output in a variable, which can be referred to in the main query or later part of the batch.
25. Avoid Multi-statement Table-Valued Functions (TVFs). Multi-statement TVFs are more costly than the inline TVFs.
    1. Triggers - types

A SQL trigger is a database object which fires when an event occurs in a database. We can execute a SQL query that will "do something" in a database when a change occurs on a database table such as a record is inserted or updated or deleted. For example, a trigger can be set on a record insert in a database table. For example, if you want to increase the count of blogs in the Reports table when a new record is inserted in the Blogs table, we can create a trigger on the Blogs' table on INSERT and update the Reports table by increasing blog count to 1.

## Types of Triggers

There are two types of triggers:

1. DDL Trigger
2. DML Trigger

## DDL Triggers

The DDL triggers are fired in response to DDL (Data Definition Language) command events that start with Create, Alter and Drop, such as Create\_table, Create\_view, drop\_table, Drop\_view and Alter\_table.

## DML Triggers

The DML triggers are fired in response to DML (Data Manipulation Language) command events that start with Insert, Update, and Delete. Like insert\_table, Update\_view and Delete\_table.

## There are two types of DML triggers

### 

### AFTER Triggers

AFTER triggers are executed after the action of an INSERT, UPDATE, or DELETE statement.

### INSTEAD Of Triggers

It will tell the database engine to execute the trigger instead of executing the statement. For example an insert trigger executes when an event occurs instead of the statement that would insert the values in the table .

To find nth highest salary using Sub-Query

SELECT TOP 1 SALARY FROM

( SELECT DISTINCT TOP N SALARY FROM EMPLOYEES ORDER BY SALARY DESC ) RESULT ORDER BY SALARY

To find nth highest salary using CTE

WITH RESULT AS

( SELECT SALARY,

DENSE\_RANK() OVER (ORDER BY SALARY DESC) AS DENSERANK FROM EMPLOYEES )

SELECT TOP 1 SALARY FROM RESULT

WHERE DENSERANK = N

**Angular:**

* 1. Promise and Observable difference

**Promises:**

* Emits a single value
* Cannot be canceled
* Not Lazy

Even after we comment the *then()* method but the service call still be issued to the web API call over the network. In order to get the data from service call we have to handle it using *then()*method.

* Supports async/await
* Easy readable source code when we use *try/catch* & *async/await*
* Promises require the caller to have access to the original function that returned the promise in order to have a retry capability.

By default, Angular built-in HTTP services return Observable. To return Promise as a response to the called method, first, we have to convert the response to Promise using *toPromise()* method.

To support *toPrimise()* method, first we have to import toPromise() from *rxjs* as below,

*import 'rxjs/add/operator/toPromise';*

**Observables:**

* *Emits multiple values over a period of time.*  
  We can think of an observable like a stream which emits multiple values over a period of time and the same call back function is called for each item emitted. So, with an observable, we can use the same API to handle asynchronous data whether the data is emitted a single value or multiple values over a period of time.
* Can be canceled by using *unsubscribe()* method
* *Lazy*  
  Here lazy means the observable will not be called until we subscribe to the observable using the *subscribe()* method. If we comment the subscribe method and then try to call the API method then the service call will not be issued, to issue the API call we have to subscribe to the observable at any cost, that is why we call it as a lazy.
* Supports many powerful operators, like *filter, map, reduce, retry, retryWhen, forEach* and like other operators.
* Can use Reactive Extensions
* An array whose items arrive asynchronously over time

By default, Angular built-in *HTTP* services return Observable. Observable is a more powerful way of handling HTTP asynchronous requests. We can convert the promised call back to Observable also by using *Observable.from promise(HTTP call)* method.

* 1. Two way data binding
* In simple words, two-way data binding is a combination of both Property Binding and Event Binding.

<input [value]='data1' (input)='data1 = $event.target.value'>

* That is, it provides the bi-directional synchronization between the View and the Component.
  1. how to pass parameters between parent/child - answer is input, output, view child, content child
  2. How to pass parameters if there is no relation between parent/child ? answer is create new service, shared data.
  3. routing
  4. DI Concept
* A dependency is an object that can be used (a service). An injection is the passing of a dependency to a dependent object (a client) that would use it. Dependency Injection (DI) is a software design pattern that deals with how components get hold of their dependencies. The AngularJS injector subsystem is in charge of creating components, resolving their dependencies, and providing them to other components as requested.
  1. custom decorators
  2. pipes - what does it mean, how to customize
  3. rxjs - how many methods we can implemet apart from map() methid.
  4. How to call http methods from services.
  5. Interceptor() how it helps.