

```

1  <script>
2  function submitTime() {
3      var now = new Date();
4      var subTime = (Math.floor(now.getTime()/1000)).toString(36);
5      var cht = $('#secval');
6      cht.val(subTime);
7  }
8  function addURL(element) {
9      var now = new Date();
10     var subTime = (Math.floor(now.getTime()/1000)).toString(36);
11     $(element).attr('href', function() {
12         return this.href + "&cht="+subTime;           I
13     });
14 }
15 </script>
16
17 <?php
18 date_default_timezone_set("UTC"); // Same timezone in PHP and JS
19 $rdtime = time();
20 $expired = false;
21 if (isset($_GET['cht'])) { // Check for an old URL (bookmark)
22     $cht=$_GET['cht'];
23     $cht = base_convert($cht, 36, 10); // Extract time string to number_base 10
24     if( ($rdtime - $cht) > 5 || ($rdtime - $cht) < 0) $expired = true;
25 } else { $cht = 0; }
26
27 // Pick the expired query data
28 if ($expired == true && isset($_GET['zoom_query']) && isset($_GET['zoom_and']) ) {
29     $bookmark = true;
30     $qry_string = $_GET['zoom_query'];
31     $qry_and = $_GET['zoom_and'];
32 } else {
33     $bookmark = false;
34     $qry_string = '';
35     $qry_and = 0;
36 }
37
38 if ( $cht == 0 || $expired == true) { // No fresh action from 'search.php'
39     unset($_GET['zoom_query']); // Clear zoom_query (search box) --> Stops further action
40 }
41 ?>
42
43 <script>
44 // Prefill the form when an expired bookmark -query
45 $(document).ready(function() {
46     var expired = <?php echo json_encode ($expired); ?>;
47     var qry_string = <?php echo json_encode ($qry_string); ?>;
48     var zoom_match_all = <?php echo json_encode ($qry_and); ?>;
49     if (expired == true) {
50         var qry_box = $('#zoom_searchbox');
51         qry_box.val(qry_string); // Query-string
52         var qry_and = $('#zoom_match_all');
53         if (zoom_match_all == 1) qry_and.val = checked; // And/Or -selection
54     }
55 });
56 </script>
57
58 <?php
59
60 // -----
61 // Zoom Search Engine 8.0 (16/Jan/2019)

```

```

$templateQueryTag = "<!--ZOOM_SHOW_QUERY-->";
$templateQueryTagLen = strlen($templateQueryTag);

$outputBuffers[OUTPUT_FORM_START] = "<form onsubmit=\"submitTime()\" method=\"get\" action=\"\".$SelfURL.\"\" class=\"zoom_searchform\">";
$outputBuffers[OUTPUT_FORM_START] .= "<input type=\"hidden\" name=\"cht\" value=\"\" id=\"secval\" />";
$outputBuffers[OUTPUT_FORM_END] = "</form>";

// Indexes for dict structure
$DICT_WORD = 0;
$DICT_PTR = 1;

```

```

$outputBuffers[OUTPUT_PAGENUMBERS] .= "<div class=\"result_pages\">\n" . $STR_RESULT_PAGES . " ";
if ($page > 1)
    $outputBuffers[OUTPUT_PAGENUMBERS] .= "<a href=\"\".$SelfURL.$LinkBackJoinChar.\"zoom_query\".$QueryForURL.$metaParams.\"&zoom_page\".$(page-1).
    \"&zoom_per_page\".$per_page.$query_zoom_cats.$dateRangeParams.\"&zoom_and\".$and.\"&zoom_sort\".$sort.\"\" onclick=\"addURL(this)\">lt;lt; " .
    $STR_RESULT_PAGES_PREVIOUS . "</a> ";
for ($i = $start_range; $i <= $end_range; $i++)
    {
        if ($i == $page)
            $outputBuffers[OUTPUT_PAGENUMBERS] .= $page . " ";
        else
            $outputBuffers[OUTPUT_PAGENUMBERS] .= "<a href=\"\".$SelfURL.$LinkBackJoinChar.\"zoom_query\".$QueryForURL.$metaParams.\"&zoom_page\".$(i).
            \"&zoom_per_page\".$per_page.$query_zoom_cats.$dateRangeParams.\"&zoom_and\".$and.\"&zoom_sort\".$sort.\"\" onclick=\"addURL(this)\">\".$i.\"</a> ";
    }
if ($page != $num_pages)
    $outputBuffers[OUTPUT_PAGENUMBERS] .= "<a href=\"\".$SelfURL.$LinkBackJoinChar.\"zoom_query\".$QueryForURL.$metaParams.\"&zoom_page\".$(page+1).
    \"&zoom_per_page\".$per_page.$query_zoom_cats.$dateRangeParams.\"&zoom_and\".$and.\"&zoom_sort\".$sort.\"\" onclick=\"addURL(this)\"> " .
    $STR_RESULT_PAGES_NEXT . " &gt;&gt;</a> ";
$outputBuffers[OUTPUT_PAGENUMBERS] .= "</div>";
}

//Let others know about Zoom.
if ($ZoomInfo == 1)

```