

Printing Web Pages

- Meghna Vora and Kasturi Reddy

How to make your Webpage Printable?

This question of "How to print Webpages?" must have intrigued almost all of us at some point of time. This article aims at solving your problem in a jiffy. The answer to this question lies in the form of the simple algorithm given below.

Algorithm for printing webpages

There are two ways of printing webpages.

1. This method directly specifies the URL of the page to be printed.
2. This method uses the index approach where all the data is first stored in a temporary buffer and then printed.

The second method is more reliable as there are no issues of session variables.

Here is the algorithm for the second method.

This algorithm is very simple to implement as well as understand.

1. First of all this method states that the file to be printed should traverse the path via the index, so that there are no issues about accessing "Session Variables".
2. Secondly the data to be printed should be demarcated by <startprint> and <stopprint> tags. Doing this will print data from only in-between the tags. These tags should be included in the template file.
3. The algorithm also provides an unique feature of allowing the user to print, without including CSS or images, and all that is needed to do this is just setting the flags corresponding to images and css respectively.
4. Now that the above tasks have been completed, the next steps consists of specifying the filename, that has to be read in the main php file.
5. Once the code gets the filename, it fetches all the contents of the file into a temporary buffer. It then checks whether to include CSS, images and the length of the file.
6. The final step is to echo all the data which is fetched onto the screen. Whatever is echoed on the screen, in exactly the same manner the file is printed.

How to detect user inactivity? Has this question been troubling you?

Well, php provides an efficient solution to this. This logic is applicable where all the pages traverse through the index i.e. all the pages are called through the index. If a user is not calling the index within the specified time period then it will be treated as the user is dormant or inactive and will be forcefully logged off.

The solution involves just implementing following algorithm:

- Include a file named auto Logout in the index.
- In the auto Logout file check for session userID.
- If session userID is not found, which indicates that first time the index page is called , that's when u define another session variable named as last click.
- In this variable store the current time out.
- Also we need to define one more variable that is the time out. This variable holds the time period in seconds of user inactivity after which the user will be forced to re-login.
- Now check whether the current time difference timeout is greater than the last click time.
- If it is greater than last click time, it indicates that timeout has already occurred and now the user should be forced to re-login if he wants to continue further.
- Now forcefully destroy the session and redirect to the login page.

(This algorithm is suitable for implementation only where events such as key press, mouse click and scroll are not considered as active.)