

The architecture of the.Net framework is based on the following key components

1. **Language** – A variety of languages exists for .net framework. They are VB.net and C#. These can be used to develop web applications.
2. **Library**- The .NET Framework includes a set of standard class libraries. The most common library used for web applications in .net is the Web library. The web library has all the necessary components used to develop.Net web-based applications.
3. **Common Language Runtime** - The Common Language Infrastructure or CLI is a platform. .Net programs are executed on this platform. The CLR is used for performing key activities. Activities include Exception handling and Garbage collection.

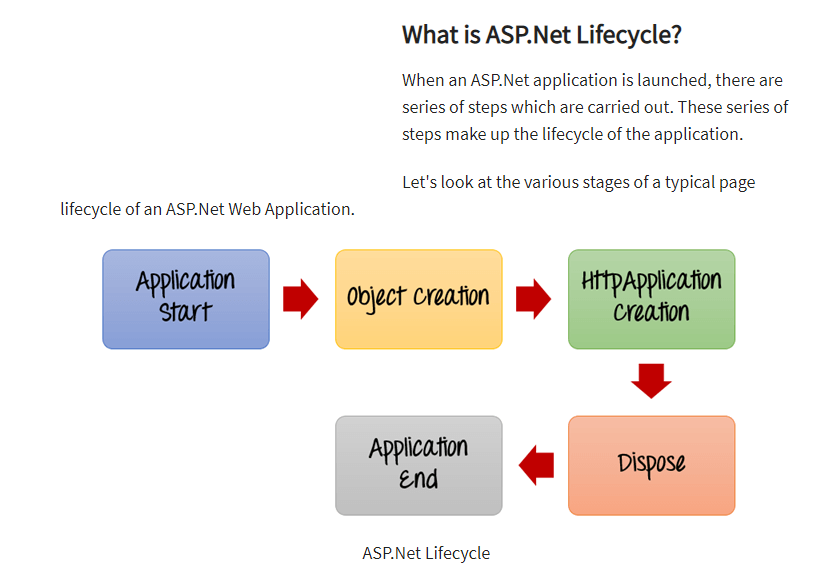
Below are some of the key characteristics of the ASP.Net framework

1. **Code Behind Mode** – This is the concept of separation of design and code. By making this separation, it becomes easier to maintain the ASP.Net application. The general file type of an ASP.Net file is aspx. Assume we have a web page called MyPage.aspx. There will be another file called MyPage.aspx.cs which would denote the code part of the page. So Visual Studio creates separate files for each web page, one for the design part and the other for the code.
2. **State Management**– ASP.Net has the facility to control state management. HTTP is known as a stateless protocol. Let's take an example of a shopping cart application. Now, when a user decides what he wants to buy from the site, he will press the submit button.

The application needs to remember the items the user choose for the purchase. This is known as remembering the state of an application at a current point in time. HTTP is a stateless protocol. When the user goes to the purchase page, HTTP will not store the information on the cart items. Additional coding needs to be done to ensure that the cart items can be carried forward to the purchase page. Such an implementation can become complex at times. But ASP.Net can do state management on your behalf. So ASP.Net can remember the cart items and pass it over to the purchase page.

1. **Caching**– ASP.Net can implement the concept of Caching. This improve's the performance of the application. By caching those pages which are often requested by the user can be stored in a temporary location. These pages can be retrieved faster and better responses can be sent to the user. So caching can significantly improve the performance of an application.

ASP.Net is a development language used for constructing web-based applications. ASP.Net is designed to work with the standard HTTP protocol.



1) **Application Start** - The life cycle of an ASP.NET application starts when a request is made by a user. This request is to the Web server for the ASP.Net Application. This happens when the first user normally goes to the home page for the application for the first time. During this time, there is a method called Application\_start which is executed by the web server. Usually, in this method, all global variables are set to their default values.

2) **Object creation** - The next stage is the creation of the HttpContext, HttpRequest & HttpResponse by the web server. The HttpContext is just the container for the HttpRequest and HttpResponse objects. The HttpRequest object contains information about the current request, including cookies and browser information. The HttpResponse object contains the response that is sent to the client.

3) **HttpApplication creation** - This object is created by the web server. It is this object that is used to process each subsequent request sent to the application. For example, let's assume we have 2 web applications. One is a shopping cart application, and the other is a news website. For each application, we would have 2 HttpApplication objects created. Any further requests to each website would be processed by each HttpApplication respectively.

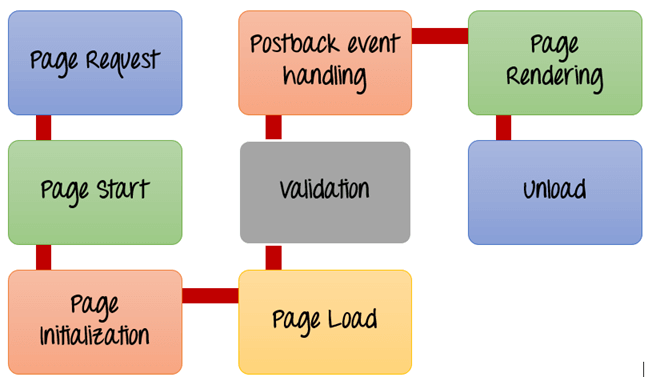
4) **Dispose** - This event is called before the application instance is destroyed. During this time, one can use this method to manually release any unmanaged resources.

5) **Application End** - This is the final part of the application. In this part, the application is finally unloaded from memory.

**What is ASP.Net Page Lifecycle?**

When an ASP.Net page is called, it goes through a particular lifecycle. This is done before the response is sent to the user. There are series of steps which are followed for the processing of an ASP.Net page.

Let's look at the various stages of the lifecycle of an ASP.Net web page.

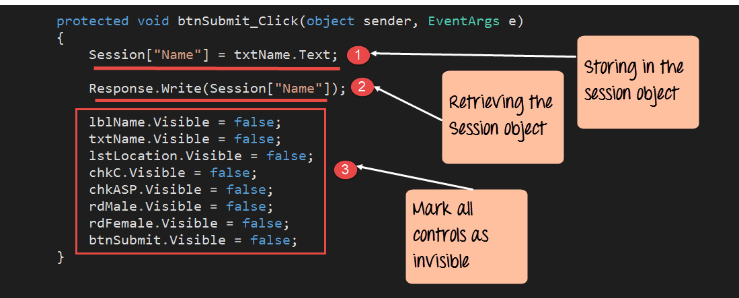
[](https://www.guru99.com/images/asp-net/061516_0807_ASPNetIntro3.png)

ASP.Net Page Lifecycle

1. **Page Request**- This is when the page is first requested from the server. When the page is requested, the server checks if it is requested for the first time. If so, then it needs to compile the page, parse the response and send it across to the user. If it is not the first time the page is requested, the cache is checked to see if the page output exists. If so, that response is sent to the user.
2. **Page Start** – During this time, 2 objects, known as the Request and Response object are created. The Request object is used to hold all the information which was sent when the page was requested. The Response object is used to hold the information which is sent back to the user.
3. **Page Initialization**– During this time, all the controls on a web page is initialized. So if you have any label, textbox or any other controls on the web form, they are all initialized.
4. **Page Load**– This is when the page is actually loaded with all the default values. So if a textbox is supposed to have a default value, that value is loaded during the page load time.
5. **Validation** – Sometimes there can be some validation set on the form. For example, there can be a validation which says that a list box should have a certain set of values. If the condition is false, then there should be an error in loading the page.
6. **Postback event handling**– This event is triggered if the same page is being loaded again. This happens in response to an earlier event. Sometimes there can be a situation that a user clicks on a submit button on the page. In this case, the same page is displayed again. In such a case, the Postback event handler is called.
7. **Page Rendering**– This happens just before all the response information is sent to the user. All the information on the form is saved, and the result is sent to the user as a complete web page.
8. **Unload**– Once the page output is sent to the user, there is no need to keep the ASP.net web form objects in memory. So the unloading process involves removing all unwanted objects from memory.

**Note :** The Response object in ASP.Net is used to send information back to the user. So in our case, we are using the method "Write" of the Response object to write the text "Hello World." The <% and %> markers are used to add ASP.net specific code.

# Session Management



Session["Key"]=value

* Session management is a way in ASP.net to ensure that information is passed over from one page to the other.
* The view state property of a page is used to automatically pass the information of controls from one page to the other.
* The 'Session' object is used to store and retrieve specific values within a web page.

# User Controls

In ASP.Net, it is possible to create re-usable code. The re-usable code can be used in many places without having the need to write the code again.

The re-usable code helps in reducing the amount of time spent by the developer after writing the code. It can be done once and reused at multiple places.

* ASP.Net has the ability to create user controls. User controls are used to have code which is used multiple times in an application. The user control can then be reused across the application.
* The user control needs to be registered on the ASP.Net page before it can be used.
* To use user control across all pages in an application, register it into the web.config file.
* Properties can also be added to a web user control.