Tutorial written by BetaWar Finished on 06/05/2008

Pagination, what is it and why would I want to use it?

Pagination is simply a way to split content of web pages up into multiple pages. People generally want to use it because it is easy enough to make and customize as well as the nice fact that it saves on server strain and load time of pages. If that isn’t enough to make you want to use pagination on your pages, this should help. The whole pagination script (including our example) can be written in under 59 lines of code!

Have I seen pagination working around the web?

The simple answer to this question is ‘yes’, If you are reading this tutorial on a forum then you have seen pagination in almost every topic and page the forum has (the little numbers at the bottom of the screen that look something like this:
Page 1 2 3 4 5 … 190
For a look at pagination in action feel free to go here (towards the bottom of the page):
http://www.dreamincode.net/forums/showforum93.htm

Onto the tutorial

For the purpose of this tutorial we will say that we have a database called DATABASE that has a table called images in which there are a number of image urls. The whole database looks like this:

```
DATABASE
        Images
                url (varchar)
                image_id (int)
```

Here is a look at what you will be creating:

```
<div style="float: right;">
<?php
$dbh=mysql_connect ("HOST", "USERNAME", "PASSWORD") or die ('Database connection failed because: ' . mysql_error());
mysql_select_db ("DATABASE");
$page = $_GET['page'];
if(!$page || $page == 0 || !is_numeric($page)){
  $page = 1;
}
$limit = 5;
class paginate{
  function init($limit, $page){
    $query1 = mysql_query("SELECT * FROM images");
    $total_pages = ceil(mysql_num_rows($query1)/$limit);
    $prev = '<a href="?page=' . ($page-1) . '">Prev</a>';
    $next = '<a href="?page=' . ($page+1) . '">Next</a>';
    if($page <= 1){
      $page = 1;
      $prev = '';
    }
    elseif($page >= $total_pages){
      $page = $total_pages;
      $next = '';
    }
    if($page > 2){
      $prev = '<a href="?page=1">First</a> ' . $prev;
    }
    if($page < ($total_pages - 1)){
      $next = $next . ' <a href="?page=' . $total_pages . '">Last</a>';
    }
    $start = ($limit*($page-1));
    $query2 = mysql_query("SELECT * FROM images LIMIT $start, $limit") or die("Error
In Query " . mysql_error());
    $times_done=0;
    while($row = mysql_fetch_array($query2)) {
      if($times_done=='5'){
        $content .= '<br><br>';
        $times_done=1;
      }
      else{
        $times_done++;
      }
      $content .= '<a href="' . $row['src'] . '"> <img width="100" height="100" src="' .
$row['src'] . '"> </a>';
    }
    $content .= '<br><hr>' . $prev . ' Page ' . $page . '/' . $total_pages . ' ' . $next;
    return $content;
  }
}
$class = &new paginate;
echo $class->init($limit, $page);
mysql_close($dbh);
?>
```

</div>
To see the above code in action follow the link belowhttp://www.reigninggames.com/images/halo2.php

Now, we have the basic code set out, but how does it work?

Here is what each line of code does for the final application:

<div style=”float: right;”>	Simple enough, it isn’t anything more than a bit of HTML. We are creating a div that floats right
<?php $dbh=mysql_connect ("HOST", "USERNAME", "PASSWORD") or die ('Database connection failed	This is the start of the php code and a connection to the mysql database
mysql_select_db ("DATABASE");	Here we select a database for future mysql queries
$page = $_GET['page'];	Set $page as the number that passed through the URL variable page

if(!$page || $page==0 ||
!is_numeric($page)){
  $page = 1;
}
	Check the $page variable to make sure it is something that the application can use. So, if $page is not set, is equal to 0 or is not a number it is set to 1

$limit = 5;	Sets a limit for the number of results displayed per page
class paginate{	Starts up the class paginate
function init(**$limit**, **$page**){	Init function
$query1 = mysql_query("SELECT * FROM images");	Query the database to find the number of total rows
$total_pages = ceil(mysql_num_rows($query1)/$limit);	Set the number of total pages
$prev = '<a href="?page=' . ($page-1) . '">Prev</a>'; $next = '<a href="?page=' . ($page+1) . '">Next</a>';	Sets the initial previous and next pages to go forward and backwards by 1 page
if($page <= 1){ $page = 1; $prev = '';	Checks if the $page is less than or equal to 1 If it is the $page is set to 1 and $prev is set to nothing

```
  elseif($page >= $total_pages){
    $page = $total_pages;
    $next = '';
  }
```

Otherwise check if $page is greater than or equal to the number of total pages. If it is set $page to the total number of pages and set $next to nothing

```
  if($page > 2){
    $prev = '<a href="?page=1">First</a> ' . $prev;
  }
```

Check if $page is greater than 2
If it is set $prev equal to a First page link followed by whatever $prev was before

```
  if($page < ($total_pages - 1)){
    $next = $next . ' <a href="?page=' . $total_pages . '">Last</a>';
  }
```

Check if $page is less than 1 less than the total number of pages, and if it is, set $next equal to itself followed by a Last page link

```
  $start = ($limit*($page-1));
```

Set where to start collecting results depending on the current page and limit

```
  $query2 = mysql_query("SELECT * FROM images LIMIT $start, $limit") or die("Error In Query " . mysql_error());
```

Query the database and select all results from the table starting at entry number $start and ending $limit after $start

```
  $times_done=0;
  while($row = mysql_fetch_array($query2)) {
    if($times_done=='5'){
      $content .= '<br><br>';
      $times_done=1;
    }
```

Set $times_done equal to 0
While there is a result available from the query…
If $times_done is 5 add a new line to $content and set $times_done back to 1

```
    else{
      $times_done++;
    }
```

Otherwise set $times_done to 1 greater than it is currently

```
    $content .= '<a href="' . $row['src'] . '"> <img width="100" height="100" src="' . $row['src'] . '"> </a>';
  }
```

Add an image thumbnail and link to the image to $content

$content .= ' <hr>' . $prev . ' Page ' . $page . '/' . $total_pages . ' ' . $next;	Enter an extra line break and horizontal line to $content then add the page navigation links
return $content;	Return the content
} }	
$class = &new paginate;	Set $class as a pointer to the class paginate

echo $class->init($limit, $page);
mysql_close($dbh);
	Output the returned value from paginate’s function init, and close the database connection

?>	End the php code
</div>	Close the div

Okay, now that I have finished understanding the code, how do I implement it?

Implementing the code can be one of the hardest, or one of the easiest things to do, depending on what you personally need to change for it to fit what you are trying to accomplish. If you are just going for a quick and ugly thing like what we created in this tutorial then all you really need to do is change some of the variables to suit your needs. If that was the case, just change the red, and green text to what fits your needs and everything will be fine.

However, if you want to do something more complex it will take more work. You will still have to change the red and green text, but you will probably also have to change some of the normal black text as well. For instance, this tutorial was built so that if $limit is greater than 5 there will be multiple rows of images per page. So those things would need to go.

The basics behind implementing pagination on a website are to have a database that you can connect to and then have data in it that can be queried. Then you need to query for the data and use it to add to content as what we did with the $row[‘src’] variable. The loop will assign all the variables to a part of $row, making it an array; you just call for the variables from the array. The array key names are easy enough to figure out as they are exactly the same as the column names in your database. Meaning that, in addition to the $row[‘src’] variable we could have called to $row[‘image_id’] and received a number. Realize though that if you change the SELECT * FROM to something like SELECT src FROM you will only get the src column from the database, so make sure that you are querying everything that you need from the database to avoid making multiple, meaningless queries.