ASP.NET Application Life Cycle Events and Global.asax

1. During the application life cycle, the application raises events that you can handle and calls particular methods.
2. To handle application events or methods, we can create a file named **Global.asax** in the root directory of your application.
3. When we create asp.net project in Visual Studio, it will have Global.asax file created by default If not we create a Global.asax file, ASP.NET compiles it into a class derived from the HttpApplication class, and then uses the derived class to represent the application.
4. ASP.NET automatically binds application events to handlers in the Global.asax file using the naming convention ***Application_event***, such as ***Application_Start.***

Application_Disposed: Fired just before an application is destroyed. This is the ideal location for cleaning up previously used resources.

Application_Error: Fired when an unhandled exception is encountered within the application.

Application_Start: Fired when the first instance of the HttpApplication class is created. It allows you to create objects that are accessible by all HttpApplication instances.

Application_End: Fired when the last instance of an HttpApplication class is destroyed. It is fired only once during an application's lifetime.

Session_Start: Fired when a new user visits the application Web site.
Session_End: Fired when a user's session times out, ends, or they leave the application Web site.

ASP.NET Application and Page Life Cycle Overview

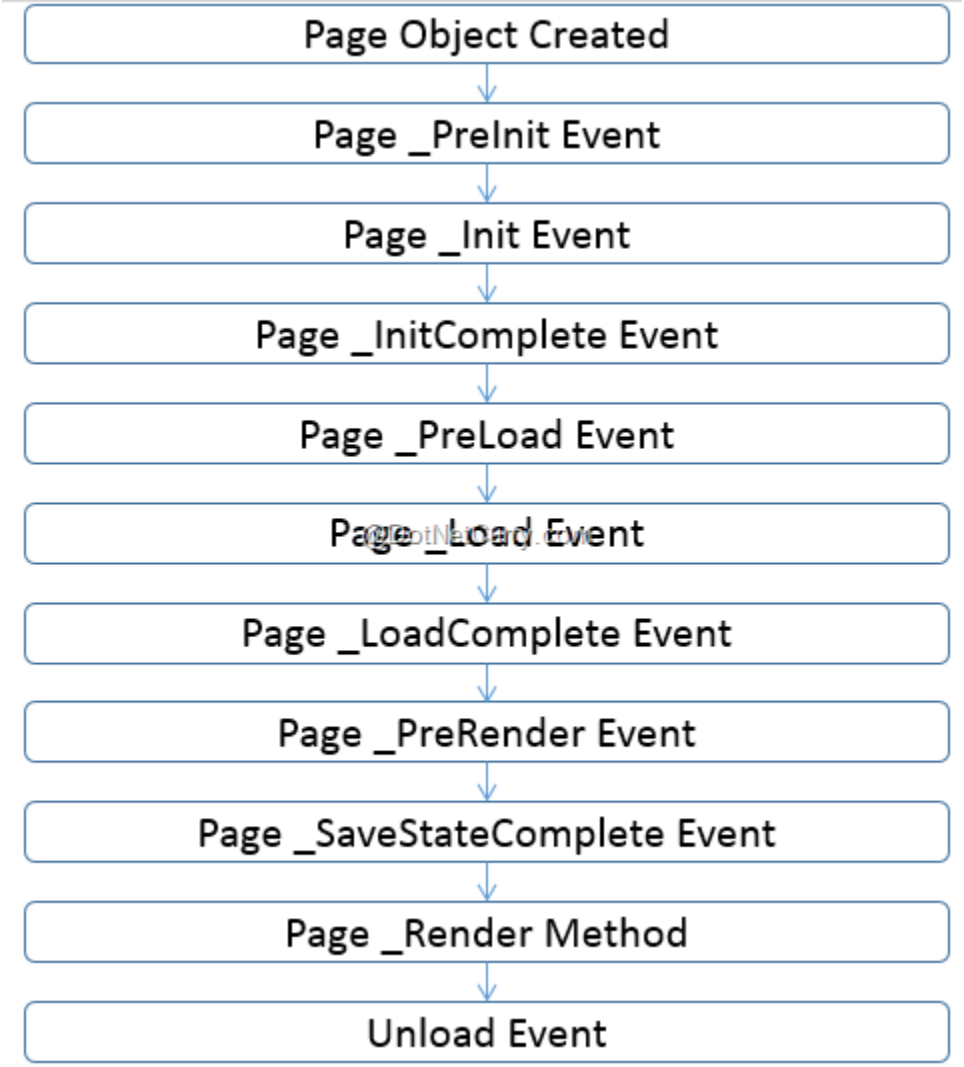
ASP.NET Page goes through a number of events when the page life cycle begins. These events can be used by developers to implement custom logic during the page execution life cycle. ASP.NET Page Life Cycle goes through the number of events in two different scenarios -

- Page Requested for the first time.

- Page requested during Post back (clicking of button etc.)

ASP.NET Page Lifecyle during first request

Let's see how the Page Lifecycle executes when the page is requested first time -



1. **Page_PreInit Event** - This event is the first event which gets executed during the page life cycle. You can use this event to change Master Page and Theme of the page. You can also change the Profile properties under this event.

2. **Page_Init Event** - This event gets fired when all the controls of the page are initialized. Developers can change the properties of a control in this event.

3. **Page_InitComplete** - This event is raised by the Page Object where you can perform different operations to complete the initialization.

4. **Page_PreLoad Event** - This event is used for performing operations on page or controls before load event gets fired.

5. **Page_Load Event** - This event loads all the controls and their children controls. You can use this event for writing database logic or filling the data into Dropdown list control. Developers can perform custom logic against a control, like setting control properties.

6. **Page_LoadComplete** - This event can be used for loading all the other controls.

7. **Page_PreRender** - This event can be used for making final changes to the page properties or control properties.

8. **Page_SaveStateComplete Event** - Before this event, ViewState created for the controls and page is saved with Key-Value pair. It is saved in a HiddenField. Do not use this event for changing the control or page properties as they will be ignored.

9. **Page_Render Method** - This is a page method and not an event. Each server side control has this event which generates the mark-up for rending it on a browser.

10. **Page_Unload Event** - This event unloads all the controls and then the page. Developers can use this method for custom logging or any cleanup operations for controls and page.