**What is a Page Directive?**  
  
Basically Page Directives are commands. These commands are used by the compiler when the page is compiled.  
  
**How to use the directives in an ASP.NET page**  
  
It is not difficult to add a directive to an ASP.NET page. It is simple to add directives to an ASP.NET page. You can write directives in the following format:  
  
<%@[Directive][Attributes]%>  
  
See the directive format, it starts with "<%@" and ends with "%>". The best way is to put the directive at the top of your page. But you can put a directive anywhere in a page. One more thing, you can put more than one attribute in a single directive.  
  
Here is the full list of directives:

* @Page
* @Master
* @Control
* @Import
* @Implements
* @Register
* @Assembly
* @MasterType
* @Output Cache
* @PreviousPageType
* @Reference

Let's discuss something about each directive.  
  
**@Page**  
When you want to specify the attributes for an ASP.NET page then you need to use @Page Directive. As you know, an ASP.NET page is a very important part of ASP.NET, so this directive is commonly used in ASP.NET.  
  
Example:  
  
<%@Page Language="C#" AutoEventWIreup="false" CodeFile="Default.aspx.vb" Inherits="\_Default"%>  
  
**@Master**  
Now you have some information about @Page Directives. The @Master Directive is quite similar to the @Page Directive. The only difference is that the @master directive is for Master pages. You need to note that, while using the @Master Directive you define the template page's property. Then any content page can inherit all the properties defined in the Master Page. But there are some properties that are only available in a Master Page.  
Example  
<%@Master Language="C#" AutoEventWIreup="false" CodeFile="MasterPage1.master.cs" Inherits="MasterPage"%>

**@Control**@Control builds ASP.NET user controls. When you use the directive you define the properties to be inherited by the user controls and theses values are assigned to the user controls  
Example:  
<%@Control Language="VB" Explicit="True" CodeFile="WebUserControl.ascx.vb" Inherits="WebUserControl" %>  
  
**@Import**  
As you know you need to define namespaces in your .vb class before using a C# or VB class. So the @Import Directive imports namespaces. This directive supports just a single attribute "namespace" and this attribute takes a string value that specifies the namespace to be imported. One thing you need to note is that the @Import Directive cannot contain more than one attribute/value pair. But you can use multiple lines.  
  
Example:  
  
<%@Import Namespace="System.Data"%>  
  
**@Implements**  
The @Implements Directive gets the ASP.NET pages to implement .Net framework interfaces. This directive only supports a single attribute interface.  
  
Example:  
  
<%@Implements Interface="System.Web.UI.IValidator"%>  
  
**@Register**  
When you create a user control and you drag that user control onto your page then you will see the @Register directive. This directive registers your user control on the page so that the control can be accessed by the page.  
  
Example:  
<%@ Register TagPrefix="MayTag” Namespace="MyName.MyNameSpace" Assembly="MyAssembly"%>

**@Assembly**  
The @Assembly Directive attaches assemblies to the page or an ASP.NET user control thereby all the assembly classes and interfaces are available to the class. This directive supports the two attributes Name and src. The Name attribute defines the assembly name and the src attribute defines the source of the assembly.

Example:  
<%@Assembly Name="MyAssembly"%>  
<%@Assembly src="MYAssembly.vb">  
  
**@MasterType**  
The @MasterType Directive connects a class name to the ASP.NET page for getting strongly typed references or members contained in the specified Master Page. This directive supports the two attributes Typename and virtualpath. Typename sets the name of the derived class from which to get the strongly typed or reference members and virtualpath sets the location of the page from which these are retrieved.

Example:  
  
<%@MasterType VirtualPath="/MasterPage1.master"%>  
  
**@output cache**  
It controls the output caching policies of an ASP.NET page.  
Example:  
<%@ OutputCache Duration ="180" VaryByParam="None"%>

**@Previouspagetype**This directive specifies the page from which any cross-page posting originates.

**@Reference**  
This directive declares that another page or user control should be complied along with the active page or control. This directive supports the single attribute virtualpath. It sets the location of the page or user control from which the active page will be referenced.  
Example:  
<%@Reference VirtualPath="~/MyControl.ascx"%>