tuts+

CODE > PHP

How to Paginate Data With PHP

by Jason   6 Feb 2009

Difficulty: Intermediate Length: Long Languages: English   ▾

PHP   Web Development

💬 ⤳

I can remember years ago when I first began coding in PHP and MySQL, how excited I was the first time I got information from a database to show up in a web browser.

For someone who had little database and programming knowledge, seeing those table rows show up onscreen based on the code I wrote (okay so I copied an example from a book -- let's not split hairs) gave me a triumphant high. I may not have fully understood all the magic at work back then, but that first success spurred me on to bigger and better projects.
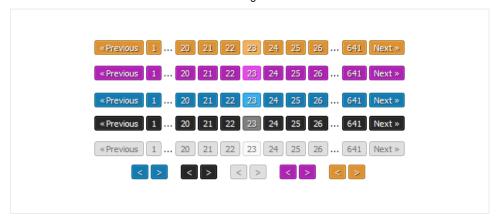
While my level of exuberance over databases may not be the same as it once was,
ever since my first 'hello world' encounter with PHP and MySQL I've been hooked
on the power of making things simple and easy to use.

As a developer, one problem I'm constantly faced with is taking a large set of information and making it easy to digest. Whether its a large company's client list or a personal mp3 catalog, having to sit and stare at rows upon rows upon rows of data can be discouraging
and frustrating. What can a good developer do? Paginate!
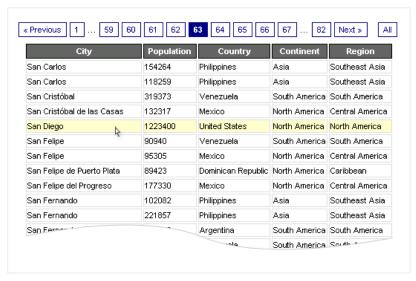
Looking for a Quick Solution?

If you're looking for a quick solution, there's a great collection of pagination scripts and helpers over at Envato Market.

This PHP pagination class is really useful. It can be implemented with any PHP supported database engine, it's very easy to customize and implement, supports two displays by default, has multicolor design schemes, and more.

1. Pagination

Pagination is essentially the process of taking a set of results and spreading them out over pages to make them easier to view.

I realized early on that if I had 5000 rows of information to display not only would it be a headache for someone to try and read, but most browsers would take an Internet eternity (i.e. more than about five seconds) to display it.

To solve this I would code various SQL statements to pull out chunks of data, and if I was in a good mood I might even throw in a couple of "next" and "previous" buttons. After a while, having to drop this code into every similar project and customize it to fit got old. Fast. And as every good developer knows, laziness breeds inventiveness or something like that. So one day I sat down and decided to come up with a simple, flexible, and easy to use PHP class that would automatically do the dirty work for me.

A quick word about me and PHP classes. I'm no object-oriented whiz. In fact, I hardly ever use the stuff. But after reading some OOP examples and tutorials, and some simple trial and error examples, I decided to give it a whirl and you know what? It works perfectly for pagination. The code used here is written in PHP 4 but will work in PHP 5.
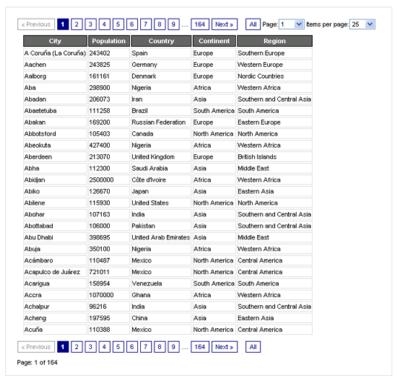
2. The Database

Gotta love MySQL. No offense to the other database systems out there, but for me, all I need is MySQL. And one great feature of MySQL is that they give you some free sample databases to play with at http://dev.mysql.com/doc/#sampledb.

For my examples, I'll be using the world database (~90k zipped) which contains over 4000 records to play with, but the beauty of the PHP script we'll be creating is that it can be used with any database. Now I think we can all agree that if we decided not to paginate our results that we would end up with some very long and unwieldy results like the following:

(click for full size, ridiculously long image ~ 338k)

So lets gets down to breaking up our data into easy to digest bites like this:

Beautiful isn't it? Once you drop the pagination class into your code you can quickly and easily transform a huge set of data into easy to navigate pages with just a few lines of code. Really.

3. Paginator

This example will be composes of two scripts, the reusable paginator class and the index file that will display the table items and controls.

Paginator.class.php

The paginator class will have only two methods and the constructor, we will build it gradually explaining each step as we move forward.

```php
01  <?php
02
03  class Paginator {
04
05      private $_conn;
06          private $_limit;
07          private $_page;
08          private $_query;
09          private $_total;
10
11  }
```

This definition only set's the paginator required member variables, since this is a helper class and it's destined for pagination only it will rely on a valid connection to the MySQL server and an already defined query that we will append the parameters necessary to paginate the results. We'll start with the constructor method.

```php
01  <?php
02
03
```

```
04   public function __construct( $conn, $query ) {
05
06       $this->_conn = $conn;
07       $this->_query = $query;
08
09       $rs= $this->_conn->query( $this->_query );
10       $this->_total = $rs->num_rows;
11
     }
```

Quite simple right? This method only set's the object's database connection and the necessary query, after that it calculates the total number of rows retrieved by that query without any limit nor skip parameters, this total is necessary to create the links for the paginator.

Note that for simplicity we are not doing error checking or any other validation of the given parameters, but in a real world application this checks will be necessary.

Retrieving Results

Now let's create the method that will actually paginate the data and return the results.

```
01   <?php
02   public function getData( $limit = 10, $page = 1 ) {
03
04       $this->_limit   = $limit;
05       $this->_page    = $page;
06
07       if ( $this->_limit == 'all' ) {
08           $query       = $this->_query;
09       } else {
10           $query       = $this->_query . " LIMIT " . ( ( $this->_page - 1 ) * $this->_limit ) . ", $this->_limit";
11       }
12       $rs              = $this->_conn->query( $query );
13
14       while ( $row = $rs->fetch_assoc() ) {
15           $results[]  = $row;
16       }
17
18       $result          = new stdClass();
19       $result->page    = $this->_page;
20       $result->limit   = $this->_limit;
21       $result->total   = $this->_total;
22       $result->data    = $results;
23
24       return $result;
25   }
```

Let's analyze this one step at a time, first we set the limit and page parameters, which by default are set the 10 and 1 respectively. The we check if the user is requiring a given number of rows or all of them, base on this and the page parameter we set the `LIMIT` parameter of the query, the "- 1" of the page is taking into account the fact that we are starting the pages in 1 instead of 0.

After this we simply evaluate the query and get the results, finally we create a new results object which contain the limit, page and total parameters of the executed query as well as the data for each of the retrieved rows.

Displaying Pagination Links

Now let's write the method used to get the pagination links.

```
01   <?php
02   public function createLinks( $links, $list_class ) {
```

```
03        if ( $this->_limit == 'all' ) {
04            return '';
05        }
06
07        $last      = ceil( $this->_total / $this->_limit );
08
09        $start     = ( ( $this->_page - $links ) > 0 ) ? $this->_page - $links : 1;
10        $end       = ( ( $this->_page + $links ) < $last ) ? $this->_page + $links : $last;
11
12        $html      = '<ul class="' . $list_class . '">';
13
14        $class     = ( $this->_page == 1 ) ? "disabled" : "";
15        $html      .= '<li class="' . $class . '"><a href="?limit=' . $this->_limit . '&page=' . ( $this->_page - 1
16
17        if ( $start > 1 ) {
18            $html   .= '<li><a href="?limit=' . $this->_limit . '&page=1">1</a></li>';
19            $html   .= '<li class="disabled"><span>...</span></li>';
20        }
21
22        for ( $i = $start ; $i <= $end; $i++ ) {
23            $class = ( $this->_page == $i ) ? "active" : "";
24            $html   .= '<li class="' . $class . '"><a href="?limit=' . $this->_limit . '&page=' . $i . '">' . $i . '<
25        }
26
27        if ( $end < $last ) {
28            $html   .= '<li class="disabled"><span>...</span></li>';
29            $html   .= '<li><a href="?limit=' . $this->_limit . '&page=' . $last . '">' . $last . '</a></li>';
30        }
31
32        $class     = ( $this->_page == $last ) ? "disabled" : "";
33        $html      .= '<li class="' . $class . '"><a href="?limit=' . $this->_limit . '&page=' . ( $this->_page + 1
34
35        $html      .= '</ul>';
36
37        return $html;
38    }
```

This is a rather long method, something like 34 lines of code, so now let's explain what is going on in this method.

1. First we evaluate if the user is requiring a given number of links or all of them, in the second case we simply return an empty string since no pagination is required.

2. After this we calculate the last page based on the total number of rows available and the items required per page.

3. Then we take the links parameter which represents the number of links to display below and above the current page, and calculate the start and end link to create.

4. Now we create the opening tag for the list and set the class of it with the list class parameter and add the "previous page" link, note that for this link we check if the current page is the first, and if so, we set the disabled property of the link.

5. At this point we display a link to the first page and an ellipsis symbol in case that the start link is not the first one.

6. Next we add the links below and above the current page based on the previously calculated start and end parameters, in each step we evaluate the current page again the link page displayed and set the active class accordingly.

7. After this we display another ellipsis symbol and the link to the last page in case that the end link is not the last one.

8. Finally we display the "next page" link and set the disabled state when the user is viewing the last page, close the list and return the generated HTML string.

That's all there is to the Paginator.class, of course we could add setters and getters for the database connection, limit, page, query and total parameters but for simplicity we'll keep it this way.

4. Index.php

Now we'll create the file in charge of using the Paginator class and displaying the data, so first let me show you the base HTML.

```
01    <!DOCTYPE html>
02        <head>
03            <title>PHP Pagination</title>
04            <link rel="stylesheet" href="css/bootstrap.min.css">
05        </head>
06        <body>
07            <div class="container">
08                    <div class="col-md-10 col-md-offset-1">
09                    <h1>PHP Pagination</h1>
10                    <table class="table table-striped table-condensed table-bordered table-rounded">
11                        <thead>
12                                <tr>
13                                <th>City</th>
14                                <th width="20%">Country</th>
15                                <th width="20%">Continent</th>
16                                <th width="25%">Region</th>
17                        </tr>
18                        </thead>
19                        <tbody></tbody>
20                    </table>
21                    </div>
22            </div>
23            </body>
24    </html>
```

Quite simple, this file only displays a table that we will populate with the information retrieved from the database, note that for this example I'm using bootstrap for basic page styling.

Using The Paginator

```
1    <?php for( $i = 0; $i < count( $results->data ); $i++ ) : ?>
2        <tr>
3                <td><?php echo $results->data[$i]['Name']; ?></td>
4                <td><?php echo $results->data[$i]['Country']; ?></td>
5                <td><?php echo $results->data[$i]['Continent']; ?></td>
6                <td><?php echo $results->data[$i]['Region']; ?></td>
7        </tr>
8    <?php endfor; ?>
```

Now for make use of our Paginator class add the following php code at the top of the document.

```
01    <?php
02        require_once 'Paginator.class.php';
03
04        $conn        = new mysqli( '127.0.0.1', 'root', 'root', 'world' );
05
06        $limit       = ( isset( $_GET['limit'] ) ) ? $_GET['limit'] : 25;
07        $page        = ( isset( $_GET['page'] ) ) ? $_GET['page'] : 1;
08        $links       = ( isset( $_GET['links'] ) ) ? $_GET['links'] : 7;
09        $query       = "SELECT City.Name, City.CountryCode, Country.Code, Country.Name AS Country, Country.Continent, Cou
10
11        $Paginator   = new Paginator( $conn, $query );
12
13        $results     = $Paginator->getData( $page, $limit );
14    ?>
```

This script is quite simple, we just required or Paginator class, note that this code assumes that this file is in the same directory as the `index.php` file, if this is not the case you should update the path accordingly.

The we create the connection to our database using the MySQLi library, retrieve the paginator parameters from the GET request and set the query, since this is not an article on MySQL or any ot that I will not get into details about the connection or the query used here.

Lastly we create the Paginator object and retrieve the results for the current page.

Displaying The Results

Now to display the obtained results add the following code to the table body.

```php
1   <?php for( $i = 0; $i < count( $results->data ); $i++ ) : ?>
2         <tr>
3                 <td><?php echo $results->data[$i]['Name']; ?></td>
4                 <td><?php echo $results->data[$i]['Country']; ?></td>
5                 <td><?php echo $results->data[$i]['Continent']; ?></td>
6                 <td><?php echo $results->data[$i]['Region']; ?></td>
7         </tr>
8   <?php endfor; ?>
```

Here we simple are iterating through the results data attribute containing the cities records and creating a table row for each one of them.

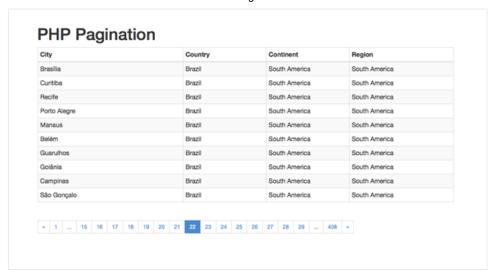## Pagination Links

Now to display the paginator links add the following code below the table.

```php
1   <?php echo $Paginator->createLinks( $links, 'pagination pagination-sm' ); ?>
```

To the Paginator createLinks method we pass the obtained *links* parameter and the css class for the pagination links used from bootstrap. Here is the result of the created page.

Conclusion

This should provide you with everything that you need to know in order to get up and running with pagination in your application.

Please don't hesitate to leave questions, comments, or general feedback in the feed below!

Still stuck? Try contacting one of our experienced PHP developers on Envato Studio.

Jason

N/A

Weekly email summary

Subscribe below and we'll send you a weekly email summary of all new Code tutorials. Never miss out on learning about the next big thing.

Email Address

Update me weekly

Translations

Envato Tuts+ tutorials are translated into other languages by our community members—you can be involved too!

Translate this post

Powered by ⊙ native

---

336 Comments   **Nettuts+**   1 **Login** ▾

♡ **Recommend** **20**   ⬆ **Share**   Sort by Best ▾

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

名前 **(必須)** • 5 years ago
This tutorial is vague!
25 ∧ | ∨ • Reply • Share ›

Inspired Prynce Kaká → 名前 (必須) • a year ago

Seems like you don't know OOP

⌃ | ⌄ • Reply • Share ›

Nouaman Bentaj Naciri → 名前 (必須) • 3 years ago

True -_-

⌃ | ⌄ • Reply • Share ›

nabeel → Nouaman Bentaj Naciri • a month ago

Multiple image upload script need for urgent base. file upload on database and display form database

⌃ | ⌄ • Reply • Share ›

Raza • 5 years ago

I have the good pagination in my site and I also attached a zip.
you can get benefit from it :-)

83 ⌃ | ⌄ • Reply • Share ›

syed → Raza • 4 years ago

where is it?

10 ⌃ | ⌄ • Reply • Share ›

mojal → Raza • 5 years ago

i need help regarding this paging.could you help me.

5 ⌃ | ⌄ • Reply • Share ›

Mary Anne → Raza • 2 years ago

can i have the copy of the source code of the pagination:) Thanks

2 ⌃ | ⌄ • Reply • Share ›

Asika Abulu → Mary Anne • 5 months ago

hello ma,can i send my source code.I am in need of paging it.Thanks in advance

⌃ | ⌄ • Reply • Share ›

Birimuye Gideon → Raza • 3 years ago

Link us to the site so that we can behold and appreciate the feel of the demo. And by the way the zipped file serves me best.

⌃ | ⌄ • Reply • Share ›

dannybarh • 5 years ago

can you pls show us how to write the sql query.

this is what i have ,but i have some errors

$query = "SELECT * FROM cntacts" .$pages->limit;

$rs_query = mysql_query($query) or die(mysql_error());

echo $num_rows = mysql_num_rows($rs_query);

$pages = new Paginator;

$pages->items_total = $num_rows[0];

$pages->mid_range = 9;

$pages->paginate();

//echo $pages->display_pages();

while ($row = mysql_fetch_assoc($rs_query))

see more

18 ^ | ⌄ • Reply • Share ›

Adedayo Adeniyi • 6 years ago

Thanks so much for this tutorial! It was a great help! I was getting Warning Notices about 'page', and 'ipp' on the first page so I modified the 'paginator.class.php' file a bit.
I added the following to the Paginator function:
if(!(isset($_GET['page']))){$_GET['page'] = 1;}
if(!(isset($_GET['ipp']))){$_GET['ipp'] = $this->default_ipp; }

Hope this helps someone else out there!

17 ^ | ⌄ • Reply • Share ›

Praveen Kumar Jha ➜ Adedayo Adeniyi • 4 months ago

^ | ⌄ • Reply • Share ›

Praveen Kumar Jha ➜ Praveen Kumar Jha • 4 months ago

Hello this is a testing comment on disquss

∧ | ∨ • Reply • Share ›

upo ➜ Adedayo Adeniyi • 3 years ago

You sir, are awesome!

∧ | ∨ • Reply • Share ›

Atul • 5 years ago

This tutorial is really good, thanks Jason.

I am new to PHP and would appreciate if somebody could supply the index.php so that I can view and compare with the one I have started.

Thanks

11 ∧ | ∨ • Reply • Share ›

Obaid Dexter • 3 years ago

Thank you very much for this tutorial. That's too helpful for guiding...

8 ∧ | ∨ • Reply • Share ›

Gizel • 3 years ago

Please hlp me...
I hve 26691 rcords...
i'm beginr..

<html>
<head><title>Entry Location</title>
<script type="text/javascript">

var TableBackgroundNormalColor = "#ffffff";
var TableBackgroundMouseoverColor = "#9999ff";

function ChangeBackgroundColor(row) { row.style.backgroundColor = TableBackgroundMouseoverColor; }
function RestoreBackgroundColor(row) { row.style.backgroundColor = TableBackgroundNormalColor; }
</script>
</head>
<body bgcolor="#EFEFEF">

<table bgcolor="#BBFFFF" bordercolor="#FF9966" width="40%" border="1">
<form action="" method="post" name="saring">

see more

6 ∧ | ∨ • Reply • Share ›

Bowenac • 4 years ago

For people asking for index.php this should be all you need.

Now this was for a custom table with wordpress but you should be able to figure it out.

```
require_once ("paginator-class.php");
global $wpdb;
// this adds the prefix which is set by the user upon instillation of wordpress
$table_name = $wpdb->prefix . "catalog";
// Counts how many rows in the table
$num_rows = $wpdb->get_var("SELECT COUNT(*) FROM $table_name" );
$pages = new Paginator;
// this sets the items_total using the count above
$pages->items_total = $num_rows;
$pages->mid_range = 7;
$pages->paginate();
// displays your page numbers
echo $pages->display_pages();
// the query to get what we are looking for from the db
$data = $wpdb->get_results( "SELECT * FROM $table_name ORDER BY post_time DESC $pages->limit");
// display the data
foreach ($data as $retrieved_data){
//example
echo "$retrieved_data->post_title"
}
```

6 ∧ | ∨ • Reply • Share ›

Daz Smith ➔ Bowenac • 3 years ago

Good, if you want to kill your server. I was doing it that way but what if you have 10,000+ results? it's going to take ages? correct me if I'm wrong, it's probably not the best way to go about it, especially if the script is going to be used by people other than yourself.

$num_rows = $wpdb->get_var("SELECT COUNT(*) FROM $table_name");

∧ | ∨ • Reply • Share ›

upo ➔ Bowenac • 3 years ago

Super useful, you saved me man

∧ | ∨ • Reply • Share ›

Shishir • 4 years ago

Where is the index.php.. page.. upload please

6 ∧ | ∨ • Reply • Share ›

Lito • 5 years ago

Thanks a lot for sharing such good script.

6 ∧ | ∨ • Reply • Share ›

Xavier Gómez • 6 years ago

Hello

I found an error on your class, when we use it and "$pages->items_total" is zero. After execute "paginate()", the value of "$pagination->limit" is "-25, 25" that cause an SQL error.

Maybe you are interested on fix and update your faboulus class.

Cheers from barcelona

6 ∧ | ∨ • Reply • Share ›

Abhimanyu ➔ Xavier Gómez • 4 years ago

This is what I did:

if ($num_results != 0) {
$pages->paginate();
}

This way, the rest of the page continues to load but the paginate function does not run.

3 ∧ | ∨ • Reply • Share ›

James ➔ Xavier Gómez • 4 years ago

This is correct, I would also like a fix

3 ∧ | ∨ • Reply • Share ›

Guest • 4 years ago

Where in the index.php.. page.. upload please

5 ∧ | ∨ • Reply • Share ›

Nathan Daly • 4 years ago

Thanks for this great tutorial, very well written! I know it was posted all the way back in 2009 but I have just used this class for my WordPress Plugin and it works great.

I had to make a couple of changes to get this working in WordPress:

1) Had to change `href=\"$_SERVER[PHP_SELF]?page=$prev_page&ipp=` to `href=\"?page=your_plugin&pg=$i&ipp=` so it would stay on the plugin page I was viewing.

2) I also had to change all the `$_GET['page']` to `$_GET['pg']` as there was already a post variable WordPress was using called

"Page".

Just a heads up if you want to integrate this into WordPress, other than that, great article! :)

5 ∧ | ∨ • Reply • Share ›

arlon → Nathan Daly • 4 years ago

you said you changed it to href=\"?page=your_plugin&pg=$i&ipp= then how did you write your plugin there? is it like this_plugin.php or the root_folder(of plugin)?

∧ | ∨ • Reply • Share ›

bobzyouruncle • 5 years ago

Now a great follow up tut would be using this with ajax. Please do it one day.

5 ∧ | ∨ • Reply • Share ›

anand • 4 years ago

Hi I am new in php I m trying to get pagination in my index.php and I am using sqlsrv database it is not showing pagination. plz tell me how it can work step by step with the database of sqlsrv.

3 ∧ | ∨ • Reply • Share ›

Android Example • 4 years ago

Nice Tutorial Dude ....

thanks

3 ∧ | ∨ • Reply • Share ›

Mander • 4 years ago

wooot thx !. took me 5 min to understand/implement pretty nice

2 ∧ | ∨ • Reply • Share ›

Nadia • 4 years ago

Thank you so much for your great pagination class. You have saved me heaps of time in writing something similar from scratch which likely wouldn't be even half as good as this. Highly appreciate you sharing it.

2 ∧ | ∨ • Reply • Share ›

mojal • 5 years ago

help me to fixed the error

2 ∧ | ∨ • Reply • Share ›

DasLicht → mojal • 3 years ago

which?

∧ | ∨ • Reply • Share ›

Aissa • 5 years ago

Hi guys, I have a problem with this code! It works great except when I use it in my search form. When I fetch my results it shows all the INPUTS the form has in the pages link. For example page1 page2 page3 when I hover the mouse over I could see all the form inputs, and when I click on page2 to show the next results, it shows everything:
Anybody else got into this issue??

2 ∧ | ∨ • Reply • Share ›

LAura • 5 years ago

Hi, is the css used available? It is not in the download link...
Thanks for the tutorial BTW

2 ∧ | ∨ • Reply • Share ›

Ena Bartosh • 6 years ago

Pretty component to content. I just stumbled upon your weblog and in accession capital to say that I get in fact enjoyed account your weblog posts. Any way I'll be subscribing in your feeds or even I success you get admission to consistently quickly.

2 ∧ | ∨ • Reply • Share ›

Tommy • 6 years ago

Absolutely perfect tutorial, clearly explained, easy to understand.

Although I didn't get to finish it off becuase MSSQL doesn't allow LIMIT 0,25! If I can get this to work with MSSQL, I'll let ya'll know.

Unless anyone else has, then let me know!!!!!!!

2 ∧ | ∨ • Reply • Share ›

    NROME ➜ Tommy • 5 years ago

    Tommy,

    MSSQL doesn't have LIMIT in it's library. It uses the keyword TOP.

    (i.e.

    MSSQL:

    SELECT TOP 1 tablename.fieldname
    FROM tablename
    WHERE tablename.fieldname = 'criteria'

    over

    MySQL:
    SELECT tablename.fieldname
    FROM tablename
    WHERE tablename.fieldname = 'criteria' LIMIT 1

    4 ∧ | ∨ • Reply • Share ›

Dee • 7 years ago

Super Awesome This was VERY helpful. Y'all are great. (hug)

2 ∧ | ∨ • Reply • Share ›

    mnj ➜ Dee • 5 years ago

    jh

    1 ∧ | ∨ • Reply • Share ›

Bob Belderbos • 4 years ago

This is very useful, could plug it in directly into my project, thanks!

1 ∧ | ∨ • Reply • Share ›

Alessandro Reis • 4 years ago

Does anybody know why the button "Previous" and "Next" only shows when Items per page = 10?

1 ∧ | ∨ • Reply • Share ›

    DasLicht ➜ Alessandro Reis • 3 years ago

    Have a look here:

    https://gist.github.com/das...

    ∧ | ∨ • Reply • Share ›

geno11x11 • 4 years ago

Having a little problem... The first page displays 25 items but navigation to other pages or "all" do nothing. I have made a few minor changes to the class to eliminate errors but nothing that would change the logic. Has anyone else had this problem -- or any suggestions?

1 ∧ | ∨ • Reply • Share ›

    geno11x11 ➜ geno11x11 • 4 years ago

    Just discovered that the page number and items per page are listed accurately but again, select a page or changing the number of items displayed per page does not work.

    1 ∧ | ∨ • Reply • Share ›

iven • 4 years ago

dears can i get full file sample for this link

http://www.catchmyfame.com/...

plz

my email : ivenprvb@gmail.com

1 ⌃ | ⌄ • Reply • Share ›

Siby P Varkey • 5 years ago

Thank you very much. Great script, Great help. I was experimenting with the script for the last two days to solve one issue with it. The problem was that it couldn't display the rows of the search query of mine, which is comprised of many variables. At last a few hours back, I found-out the culprit, the POST variables, which I changed to GET, which solved my problem. So my experience makes me say "DON'T USE POST, BUT USE GET variables with this script". Thanks once again.

1 ⌃ | ⌄ • Reply • Share ›

Load more comments

✉ Subscribe   ⒹAdd Disqus to your siteAdd DisqusAdd   🔒 Privacy

Advertisement

ENVATO TUTS+   +

JOIN OUR COMMUNITY   +

HELP   +

🌱   tuts+

24,220   1,034   14,475
Tutorials   Courses   Translations

Envato.com   Our products   Careers

Follow Envato Tuts+