ASP.NET - DIRECTIVES

ASP.NET directives are instructions to specify optional settings, such as registering a custom control and page language. These settings describe how the web forms . *aspx* or user controls . *ascx* pages are processed by the .Net framework.

The syntax for declaring a directive is:

```
<%@  directive_name attribute=value  [attribute=value]  %>
```

In this section, we will just introduce the ASP.NET directives and we will use most of these directives throughout the tutorials.

The Application Directive

The Application directive defines application-specific attributes. It is provided at the top of the global.aspx file.

The basic syntax of Application directive is:

```
<%@ Application Language="C#" %>
```

The attributes of the Application directive are:

Attributes	Description
Inherits	The name of the class from which to inherit.
Description	The text description of the application. Parsers and compilers ignore this.
Language	The language used in code blocks.

The Assembly Directive

The Assembly directive links an assembly to the page or the application at parse time. This could appear either in the global.asax file for application-wide linking, in the page file, a user control file for linking to a page or user control.

The basic syntax of Assembly directive is:

```
<%@ Assembly Name ="myassembly" %>
```

The attributes of the Assembly directive are:

Attributes	Description
Name	The name of the assembly to be linked.
Src	The path to the source file to be linked and compiled dynamically.

The Control Directive

The control directive is used with the user controls and appears in the user control . *ascx* files.

The basic syntax of Control directive is:

```
<%@ Control Language="C#"  EnableViewState="false" %>
```

The attributes of the Control directive are:

Attributes	Description
AutoEventWireup	The Boolean value that enables or disables automatic association of events to handlers.
ClassName	The file name for the control.
Debug	The Boolean value that enables or disables compiling with debug symbols.
Description	The text description of the control page, ignored by compiler.
EnableViewState	The Boolean value that indicates whether view state is maintained across page requests.
Explicit	For VB language, tells the compiler to use option explicit mode.
Inherits	The class from which the control page inherits.
Language	The language for code and script.
Src	The filename for the code-behind class.
Strict	For VB language, tells the compiler to use the option strict mode.

The Implements Directive

The Implement directive indicates that the web page, master page or user control page must implement the specified .Net framework interface.

The basic syntax for implements directive is:

```
<%@ Implements  Interface="interface_name" %>
```

The Import Directive

The Import directive imports a namespace into a web page, user control page of application. If the Import directive is specified in the global.asax file, then it is applied to the entire application. If it is in a page of user control page, then it is applied to that page or control.

The basic syntax for import directive is:

```
<%@ namespace="System.Drawing" %>
```

The Master Directive

The Master directive specifies a page file as being the mater page.

The basic syntax of sample MasterPage directive is:

```
<%@ MasterPage Language="C#"  AutoEventWireup="true"  CodeFile="SiteMater.master.cs"
Inherits="SiteMaster"  %>
```

The MasterType Directive

The MasterType directive assigns a class name to the Master property of a page, to make it strongly typed.

The basic syntax of MasterType directive is:

```
<%@ MasterType attribute="value"[attribute="value" ...]  %>
```

The OutputCache Directive

The OutputCache directive controls the output caching policies of a web page or a user control.

The basic syntax of OutputCache directive is:

```
<%@ OutputCache Duration="15" VaryByParam="None"  %>
```

The Page Directive

The Page directive defines the attributes specific to the page file for the page parser and the compiler.

The basic syntax of Page directive is:

```
<%@ Page Language="C#"  AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default"  Trace="true" %>
```

The attributes of the Page directive are:

Attributes	Description
AutoEventWireup	The Boolean value that enables or disables page events that are being automatically bound to methods; for example, Page_Load.
Buffer	The Boolean value that enables or disables HTTP response buffering.
ClassName	The class name for the page.
ClientTarget	The browser for which the server controls should render content.
CodeFile	The name of the code behind file.
Debug	The Boolean value that enables or disables compilation with debug symbols.
Description	The text description of the page, ignored by the parser.
EnableSessionState	It enables, disables, or makes session state read-only.
EnableViewState	The Boolean value that enables or disables view state across page requests.
ErrorPage	URL for redirection if an unhandled page exception occurs.
Inherits	The name of the code behind or other class.
Language	The programming language for code.
Src	The file name of the code behind class.
Trace	It enables or disables tracing.
TraceMode	It indicates how trace messages are displayed, and sorted by time or category.
Transaction	It indicates if transactions are supported.
ValidateRequest	The Boolean value that indicates whether all input data is validated against a hardcoded list of values.

The PreviousPageType Directive

The PreviousPageType directive assigns a class to a page, so that the page is strongly typed.

The basic syntax for a sample PreviousPagetype directive is:

```
<%@ PreviousPageType attribute="value"[attribute="value" ...]   %>
```

The Reference Directive

The Reference directive indicates that another page or user control should be compiled and linked to the current page.

The basic syntax of Reference directive is:

```
<%@ Reference Page ="somepage.aspx" %>
```

The Register Directive

The Register derivative is used for registering the custom server controls and user controls.

The basic syntax of Register directive is:

```
<%@ Register Src="~/footer.ascx" TagName="footer" TagPrefix="Tfooter" %>
```