[Asp.Net WebForm生命周期的详解](https://www.cnblogs.com/hgmyz/p/7006964.html)



**一．Asp.Net页面生命周期的概念**

当我们在浏览器地址栏中输入网址，回车查看页面时，这时会向服务器端IIS）发送一个request请求，服务器就会判断发送过来的请求页面，当完全识别 TTP页面处理程序类后，ASP.NET运行时将调用处理程序的 ProcessRequest 方法来处理请求、创建页面对象。通常情况下，无需更改此方法的实现，因为它是由 Page 类提供的。接下来被创建页面对象的ProcessRequest方法使页面经历了各个阶段：初始化、加载视图状态信息和回发数据、加载页面的用户代码以及执行回发服务器端事件。之后，页面进入显示模式：收集更新的视图状态，生成 HTML 代码并随后将代码发送到输出控制台。最后，卸载页面，并认为请求处理完毕。其中页面对象ProcessRequest方法 完成的这一系列事件的处理过程就是Asp.Net页面生命周期。

**二、为什么需要了解Asp.Net页面生命周期**

因为了解Asp.Net页面生命周期，可以帮助开发者在生命周期的合适阶段编写程序，以达到预期的效果，另外如果你想自己开发自定义控件，就必须熟悉页面的生命周期，以便正确进行控件的初始化，使用视图状态数据填充控件的属性，以及运行任何控件行为代码。也就是说只有熟悉了从创建到最后卸载等一系列的事件，开发起来才会顺心顺手，不会出现云里雾里的感觉。

**三、生命周期阶段**

1、请求页面：页请求发生在页生命周期开始之前。

2、开始：在开始阶段，将设置页属性，如Request和Response。在此阶段，页还将确定请求是回发请求还是新请求，并设置IsPostBack属性。

3、初始化页面：页面初始化期间，可以使用页中的控件，并将设置每个控件的UniqueID属性。如果当前请求是回发请求，则回发数据尚未加载，并且控件属性值尚未还原为视图状态中的值。

4、加载页面：加载期间，如果当前请求是回发请求，则将使用从视图状态和控件状态恢复的信息加载控件属性。

5、验证：在验证期间，将调用所有验证程序控件的Validate方法，此方法将设置各个验证程序控件和页的IsValid属性。

6、回发事件处理：如果请求是回发请求，则将调用所有事件处理程序。

7、呈现页面：在页面呈现期间，视图状态将被保存到页面，然后页面将调用每个控件，以将其呈现的输出提供给页的Response属性的OutputStream。

8、卸载页面：完全呈现页、将页发送至客户端并准备丢弃时，将调用卸载。此时，将卸载页属性（如Response和Request）并执行清理。

**四、生命周期的事件**

1、PreInit

使用该事件来执行下列操作：

检查 IsPostBack 属性来确定是不是第一次处理该页。

创建或重新创建动态控件。

动态设置主控页。

动态设置 Theme 属性。

读取或设置配置文件属性值。

注意：

如果请求是回发请求，则控件的值尚未从视图状态还原。如果在此阶段设置控件属性，则其值可能会在下一事件中被重写。

2、Init

在所有控件都已初始化且已应用所有外观设置后引发。使用该事件来读取或初始化控件属性。

3、InitComplete

由 Page 对象引发。使用该事件来处理要求先完成所有初始化工作的任务。

4、PreLoad

如果需要在 Load 事件之前对页或控件执行处理，请使用该事件。

在 Page 引发该事件后，它会为自身和所有控件加载视图状态，然后会处理 Request 实例包括的任何回发数据。

5、Load

Page 在 Page 上调用 OnLoad 事件方法，然后以递归方式对每个子控件执行相同操作，如此循环往复，直到加载完本页和所有控件为止。

使用 OnLoad 事件方法来设置控件中的属性并建立数据库连接

6、控件事件

使用这些事件来处理特定控件事件，如 Button 控件的 Click 事件或 TextBox 控件的 TextChanged 事件。

注意：

在回发请求中，如果页包含验证程序控件，请在执行任何处理之前检查 Page 和各个验证控件的 IsValid 属性。

7、LoadComplete

对需要加载页上的所有其他控件的任务使用该事件。

8、PreRender

在该事件发生前：

Page 对象会针对每个控件和页EnsureChildControls。

设置了 DataSourceID 属性的每个数据绑定控件会调用 DataBind 方法。有关更多信息，请参见下面的数据绑定控件的数据绑定事件。

页上的每个控件都会发生 PreRender 事件。使用该事件对页或其控件的内容进行最后更改。

9、SaveStateComplete

在该事件发生前，已针对页和所有控件保存了 ViewState。将忽略此时对页或控件进行的任何更改。

使用该事件执行满足以下条件的任务：要求已经保存了视图状态，但未对控件进行任何更改。

10、Render

这不是事件；在处理的这个阶段，Page 对象会在每个控件上调用此方法。所有 ASP.NET Web 服务器控件都有一个用于写出发送给浏览器的控件标记的 Render 方法。

如果创建自定义控件，通常要重写此方法以输出控件的标记。不过，如果自定义控件只合并标准的 ASP.NET Web 服务器控件，不合并自定义标记，则不需要重写 Render 方法。有关更多信息，请参见开发自定义 ASP.NET 服务器控件。

用户控件（.ascx 文件）自动合并呈现，因此不需要在代码中显式呈现该控件。

11、Unload

该事件首先针对每个控件发生，继而针对该页发生。在控件中，使用该事件对特定控件执行最后清理，如关闭控件特定数据库连接。

对于页自身，使用该事件来执行最后清理工作，如：关闭打开的文件和数据库连接，或完成日志记录或其他请求特定任务。

注意

在卸载阶段，页及其控件已被呈现，因此无法对响应流做进一步更改。如果尝试调用方法（如 Response.Write 方法），则该页将引发异常。

