Introduction to Active Server Pages (ASP)

Active Server Pages is an Active Plataform developed by Microsoft that runs under a client/server architecture .

ASP (Active Server Pages) is a programming environment used to create an run dynamic and interactive HTML documents . The scripts contained in an ASP file are executed from the server. The results of the operation are interpreted and resent to the browser, and displayed in HTML format.The user is totally unaware that a script is being executed since the execution takes place in the background and is not visible to the user.

An ASP document is made up of executables scripts and HTML tags which allow the formatting of results. These files must have the extension **`.asp`**.

ASP Applications

ASP allows the creation of links between different elements so that very dynamic HTML documents can be developed. Among other applications that can be used within the HTML context, it is possible to:

- Add information to a database or gather information from an HTML form.
- Customize the contents of a page according to the profile of a particular user.
- Use HTML functions depending on the user's browser.
- Manage cookies.

Using ASP Directives

An ASP page contains   ASP directives that define how scripts are executed. These scripts can be written using different languages: JScript, Javascript, VBScript, etc.   ASP pages can also contain a diversity of different languages.

The advantage of using ASP is that it can be used to specify a set of directives that are not linked to the particular script language used. These directives are related to   output and interpretation directives.

ASP Syntax and Keywords

The ASP output directive uses   the following syntax `<%= expression %>`; and the ASP interpretation directive uses the following syntax, `<%@ keyword %>`.

The interpretation directive sends the Web Server the ASP information necessary to handle the   `.asp` file.

The interpretation directive uses the following keywords :

LANGUAGE is used to define the scripting language used in the Web page.

CODEPAGE is used to define the code page (character codes) of the page.

LCID is used to specifiy the regional parameters of the   page.

TRANSACTION is used to indicate that the page will be executed in a transaction context.

ENABLESESSIONSTATE is used to indicate if an ASP page uses the session status.

More than one keyword can be included in one directive.

For example:

`<%@ LANGUAGE=JScript CODEPAGE=932 %>`

CODEPAGE

The system uses the code page to translate symbols (letters, punctuation marks) to numbers.   Each code page corresponds to a set of characters, which in turn correspond to a specific language (for example, 932 refers to the Japanese characters).

If no particular specification is indicated under the   **CODEPAGE** directive, ASP uses the normal code **UNICODE** (fixed width, 16 bits). If a code page is specified, ASP uses it to convert the characters using the normal universal   **UNICODE**.

Example:

```
<%@ CODEPAGE= 932 %>
```

You can also use different code pages in the same document, during the same session. The **SESSION** directive is then used.

If you temporarily change to a different code page, you must use the directive   **SESSION** directive again   so that **CODEPAGE** returns to its initial value.

Example:

```
<%@ SESSION.CODEPAGE= 950 %>
```

ENABLESESSIONSTATE

This directive is used to define the session status Defining a   (**SESSION**) in a script is used to prevent the user from browsing a particular Web page by means of   COOKIES. By default, the **SESSION** directive is activated in ASP scripts.

Preventing a session however, has the disadvantage of slowing down ASP directives. Therefore it is best to define and ASP page without a session.

The **ENABLESESSIONSTATE** directive is thus used until the user finds a different directive.

The **ENABLESESSIONSTATE** directive must be indicated on the first line of the ASP file and precedes all other scripts. If the   **ENABLESESSIONSTATE** is omitted, the regular session is activated by default.

For example:

```
<%@ EnableSessionState=False %>
```

LCID

The **LCID** directive is used to define the regional parameters used by ASP scripts.

The regional parameters regroups information according to the user's working environment: language, date and hour formatting, etc.

By default, an ASP script uses the regional parameters used by the Web server.

The **LCID** directive is therefore used to specify a regional parameters identifier for an ASP page .

For example:

`<%@ LCID =  1041 %>` (used to identify the Japanese regional parameters)

You can also use different regional parameter definitions in a document during the same session. In this case, the **SESSION** directive is used.

If you temporarily change to different regional parameters, you must use the directive   **SESSION** directive again   so that **LCID** returns to its initial value.

`<% Session.LCID = 1036 %>` (used to identify the French regional parameters.)

TRANSACTION

ASP is often used to develop a transaction page. When a **TRANSACTION** directive is placed on an ASP page, the   script objects and commands used in the page are executed within the same transaction context.

TRANSACTION is executed through the **Transaction Server** that manages the set of activities related to the transaction which also determines if the transaction was successful (validated) or ended (canceled).

TRANSACTION must appear on the page's first line otherwise an error is generated. The transaction ends when the script is executed. The **TRANSACTION** directive   must be included in each page that requires managing transactions.

This directive uses the **VALUE** argument which may correspond to one of the following:

Value	Explanation
Requires_New	Starts a new transaction
Required	Starts a new transaction
Supported	Does not start a transaction
Not_Supported	Does not start a transaction

Example :

```
<%@ TRANSACTION = required %>
```

LANGUAGE

An ASP page is made up of ASP directives that indicate how scripts are executed. Several languages can be used to write scripts: JScript, Javascript, VBScript, etc.   ASP pages can also contain a diversity of different scripts written in different languages.

The **LANGUAGE** directive is used to specify the language of the script used in the ASP page.

If no language is specified, the ASP page uses the default language used by the Web server (mostly VBScript).

Example:

```
<%@ LANGUAGE= JScript%>
```

You can also use several language definitions during the same session. The **SESSION** directive is then used.

If you temporarily change to a different script language, you must use the   **SESSION** directive again   so that **LANGUAGE** returns to its initial value.

```
<% Session.LANGUAGE = VBSCRIPT%>
```

In ASP development, a **SESSION** refers to a visit session or a particular directive inside an ASP script.