



I used two of these RSS feeds in the website, one to pull in the feed for the 'most popular' Tv shows and one to fetch BBC websites home news content. For each one I employed a different syntax method to fetch, echo and format the information. This was necessary due to the coding of the xml files they were being drawn from and how the php code would read them.



The php code for the Tv show had to be simplified to allow it to find the pertinent tags in the xml document. Comparing the two hyperlinks to the RSS files we can see that the TV show does not have the xml extension and in my summation I feel that this is why the example code provided didn't work.

```
(TV Show)
$feed_url='http://feeds.bbc.co.uk/iplayer/popular/tv';
$feed_content=simplexml_load_file($feed_url);
$item=0;
echo $feed_content->title;
while ($item<10) {
echo $feed_content->entry[$item]->title;
echo $feed_content->entry[$item]->content;
```



The cached html file for the TV Show also revealed styling tags around the 'title' and 'content' and pulled in the image files along with the content. Whereas the News data didn't and had to be formatted within the php; this would normally have been done via the xml style sheet attached to the webpage it was rendered out in.

```
(News)
$feed_url='http://feeds.bbci.co.uk/news/rss.xml';
$feed_content=simplexml_load_file($feed_url);
$item=0;
echo "<h4 >". $feed_content->channel->title."</h4>";
while ($item<3) {
echo "<p class='title's". $feed_content->channel->item[$item]->title."</p>";
echo "<p class='pubDate's". $feed_content->channel->item[$item]->pubDate."</p>";
echo "<p>". $feed_content->channel->item[$item]->description."</p>";
$hp = $feed_content->xpath('media:thumbnail[0]');
echo $hp[0]['url'];
echo "&nbsp;<a href='\".$feed_content->channel->item[$item]->link.\">Read More</a>";
echo "<hr />";
$item++;
```

After some investigation into how I could pull the media:thumbnail image from the news feed I created the (\$hp = \$feed\_content->xpath('media:thumbnail[0]');echo \$hp[0]['url'];) echo however it does not work I do not think I quite have the syntax's correct and I am still researching this element of code.



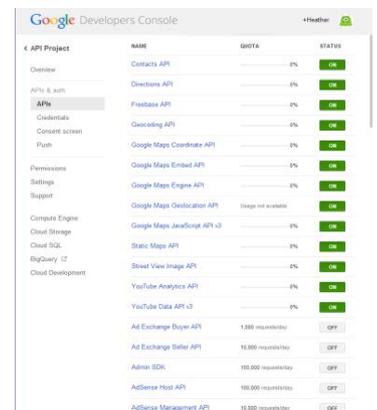
An Application Programming Interface (API) within a webpage is used to share data and information across the web. The most common use is within the distribution of images and sound files in social media sites. We don't see their usage but every time we share a photograph on Facebook we use an api to propagate the image from its current location to its shared one; the original doesn't actually move.

For instance using the Flickr api and a few selected search words I was able to pull the images displayed at the top of the website (see image above) from their extensive gallery's. For a nice style affect I merged the Flickr search with a Last FM 'audioscrobbler' search for information on an artist called BBC. They actually returned a bio summery of the British Broadcasting Corporation and a link to where I could find information on a band called BBC.

To use an api you require a key code; google has a developer platform where you can set up a key code that will control all or as many of its api applications as you want. However for Flickr and Last Fm I had to create individual api keys for each site to be able to run the php searches.

```
Last Fm - $data = "api_key=c8bc9a5b4ba188db3326b2699b44a45a&method=artist.getInfo&artist=BBC";
Flickr - $api_key = '10728b15a83509cf578a94f27e6f6544';
```

AS well as using different key codes I also had to use different php syntax's to access the relevant data bases where the information was stored. Obviously searching for text as in an artist bio would require different search terms than searching for an image.



With Last FM I used 'curl' a protocol server library that I installed on my server space using my webmin interface so when I sent the request for the artist information it could send and receive the applicable data

packages then display them on my website. Flickr uses the 'soap' protocols which are supported by the Apache server application I had already set up.

The code for the Flickr search came from a tutorial in a web development blog. (Olaf Lederer (2013) *Search for photos using PHP and the flickr API*, Available at:<http://www.web-development-blog.com/archives/search-for-photos-using-php-and-the-flickr-api/> (Accessed: 30th April 2014).) I first copied and pasted the original tutorial code which worked quite well but resulted in very large images. So I checked the reply's within the blog which normally give you some hints on how to adapt the original code and found a solution to the size problem which was to concatenate flicker's own size constraints to the image echo code.

```
s = "Square" width="75" height="75"
l = "Large Square" width="150" height="150"
t = "Thumbnail" width="100" height="75"
m = "Medium" width="500" height="375"
```

```
echo '';
```

This was very successful so I checked back to see if there was any of the other bloggers had developments I could use and found a line of code that would create hyperlinks out of the images to their original source on Flickr.

```
echo '<a href="http://www.flickr.com/photos/' . $photo["owner"] . '/' . $photo["id"] . '/' target="_blank" rel="nofollow">
</a>';
```

The Flickr code for the header image search is in two parts the api key is stored in a separate php file that is pulled into the search with the line (`require_once('flickr.php');`) which is the php function path to external file content. The external file also controlled how many images would be displayed.



The Last Fm api key was also used with the 'Get similar Artist' feature where two audioscrobbler searches are mashed together with a Flickr image search. The searches are then looped to allow the user to delve into the depth of artists who play similar music to the BBC Band by clicking on the name of a band or go to Flickr and see and more images of the band.

It is easy to copy and paste the contents of a webpage from the screen or the source code but why have all that data clogging up your page code have when with just a few lines of php you can scrape the bits you want. The benefit of this is that if the data source is updated you get the updates as well. There are plenty of websites offering scrape bot software for nicking your competitors data; google chrome apparently has a scraper extension built in.

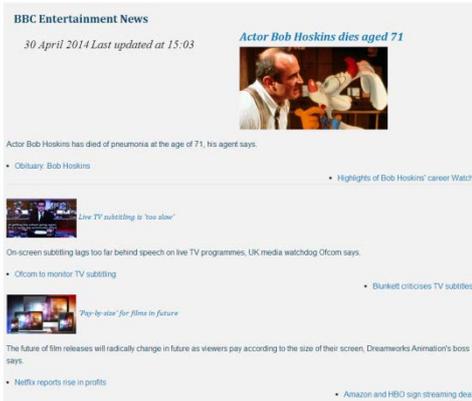
There are pros and cons to the legality of scraping websites; the former head of IT security at Ladbrokes Mike Gaffney was quoted as saying "Ladbrokes was blocking about one million IP addresses on a daily basis," However according to Frances Irving, the chief of ScraperWiki "Google and Facebook effectively grew up scraping," and the face of the internet would look very different without it. (Mark Ward (30 September 2013) *Screen scraping: How to profit from your rival's data*, Available at: <http://www.bbc.co.uk/news/technology-23988890> (Accessed: 30th April 2014).)

So if everyone is doing it how did I do it? Having so far unsuccessfully scraped the BBC iPlayer and most of the website either based on news or television I decided to go with the BBC Entertainment news page. The simple styling and framework layout with divs columns and headers was perfect for scraping with the code I had to hand.



This is how I did it .....

- Step 1 – set a variable for where to find the data / \$url="http://www.bbc.co.uk/news/entertainment\_and\_arts/";
- Step 2 – set a variable to get and store the data / \$source = file\_get\_contents(\$url);
- Step 3 – set the start position for the source to search from / \$cdstart=strpos(\$source,'<div id="full-width" class="container-full-width">')+1002;
- Step 4 – set the start position for the source to search from / \$cdstart=strpos(\$source,'<div id="full-width" class="container-full-width">')+1002;
- Step 5 – change the hyperlinks from local to global so they work / \$sourceedit=str\_replace("/news/", "http://www.bbc.co.uk/news/", \$source);
- Step 6 – echo the source path out on to the webpage / echo substr(\$sourceedit, (\$cdstart), (\$cdend-\$cdstart));



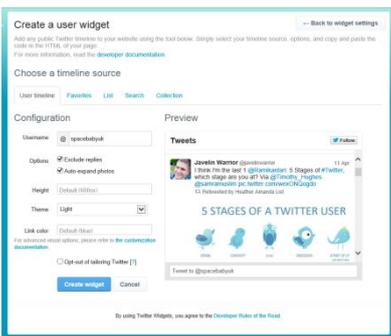
With a little extra styling I was able to pull the top headline stories and add them to one of the divs in my webpage. Using the 'replace' command php I changed the hyperlinks...  
 From - http:news/entertainment...  
 To - http://www.bbc.co.uk/news/entertainment...  
 This made them absolute links instead of local so they would work from the website.

One of the problems I had scraping the information was getting the code to independently pull in the data between the two div tags without bringing with it any extras such as advertisements that were associated with the divs class tags and the websites style sheets. To get only the information I had instructed the php code to fetch, I had to tell it to ignore the character spaces before and after the start and stop code lines. "+1002" This number was found with some trial and error but it is interesting that both the superfluous data before and after was the same length.

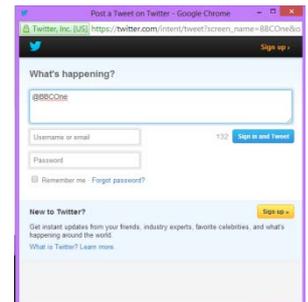
### What I achieved

In the end I was unable to provide my client with the website service I wanted to design for them. I did however create a mock up to show some of the elements a fanzine site could pull into its page using RSS feeds Api's and a little scraping.

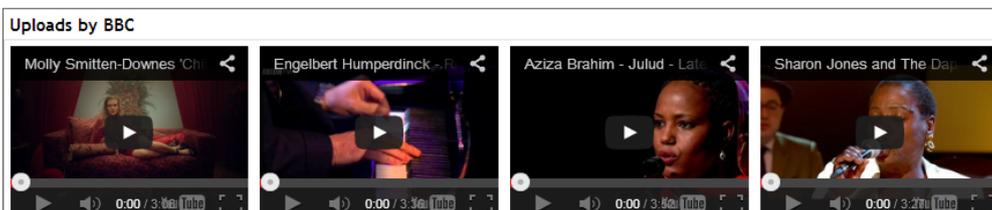
No fanzine page would be complete without a Facebook or twitter feed and this one is no exception. The Facebook php code is from an old website I created some years ago and I have completely lost track of where I acquired the simple code but it still works.



Twitter has its own widget creator which you can find via your twitter page settings. With it you can create tweet link feeds from various sources including your own. If you choose the right format you can create a feed that will allow you to tweet from the website it is embedded in.



Widgets have been around since the internet went public to allow none code savvy web developers and fanatics dedicate they server space to their latest fad. I finished my BBC fan page with a widget created by a company called RSSinclude.com. They offer a free thirty day trial after that you are required to pay for the service so I suspect my widget will stop working very shortly. This YouTube widget of the latest BBC uploads updates each time you reload the page; well it will if there have been any new uploads to YouTube with the BBC tag.



---

## Webpage Resources

---

### Spacebabyuk.com/index.php

---

```
<?php

// Identify a cache file
$cachefile = 'cached.html';

// Start the output buffer
ob_start();
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
<!-- This DTD is equal to XHTML 1.0 Transitional, but allows the use of frameset content.-->
<!-- Changed my doc type because the w3 validator did not like it I halved my code errors by changing to a strict doc type
H.A.List-->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<title>Scrape and Mash</title>
  <!-- basic foundation frame work style sheet H.A.List-->
  <link rel="stylesheet" href="css/foundation.css" type="text/css"/>
  <!-- conditional styles sheet to change discreet elements H.A.List-->
  <link rel="stylesheet" href="css/head.css" type="text/css"/>

<!-- Internal Page Scripts -->

<script type="text/javascript">
(function(d, s, id) { //facebook function call ... A little bit of dom to fetch facebook feed H.A.List
var js, fjs = d.getElementsByTagName(s)[0];
if (d.getElementById(id)) {return;}
js = d.createElement(s); js.id = id;
js.src = "//connect.facebook.net/en_US/all.js#xfbml=1";
fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));</script>
<!-- End Internal Page Scripts -->

</head>
<body>

<!-- Page Header -->

  <div class="head"><!--open head div H.A.List-->
    <div class="large-1 small-9 columns">
       <!-- icon image for top header using url H.A.List-->
    </div>
    <div class="large-8 small-9 columns">
      <h1>The BBC Fanzine </h1>
    </div>
  </div><!--close head div H.A.List-->
```

```

<div class="contain-to-grid sticky"><!--open container div this sticky effect works on my other website not sure why it does not
work on here H.A.List-->
  <nav class="top-bar"><!-- open nav H.A.List -->
    <ul class="title-area">
      <li class="name">
        <h1>Todays Highlights from the BBC</h1>
      </li>
    </ul>
  </nav><!--end nav div H.A.List -->
</div><!--end container H.A.List -->

<!-- End Page Header -->

  <!-- Top Section LastFm and Flickr api H.A.List-->
<div class="row"><!-- open row H.A.List -->
  <div class="large-12 columns"><!-- create full width column H.A.List -->
    <div class="callout panel"><!--add container H.A.List -->

<?php
// Do a POST
$data = "api_key=c8bc9a5b4ba188db3326b2699b44a45a&method=artist.getInfo&artist=BBC";//api key method of search text
BBC = artist name and looking for associated info as well H.A.List
$ch = curl_init('http://ws.audioscrobbler.com/2.0/');// the target ....database of music information
curl_setopt($ch, CURLOPT_POST, 1); // telling curl to POST
curl_setopt($ch, CURLOPT_POSTFIELDS, $data); // telling cURL what to POST
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // stops the output from appearing on screen
$strPage = curl_exec($ch); // we store the returned XML in a string
curl_close($ch); // Closing connection
//Curl is a server protocol software to send, receive, sort and post data in a website H.A.List

$xml = simplexml_load_string($strPage); // we use SimpleXML to convert the XML to an array
$xml_array = array();
foreach($xml->xpath('artist') as $artist) // Reads the whole XML tree into the array
{
  // H.A.List brackets to echo the contents of our search onto the page

  echo "<h4><strong>".$artist->name." ".LastFm."</strong></h4>";
  // H.A.List a little inline styling to make the artist name appear on the page with the same styling as the other
  headings plus the addition of the words Last FM to clarify the heading

  echo "<hr/>";// H.A.List a little line across the page for styling
}

  // Flickr code obtained from tutorial at http://www.web-development-blog.com/archives/search-for-photos-using-php-
  and-the-flickr-api/ H.A.List
  require_once('flickr.php');// link to external php file with flickr api H.A.List
  $Flickr = new Flickr;
  $data = $Flickr->search('bbc news');// search term for images H.A.List
  foreach($data['photos']['photo'] as $photo)
  {
    // http://farm{farm-id}.static.flickr.com/{server-id}/{id}_{secret}.jpg
    // The path created above renders out like the example below H.A.List
    // farm = farm = photos server id is where to find image and secret is a code that achieves the image
    // https://www.flickr.com/photos/42132002@N05/14048413746/

/*
without size concatenation H.A.List
echo '';
without hyperlink H.A.List
echo '';

```

```

*/
// H.A.List with hyperlink and size restriction
echo '<a href="http://www.flickr.com/photos/' . $photo["owner"] . '/' . $photo["id"] . '/' target="_blank" rel="nofollow">
</a>';

}
{
    echo "<hr/>"; // H.A.List a little line across the page for styling
        echo $artist->bio->summary; // H.A.List echo out the artist bio summary search results from the Last Fm search
at the start of this php segment
}

?><!-- end the php H.A.List -->

</div>
</div>
</div>

<!-- BBC iplayer feed -->
<div class="row"><!--row containing 2 columns 6 columns wide H.A.List-->
    <div class="large-6 medium-6 columns">
        <div class="callout panel">

<?php
$feed_url='http://feeds.bbc.co.uk/iplayer/popular/tv'; // H.A.List I choose to use the popular tv feed
$feed_content=simplexml_load_file($feed_url);//load feed url into content variable H.A.List
$item=0; // H.A.List start at first item

echo "<h4 >". $feed_content->title."</h4>"; // H.A.List A little inline styling for title
while ($item<5) { // H.A.List output 5 items
echo $feed_content->entry[$item]->title; // H.A.List output title tag content
echo $feed_content->entry[$item]->content; // H.A.List output contents tag contents

$item++; // H.A.List a little loops code
}

?>

</div><!--end panel H.A.List-->

<div class="callout panel"><!-- open new panel H.A.List--> <!-- H.A.List url of tweets to feed-->
<!--Twitter data widget using JavaScript just to show you do not need to be a pro coder to add feed content to your website
H.A.List-->
    <a class="twitter-timeline" href="https://twitter.com/BBCOne"
        data-widget-id="460703617050107904">Tweets by @BBCOne</a><!-- H.A.List widget id code-->
    <script type="text/javascript" >!function(d,s,id){var
js,fjs=d.getElementsByTagName(s)[0],p=/^http:/.test(d.location)?'http':'https';if(!d.getElementById(id)){js=d.createElement(s);js.i
d=id;js.src=p+"://platform.twitter.com/widgets.js";fjs.parentNode.insertBefore(js,fjs);}(document,"script","twitter-
wjs");</script>

</div><!--end panel H.A.List-->
</div> <!--end 6 columns H.A.List-->

<!-- BBC flickr search 2-->
<div class="large-6 medium-6 columns">

<div class="callout panel">

<?php
$ra="BBC"; //Set the default artist
// Do a POST if the form is completed
if (isset ($_POST['artist'])) {

```

```

$ra=$_POST['artist'];
}
// H.A.List two piece of code to post when form is activated and get info when hyperlink is clicked
// Do a GET if the artist hyperlink is clicked
if (isset ($_GET['artist'])) {
$ra=$_GET['artist'];
}

$page= $_SERVER['SCRIPT_NAME']; //store the current script location
?>

<form action="<?php echo $page;?>" method="post" enctype="application/x-www-form-urlencoded">

    <label class="bbc"><!-- H.A.List class style to format form-->
    Artists similar to:
    <input name="artist" type="text" id="artist" value="<?php echo $ra;?>" /><!-- H.A.List add $ra variable to input box-->
    </label>
    <label>
        <input type="submit" name="submit" id="submit" value="Change" />
    </label>

</form>

<?php
// H.A.List api key with method to get similar artist limited to 6 (this can be changed) and equal to artist input variable
$data = "api_key=c8bc9a5b4ba188db3326b2699b44a45a&method=artist.getSimilar&limit=6&artist=".$ra;

$ch = curl_init('http://ws.audioscrobbler.com/2.0/'); // the target
curl_setopt($ch, CURLOPT_POST, 1); // telling cURL to POST
curl_setopt($ch, CURLOPT_POSTFIELDS, $data); // telling cURL what to POST
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // stops the output from appearing on screen
$strPage = curl_exec ($ch); // we store the returned XML in a string
curl_close($ch); // Closing connection
// H.A.List api key with method to get artist image and equal to artist input variable
$data2 = "api_key=c8bc9a5b4ba188db3326b2699b44a45a&method=artist.getImage&artist=".$ra;

$ch2 = curl_init('http://ws.audioscrobbler.com/2.0/'); // the target
curl_setopt($ch2, CURLOPT_POST, 1); // telling cURL to POST
curl_setopt($ch2, CURLOPT_POSTFIELDS, $data2); // telling cURL what to POST
curl_setopt($ch2, CURLOPT_RETURNTRANSFER, 1); // stops the output from appearing on screen
$strPage2 = curl_exec ($ch2); // we store the returned XML in a string
curl_close($ch2); // Closing connection

//echo $strPage2;
$xml = simplexml_load_string($strPage); // we use SimpleXML to convert the XML to an array
    $xml_array = array();
// H.A.List two variables to operate loop for similar artist and image search
$xml2 = simplexml_load_string($strPage2); // we use SimpleXML to convert the XML to an array
    $xml_array2 = array();

foreach($xml->xpath('similarartists/artist') as $artist)

{
    $artist_url=urlencode($artist->name); //encode the artist's name so that spaces and other characters are
transferred properly in the GET
    $image= $artist->image; // put the image url into a variable for convenience
    echo "<a href='http://www.flickr.com/search/?q=".$artist_url."&s=int'><img src='".$image.'" /></a>"; //
output the image
    echo "<a href=".$page."?artist=".$artist_url.">".$artist->name.</a> "; //output the artist's name and url to
reload the page

}

```

```

        foreach($xml2->xpath('images/image/sizes/size[1]') as $image)
        {
            $original=$xml2->xpath('images/image/sizes/size');
            echo $original;
            $i= $image;
            echo "<img src='\".$i.\" ' />";

        }
    ?>

    </div><!--end panel H.A.List-->

<!-- BBC news feed rss designed using iplayer feed code example H.A.List-->
    <div class="callout panel">

<?php
$feed_url='http://feeds.bbci.co.uk/news/rss.xml';// H.A.List url of where to fetch feed from
$feed_content=simplexml_load_file($feed_url);
$item=0;// H.A.List start at the first item in the rss feed

echo "<h4 >". $feed_content->channel->title."</h4>";// H.A.List path to xml feed with some inline styling
// H.A.List while loop for items =3
while ($item<3) {
/* Removed number of feed code below to make look more professional H.A.List */
/*echo "<br />".($item+1)."&nbsp;"; */
echo "<p class='title'>". $feed_content->channel->item[$item]->title."</p>";// H.A.List in-style code because this feed dose not
come with any html tags
echo "<p class='pubDate'>". $feed_content->channel->item[$item]->pubDate."</p>";// H.A.List added feed item for published
date
echo "<p>". $feed_content->channel->item[$item]->description."</p>";// H.A.List added description feed

/* this thumbnail feed should work but it dose not H.A.List */
$hlp = $feed_content->xpath('media:thumbnail[0]');
echo $hlp[0]['url'];
echo "&nbsp; <a href =\".$feed_content->channel->item[$item]->link.\">Read More</a>";
echo '<hr />';// H.A.List a little line to seperate each news item

$item++;
}

?>

    </div><!--end panel H.A.List-->

    <div class="callout panel">

        <h4>BBC Facebook</h4>
<!--Facebook social plugin H.A.List-->
        <div class="fb-like-box" data-href="https://www.facebook.com/bbcnews" width="435" height="550"
            data-show-faces="false" data-stream="true" data-header="true"></div>
<!--this is a piece of code I used in a previous assignment not sure where I got it from but it still works H.A.List-->

        </div><!--end panel H.A.List-->

    </div><!--end 6 columns H.A.List-->
    <hr/><!-- H.A.List a little line across the page for styling-->
</div> <!--end row H.A.List-->

<!-- BBC entertainment page scrape H.A.List-->
<div class="row">
    <div class="large-12 columns">
        <div class="callout panel">
            <h4>BBC Entertainment News</h4>

            <?php

```

```

//first read the file into a variable
$url="http://www.bbc.co.uk/news/entertainment_and_arts/";
$source = file_get_contents($url);

//identify the character position of the start and end unique text labels..... H.A.List this normally works better with a div tag
$cdstart=strpos($source,'<div id="full-width" class="container-full-width">')+1002;//removes advertising java script that tags
along when doing sneaky change below for hyperlinks H.A.List
$cdend=strpos($source, '<div id="other-top-stories" class="other-top-stories">')+250;// H.A.List remove extra data that is pulled
in with the div tags

//create an HTML wrapper for the output
echo '<!DOCTYPE html>
<html lang="en" class="no-js">
<head> <meta charset="utf-8"> </head>
<div class="row">
  <div class="large-6 medium-6 columns">
    <div class="callout panel"><span class="time-text">;
// H.A.List change the hyper-links from local to absolute so they still work
$sourceedit=str_replace("/news/", "http://www.bbc.co.uk/news/", $source);
// fix broken links using this technique too

//output the scrape
echo substr($sourceedit, ($cdstart), ($cdend-$cdstart));

//close the wrapper
echo ' </div>
  </div>
</div>

//close the wrapper
</html>';

?>
  </div><!--end -->
  </div><!--end -->
</div><!--end -->

  <!-- youtube widget created with https://www.rssinclude.com/ sometimes this doesnt work H.A.List -->
<div class="row">
  <div class="large-12 columns">
    <div class="callout panel" >
      <hr/><!-- H.A.List a little line across the page for styling-->
      <?php
@readfile('http://output23.rssinclude.com/output?type=php&id=870000&hash=787690cb9e0b051e6122f8ae94f0aea2')?>
      <hr/><!-- H.A.List a little line across the page for styling-->
    </div>
  </div>
</div>

<script src="js/jquery.js" type="text/javascript" ></script>
<script src="js/foundation.min.js" type="text/javascript" ></script>

<script type="text/javascript" >
  $(document).foundation();
</script>

</body>
</html>

<?php

// Cache the output to a file

```

```
$fp = fopen($cachefile, 'w');  
fwrite($fp, ob_get_contents());  
fclose($fp);  
ob_end_flush(); // Send the output to the browser
```

```
?>
```

[Spacebabyuk.com/flickr.php](http://Spacebabyuk.com/flickr.php)

```
<?php
```

```
class Flickr {  
    private $apiKey = '10728b15a83509cf578a94f27e6f6544'; // api key H.A.List  
  
    public function __construct() {  
    }  
  
    public function search($query = null) {  
        $search = 'http://flickr.com/services/rest/?method=flickr.photos.search&api_key=' . $this->apiKey . '&text=' .  
urlencode($query) . '&per_page=11&format=php_serial';  
        $result = file_get_contents($search);  
        $result = unserialize($result);  
        return $result;  
    }  
}  
?>
```

---