

```

<?php
/*
* Contact: J Wynia <j@wynia.org>
* Description: Connects to a Wordpress database and figures out what
* your longest, shortest and average post titles are and picks 4 out
* as samples for you to use as representative.
*
* You get the longest, the shortest and an average diverse (measured
* by the number of unique characters) and long diverse title.
*
* Set the database parameters to point to your Wordpress installation.
* Then set the CSS to match the look and width of posts you're aiming for.
*
* Copyright (c) 2007, J Wynia
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions are met:
* * Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* * Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* * Neither the name of J Wynia nor the
* names of its contributors may be used to endorse or promote products
* derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY J Wynia ``AS IS" AND ANY
* EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL J Wynia BE LIABLE FOR ANY
* DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

//Database parameters
$dbhost = "localhost";
$dbuser = "username";
$dbpass = "password";
$dbname = "wordpress_db";
?>
<html>
<head>
<title>Wordpress Title Sampler</title>
<style>
h1 {
font-family: Georgia;
font-size: 18px;
width: 550px;

```

```

}
</style>
</head>
<body>
<?php
// Connecting, selecting database
$link = mysql_connect($dbhost, $dbuser, $dbpass)
    or die('Could not connect: ' . mysql_error());
mysql_select_db($dbname) or die('Could not select database');

//Query for average post length
$avgquery = "SELECT AVG(
CHAR_LENGTH(post_title)
)
FROM wp_posts;";
//Get the average post length
$avgresult = mysql_query($avgquery) or die('Query failed: ' . mysql_error());
$avgrow = mysql_fetch_row($avgresult);
//Round it to the nearest integer.
$avg = round($avgrow[0]);

//Query to get the list of posts and their title length
$query = "SELECT post_title, CHAR_LENGTH(post_title)
FROM wp_posts
ORDER BY CHAR_LENGTH(
post_title
) DESC;";
$result = mysql_query($query) or die('Query failed: ' . mysql_error());

//Set up an array for each: posts with greater than average title length,
//less than average and average.
$gt_array = array();
$lt_array = array();
$eq_array = array();

//Loop through the complete list of posts and process
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
    //Get the specific field values
    $title = $line['post_title'];
    $length = $line['CHAR_LENGTH(post_title)'];
    //Figure out how many unique characters are in the
    //title. Split up one char per key in array.
    $array_title = str_split($title);
    //Eliminate the duplicates and count
    $unique_chars = count(array_unique($array_title));

    //Put the current row into a temp array
    $temp_array = array();
    $temp_array['unique_chars'] = $unique_chars;
    $temp_array['length'] = $length;
    $temp_array['title'] = $title;

    //See which destination array it belongs in
    //and assign it.

```

```

if($length >$avg){
  $gt_array[] = $temp_array;
}
elseif ($length ==$avg)
{
  $eq_array[] = $temp_array;
}
elseif($length<$avg)
{
  $lt_array[] = $temp_array;
} else {
}
}
}

```

//Get the highest and lowest before we sort for diversity

```

$greatest = $gt_array[0];
$lt_count = count($lt_array);
$least = $lt_array[$lt_count-1];

```

//Sort by diversity, highest first

```

rsort($gt_array);
rsort($eq_array);

```

//Get the new sample values

```

$gt_diverse = $gt_array[0];
$eq_diverse = $eq_array[0];

```

//Print out a table with our 4 selected post titles.

```

print("<table>");
print("<tr><td>Longest Title:</td>");
print("<td><h1> " . $greatest['title']. "</h1></td>");
print("</tr>");
print("<tr><td>Longest Diverse Title:</td>");
print("<td><h1> " . $gt_diverse['title']. "</h1></td>");
print("</tr>");
print("<tr><td>Average Diverse Title:</td>");
print("<td><h1> " . $eq_diverse['title']. "</h1></td>");
print("</tr>");
print("<tr><td>Shortest Title:</td>");
print("<td><h1> " . $least['title']. "</h1></td>");
print("</tr>");
print("</table>");

```

?>