How to keep a copy of a web page that your server delivered to the browser

In other words how to store the Source code delivered

If the file is an HTML page with no server derived code, no database connection, and no PHP (or other server language) derived code. In other words a completely static code, the just save the page code and go no further.

But .. the moment the server has “done something” to the code you deliver to the browser then that will not work .. and this is where the technique described here comes into its own.

We will use two PHP constructs

1 Output buffer
2 Making a file in a specified place, putting stuff in it, closing the file

Output buffer
Let’s start with Output Buffer. (Buffer means a bit of memory)
What does it do?

In very basic terms it’s a construct that says to the file that will deliver the code to the browser – I want to you take all of this stuff to be delivered and put it into a bit of temporary memory over here.

Then at the bottom of the page when everything has been delivered to the browser including all of the html tags (that means everything between <html> and </html> we clear out that bit of temporary memory

In between we use the stuff that’s been put into temporary memory.

At the start we start the ball rolling with the code
<?php
ob_start();

At end end we clear out the buffer with

ob_end_flush();
?>

OK that’s the top and bottom of it .. so what do we do in the middle to make this work?

Start the Output Buffer before anything is sent to the browser. Once the buffer starts all the contents that will be displayed to the browser will be stored in the buffer. See example below

That’s ob_start()

From this point forward everything that will be sent to browser will be stored in the buffer

So just carry on to construct the entire file output including the html tags

Once everything is completed we need to collect together all of the stored information in the buffer and do something with it.

We do that like this .. Collect buffer content and lets give it a variable name $data Like so …

$data = ob_get_contents(); // get all content and store it in a variable

So all content that will be send to the browser between ob_get_contents(); and the ob_start(); will now be in $data.

So in summary

ob_start() // start the buffer process

.. make your browser output

$data = ob_get_contents() // this is browser output

.. save this to a file

ob_end_flush() // end the buffer process

Making and storing a file
At this point we need to save it into a file. To do that we need to decide what the file name and path will be.

In summary

Make an empty file
Put $data into it
Close the file

PS do not forget to ensure that the directory has permissions to permit making a file
Making and storing a file

At this point we need to save it into a file and we can do that by dealing with PHP File Handling functions.

First create an empty file and decide what the file name and path will be and store it in a variable. Make a file name that is useful for you. You can use time() then you could tie this up with the logs. Just select a file name to suit your purpose.

```
$ourFileName = "/home/username/yourStorageFolder/fileName". time()
.".html";
```

Since we want everything what the viewer sees will be stored in the server we make sure that the filename does not already exist. If you skip checking if the filename exist, the new file will overwrites the old file and then we will lose one and we don't want that.

```
if(file_exists($ourFileName))
{
```

If the filename did exist then we rename it with the unix time stamp on the end and the new file is created. Having a time stamp at the end of the filename will tell us when it was made plus it can produce a trail of exactly what happened and what the viewer saw.

```
home/username/yourStorageFolder/fileName". time()

$newFileName = " home/username/yourStorageFolder/renamed File

rename($ourFileName, $newFileName);
}
```

If the file does not actually exist in the specified place on the sever (up to here we have only given it a name not created it) we now create the file, open it and specify a mode you require using fopen() function. The first parameter of this function is the name of the file to be opened and the second parameter is the mode (that determines that the file is for reading, writing or appending, and a few more).

In our case we require writing so we use "w".
Description of "w" = Open for writing only. Create a new file if it doesn't exist.

```
$ourFileHandle = fopen($ourFileName, 'w');
```

Now we write to it - because we have set the mode to writing.

```
fwrite($ourFileHandle, $data);
```

Next we close the file and provide a return value of true on success or false on failure.

```
fclose($ourFileHandle);
```

PS do not forget to ensure that the directory has permissions to permit making a file

Here is a code example:

```
<?php
ob_start();        // start the buffer process


                   // browser output
                   // browser output
                   // browser output

$data = ob_get_contents();        // collect browser output


$fileName = "/home/example".$uniqueID.".html";        // create an empty file


if(file_exists($fileName))                                    // if file exist
{
        $newFileName = "/home/example".$uniqueID."." .time() . ".html";
        rename($fileName, $newFileName);                      // rename the file
}


if($uniqueID > 0)                                             // check if uniqueID > 0
{
        $fileHandle = fopen($fileName, 'w');                  // open file ready for writing
        fwrite($fileHandle, $data);                           // creates the file
        fclose($fileHandle);                                  // close the file
}

ob_end_flush();                                               // end the buffer process
?>
```