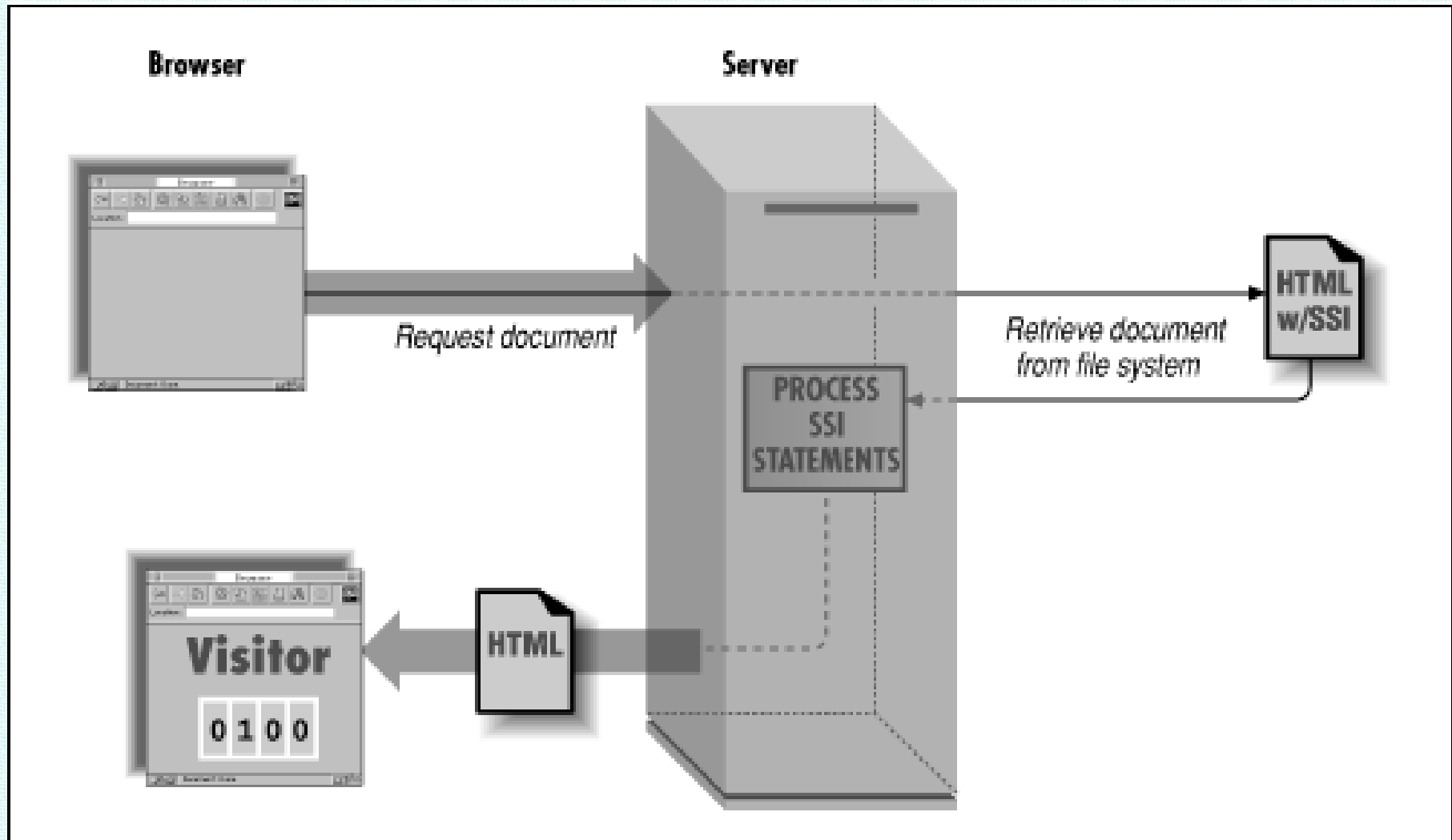


Lesson #7: Server Side Includes (SSI)

What is SSI?

- Server Side Includes (SSI) is an easy server-side scripting language used almost exclusively for the web. As its name implies, its primary use is including the contents of a file into another, via a web server.
- SSI is primarily used to "paste" the contents of one or more files into another.

What is SSI?



Advantages of SSI

- Server Side Includes are useful for including a common piece of code throughout a site, such as a navigation menu.
- You can get many pages to include the information from a single file so that you could, for example, include a standard footer on each page with your copyright information. You could then update all the pages by just changing this one page.

Uses of SSI

- You can get your SSI to execute a CGI script on your server. This allows you to have text counters, advanced advertisement rotations, random text and images and several other extras on your web pages. This is one of the best uses of SSI.

A digital counter displaying the number 9354390. The digits are white on a black background, with a slight shadow effect, giving it a three-dimensional appearance. The counter is enclosed in a white rectangular border.

SSI Needs

- SSI tags are parsed server-side, not by the user's browser. This means that SSI must be set up on your server for them to work. You are unlikely to have SSI enabled if you do not have a CGI-BIN but if you do it is very likely that the server administrator will have enabled SSI.

Testing SSI

- To test whether SSI is set up on your server place the following code in an HTML page:

```
<!--#echo var="DATE_LOCAL"-->
```

Recognizing SSI

- In order for a web server in a default configuration to recognize an SSI-enabled HTML file and therefore carry out these instructions, the file must end with the .shtml extension. SSI files can also end with .shtm but this depends on the server's ability to recognize the extension. It is possible to configure a web server to recognize any file with the .html file extension for server side include processing.

Including a file

- This is one of the most simple things you can do using SSI. Create a text file with a little bit of text in it and save it. Upload it to your server. Then create an HTML file and add the following:

```
<!--#include virtual="myfile.txt" -->
```

- The way in which this works is that the file is referred to relative to your accounts root.

Scripts

- One major use of SSI is to include the output from a script in your page.
- The script is called via a standard SSI call (like for any other file) with a slight change (although 'include virtual' usually also works and preferred by many hosts):

<!--#exec cgi="/cgi-bin/clicktrade.cgi" -->

- SSI can be used with many other scripts, for example to display a textual visitor count on your page or to do an image or text rotation.

Preset Commands

- `<!--#echo var="DATE_LOCAL"-->`: The server local time
- `<!--#echo var="DATE_GMT"-->`: The server time according to GMT.
- You may configure the way time is displayed before using the previous commands:

Ex: `<!--#config timefmt="%A, %d %B %Y"-->`

Time Formats

Code	Function	Examples
%A	Full weekday name	Monday, Wednesday
%a	Abbreviated weekday name	Mon, Wed
%B	Full month name	January, December
%b	Abbreviated month name	Jan, Dec
%d	Number of day in month	03, 1500 01/font>
%e	Number of day in month	3, 15
%H	Hour (1-24)	05, 22
%I	Hour (under 12)	03, 11
%j	Day number in year	12, 245
%m	Number of month	1, 12
%M	Minute	12, 59
%p	AM or PM	AM, PM
%n	New Line	
%S	Second	12, 57
%t	Tab	
%y	2 Digit Year	99, 03
%Y	4 Digit Year	1999, 2003
%Z	Time Zone	EST, GMT

Other Preset Commands

- Document Name (the file's name):

```
<!--#echo var="DOCUMENT_NAME"-->
```

- Document's URI (the part after the domain name)

```
<!--#echo var="DOCUMENT_URI"-->
```

- Referring Page:

```
<!--#echo var="HTTP_REFERER"-->
```

- Users IP Address:

```
<!--#echo var="REMOTE_ADDR"-->
```

Directives

- You may have already noticed that SSI has a simple syntax:
- `<!--#directive parameter=value parameter=value -->`.
- Directives are placed in HTML comments so that if SSI isn't enabled, users won't see the SSI directives on the page, unless they look at its source.
- In the next slides, we will explore the most important directives.

#include

- This is probably the most used SSI directive, allowing the content of one document to be included in another. The file or virtual parameters specify the file (HTML page, text file, script, etc) to be included.
- The file parameter defines the included file as relative to the document path.
- The virtual parameter defines the included file as relative to the document root.

```
<!--#include virtual="header.html"-->
```

#exec

- This directive executes a program, script, or shell command on the server.
- The cmd parameter specifies a server-side command.
- The cgi parameter specifies the path to a CGI script. The PATH_INFO and QUERY_STRING of the current SSI script will be passed to the CGI script.

```
<!--#exec cgi="/cgi-bin/pgm1.cgi"-->
```

```
<!--#exec cmd="ls -l"-->
```

#echo

- This directive displays the contents of a specified HTTP environment variable. Variables include HTTP_USER_AGENT, LAST_MODIFIED, and HTTP_ACCEPT.

```
<!--#echo var="REMOTE_ADDR" -->
```

#config

- This directive configures the display formats for the date, time, file size, and error message (returned when an SSI command fails).

```
<!--#config timefmt="%y %m %d" -->
```

```
<!--#config sizefmt="bytes" -->
```

```
<!--#config errmsg="SSI command failed!" -->
```

#flastmod / #fsize

- These directives display the date when the specified document was last modified, or the specified document's size. The file or virtual parameters specify the document to use. The file parameter defines the document as relative to the document path; the virtual parameter defines the document as relative to the document root.

```
<!--#flastmod virtual="index.html"-->
```

```
<!--#fsize file="script.pl"-->
```

#printenv

- This directive outputs a list of all variables and their values, including environmental and user-defined variables. It has no attributes.

<!--#printenv -->

Conditional Directives

- This lets SSI be a tiny programming language of sorts. It provides an `#if`, `#elif`, `#else`, `#endif` structure for building conditional statements. This allows you to effectively generate multiple logical pages out of one actual page.

```
<!--#if expr="`${HTTP_USER_AGENT} = /MSIE [4-9]/" -->
```

You are using IE 4 or above


```
<!--#elif expr="`${HTTP_USER_AGENT} = /MozillaV[4-9]/" -->
```

You are using Netscape 4 or above


```
<!--#else -->
```

You are using something other than IE 4+ or NS 4+


```
<!--#endif -->
```

Conclusion

- SSI is certainly not a replacement for CGI, or other technologies used for generating dynamic web pages. But it is a great way to add small amounts of dynamic content to pages, without doing a lot of extra work.
- Not all servers permit SSIs. Most Apache servers do. Check with your host first. There may be additional configurations needed to run them.
- One problem with SSIs from the hosting point of view is they slow down server processing. Another is that they create certain security concerns.

End of lesson