

```
1 <?php
2 error_reporting( E_ALL ^ E_NOTICE );
3
4 // Disabling magic quotes at runtime from php.net
5 if( get_magic_quotes_gpc() )
6 {
7     function stripslashes_deep($value)
8     {
9         $value = is_array($value) ?
10             array_map('stripslashes_deep', $value) :
11             stripslashes($value);
12         return $value;
13     }
14     $_POST = array_map('stripslashes_deep', $_POST);
15     $_GET = array_map('stripslashes_deep', $_GET);
16     $_COOKIE = array_map('stripslashes_deep', $_COOKIE);
17     $_REQUEST = array_map('stripslashes_deep', $_REQUEST);
18 }
19
20
21
22
23
24
25
26
27
28
29 // init - start
30
31 $HTMLParser_CommonCSS = Array(
32     'width','height','font','color','font-family','font-size','font-style','font-weight','letter-spacing',
33     'text-decoration','font-variant','text-indent','text-align','text-transform',
34     'background-color','background-image','background-repeat','background-position','background',
35     'first-line','first-letter',
36     'list-style-image','list-style-type','list-style-position','list-style',
37     'float','clear');
38
39 $HTMLParser_CommonAttr = Array( 'TITLE', 'LANG', 'DIR', 'STYLE' );
40
41 $HTMLParser_AllowTagsAry = Array();
42 $HTMLParser_AllowTagsAry['A'] = new HTMLParser_Tags_Class( 'A', '',
43     array_merge($HTMLParser_CommonAttr,Array('HREF','CHARSET')), $HTMLParser_CommonCSS );
44 $HTMLParser_AllowTagsAry['B'] = new HTMLParser_Tags_Class( 'B', '', $HTMLParser_CommonAttr,
45     $HTMLParser_CommonCSS );
46 $HTMLParser_AllowTagsAry['BDO'] = new HTMLParser_Tags_Class( 'BDO', '', $HTMLParser_CommonAttr,
47     $HTMLParser_CommonCSS );
48 $HTMLParser_AllowTagsAry['BIG'] = new HTMLParser_Tags_Class( 'BIG', '', $HTMLParser_CommonAttr,
49     $HTMLParser_CommonCSS );
50 $HTMLParser_AllowTagsAry['BR'] = new HTMLParser_Tags_Class( 'BR', 'f',
51     array_merge($HTMLParser_CommonAttr,Array('CLEAR')) );
52 $HTMLParser_AllowTagsAry['CITE'] = new HTMLParser_Tags_Class( 'CITE', '', $HTMLParser_CommonAttr,
53     $HTMLParser_CommonCSS );
54 $HTMLParser_AllowTagsAry['CODE'] = new HTMLParser_Tags_Class( 'CODE', '', $HTMLParser_CommonAttr,
55     $HTMLParser_CommonCSS );
56 $HTMLParser_AllowTagsAry['DFN'] = new HTMLParser_Tags_Class( 'DFN', '', $HTMLParser_CommonAttr,
57     $HTMLParser_CommonCSS );
58 $HTMLParser_AllowTagsAry['EM'] = new HTMLParser_Tags_Class( 'EM', '', $HTMLParser_CommonAttr,
59     $HTMLParser_CommonCSS );
60 $HTMLParser_AllowTagsAry['FONT'] = new HTMLParser_Tags_Class( 'FONT', '',
61     array_merge($HTMLParser_CommonAttr,Array('SIZE','COLOR','FACE')), $HTMLParser_CommonCSS );
62 $HTMLParser_AllowTagsAry['I'] = new HTMLParser_Tags_Class( 'I', '', $HTMLParser_CommonAttr,
63     $HTMLParser_CommonCSS );
64 $HTMLParser_AllowTagsAry['IMG'] = new HTMLParser_Tags_Class( 'IMG', 'f',
65     array_merge($HTMLParser_CommonAttr,Array('SRC','ALT','WIDTH','HEIGHT')), $HTMLParser_CommonCSS );
66 $HTMLParser_AllowTagsAry['KBD'] = new HTMLParser_Tags_Class( 'KBD', '', $HTMLParser_CommonAttr,
67     $HTMLParser_CommonCSS );
68 $HTMLParser_AllowTagsAry['Q'] = new HTMLParser_Tags_Class( 'Q', '',
69     array_merge($HTMLParser_CommonAttr,Array('CITE')), $HTMLParser_CommonCSS );
70 $HTMLParser_AllowTagsAry['S'] = new HTMLParser_Tags_Class( 'S', '', $HTMLParser_CommonAttr,
71     $HTMLParser_CommonCSS );
72 $HTMLParser_AllowTagsAry['SAMP'] = new HTMLParser_Tags_Class( 'SAMP', '', $HTMLParser_CommonAttr,
73     $HTMLParser_CommonCSS );
74 $HTMLParser_AllowTagsAry['SMALL'] = new HTMLParser_Tags_Class( 'SMALL', '', $HTMLParser_CommonAttr,
75     $HTMLParser_CommonCSS );
76 $HTMLParser_AllowTagsAry['SPAN'] = new HTMLParser_Tags_Class( 'SPAN', '', $HTMLParser_CommonAttr,
77     $HTMLParser_CommonCSS );
78 $HTMLParser_AllowTagsAry['STRONG'] = new HTMLParser_Tags_Class( 'STRONG', '', $HTMLParser_CommonAttr,
79     $HTMLParser_CommonCSS );
80 $HTMLParser_AllowTagsAry['STRIKE'] = new HTMLParser_Tags_Class( 'STRIKE', '', $HTMLParser_CommonAttr,
81     $HTMLParser_CommonCSS );
82 $HTMLParser_AllowTagsAry['SUB'] = new HTMLParser_Tags_Class( 'SUB', '', $HTMLParser_CommonAttr,
83     $HTMLParser_CommonCSS );
84 $HTMLParser_AllowTagsAry['SUP'] = new HTMLParser_Tags_Class( 'SUP', '', $HTMLParser_CommonAttr,
```

```

$HTMLParser_CommonCSS );
64 $HTMLParser_AllowTagsAry['TEXTAREA'] = new HTMLParser_Tags_Class( 'TEXTAREA', '',
    array_merge($HTMLParser_CommonAttr,Array('DISABLED','READONLY')), $HTMLParser_CommonCSS );
65 $HTMLParser_AllowTagsAry['TT'] = new HTMLParser_Tags_Class( 'TT', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
66 $HTMLParser_AllowTagsAry['U'] = new HTMLParser_Tags_Class( 'U', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
67 $HTMLParser_AllowTagsAry['VAR'] = new HTMLParser_Tags_Class( 'VAR', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
68
69 $HTMLParser_AllowTagsAry['ADDRESS'] = new HTMLParser_Tags_Class( 'ADDRESS', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
70 $HTMLParser_AllowTagsAry['CENTER'] = new HTMLParser_Tags_Class( 'CENTER', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
71 $HTMLParser_AllowTagsAry['DD'] = new HTMLParser_Tags_Class( 'DD', 'o', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
72 $HTMLParser_AllowTagsAry['DIR'] = new HTMLParser_Tags_Class( 'DIR', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
73 $HTMLParser_AllowTagsAry['DIV'] = new HTMLParser_Tags_Class( 'DIV', '',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
74 $HTMLParser_AllowTagsAry['DL'] = new HTMLParser_Tags_Class( 'DL', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
75 $HTMLParser_AllowTagsAry['H1'] = new HTMLParser_Tags_Class( 'H1', '',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
76 $HTMLParser_AllowTagsAry['H2'] = new HTMLParser_Tags_Class( 'H2', '',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
77 $HTMLParser_AllowTagsAry['H3'] = new HTMLParser_Tags_Class( 'H3', '',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
78 $HTMLParser_AllowTagsAry['H4'] = new HTMLParser_Tags_Class( 'H4', '',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
79 $HTMLParser_AllowTagsAry['H5'] = new HTMLParser_Tags_Class( 'H5', '',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
80 $HTMLParser_AllowTagsAry['H6'] = new HTMLParser_Tags_Class( 'H6', '',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
81 $HTMLParser_AllowTagsAry['HR'] = new HTMLParser_Tags_Class( 'HR', 'f',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN','NOSHADE')), $HTMLParser_CommonCSS );
82 $HTMLParser_AllowTagsAry['MARQUEE'] = new HTMLParser_Tags_Class( 'MARQUEE', '',
    array_merge($HTMLParser_CommonAttr,Array('BGCOLOR','DIRECTION','LOOP','SCROLLAMOUNT','SCROLLDELAY')),
    $HTMLParser_CommonCSS );
83 $HTMLParser_AllowTagsAry['MENU'] = new HTMLParser_Tags_Class( 'MENU', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
84 $HTMLParser_AllowTagsAry['OL'] = new HTMLParser_Tags_Class( 'OL', '',
    array_merge($HTMLParser_CommonAttr,Array('TYPE','START')), $HTMLParser_CommonCSS );
85 $HTMLParser_AllowTagsAry['P'] = new HTMLParser_Tags_Class( 'P', 'o',
    array_merge($HTMLParser_CommonAttr,Array('ALIGN')), $HTMLParser_CommonCSS );
86 $HTMLParser_AllowTagsAry['PRE'] = new HTMLParser_Tags_Class( 'PRE', '', $HTMLParser_CommonAttr,
    $HTMLParser_CommonCSS );
87 $HTMLParser_AllowTagsAry['UL'] = new HTMLParser_Tags_Class( 'UL', '',
    array_merge($HTMLParser_CommonAttr,Array('TYPE')), $HTMLParser_CommonCSS );
88 $HTMLParser_AllowTagsAry['LI'] = new HTMLParser_Tags_Class( 'LI', 'o',
    array_merge($HTMLParser_CommonAttr,Array('TYPE','VALUE')), $HTMLParser_CommonCSS );
89 $HTMLParser_AllowTagsAry['DEL'] = new HTMLParser_Tags_Class( 'DEL', '',
    array_merge($HTMLParser_CommonAttr,Array('CITE','DATETIME')), $HTMLParser_CommonCSS );
90 $HTMLParser_AllowTagsAry['INS'] = new HTMLParser_Tags_Class( 'INS', '',
    array_merge($HTMLParser_CommonAttr,Array('CITE','DATETIME')), $HTMLParser_CommonCSS );
91
92 // init - end
93
94
95
96
97
98 // fixed init - start
99
100 $HTMLParser_TagsForREGAry = Array();
101 $HTMLParser_SingleTagAry = Array();
102 $HTMLParser_DoubleTagsAry = Array();
103 while( $obj = current( $HTMLParser_AllowTagsAry ) )
104 {
105     $HTMLParser_TagsForREGAry[] = $obj->nodeName;
106     if( $obj->EndTag == 'f' )
107         $HTMLParser_SingleTagAry[] = $obj->nodeName;
108     else
109         $HTMLParser_DoubleTagsAry[] = $obj->nodeName;
110
111     next( $HTMLParser_AllowTagsAry );
112 }
113 $HTMLTagsForREG = implode( '|', $HTMLParser_TagsForREGAry );
114
115 // fixed init - end
116
117
118
119

```

```
120
121
122
123
124
125 /*
126 All Html Tags can be an object.
127 This class is for init only.
128 Like: <a></a>, <br>, <hr>
129 */
130 Class HTMLParser_Tags_Class
131 {
132     // eg: <a href='abc.htm' style='width:20'>
133     public $nodeName; // nodeName is a.
134     public $EndTag; // EndTag like <a> for </a>, three states: '-'>has end tag, 'o'>optional,
135     // 'f'>false, has no end tag.
136     public $Attr; // href is one attribute in this array.
137     public $CSS; // the element in style attribute.
138
139     public function __construct( $nodeName, $EndTagState, $Attr=Array(), $CSS=Array() )
140     {
141         $this->nodeName = $nodeName;
142         $this->EndTag = $EndTagState;
143         $this->Attr = $Attr;
144         $this->CSS = $CSS;
145     }
146 }
147
148 /*
149 The root of the Parsing Tree
150 */
151 Class HTMLParser_Root_Class
152 {
153     public $nodeName = 'ROOT';
154     public $Depth = 0;
155     public $childNodes = Array();
156 }
157
158 /*
159 The non-text object in the Parsing Tree
160 Like: <a></a>, <br>, <hr>
161 */
162 Class HTMLParser_Other_Class
163 {
164     public $Index;
165     public $nodeName;
166     public $Depth;
167
168     public $AttrAry = Array();
169     public $childNodes = Array();
170
171     public $Parent = NULL;
172     public $Previous = NULL;
173     public $Next = NULL;
174     public $Match = false;
175 }
176
177 /*
178 The text object in the Parsing Tree
179 Like: the 'abc testing' words in <br>abc testing<br>
180 */
181 Class HTMLParser_Text_Class
182 {
183     public $Index;
184     public $nodeName;
185     public $Depth;
186     public $nodeValue;
187
188     public $Parent = NULL;
189     public $Previous = NULL;
190     public $Next = NULL;
191 }
192
193
194
195 /*
196 The main program
197 */
198 Class HTMLParser_Class
199 {
200     /*
201     Written by David NG @ http://dwin.net
202     Copyright(c) 1998-2007 dwin.net all rights reserved
203     */
204 }
```

```

203
204 Start: 2004-08-31 20:16
205 Finish: 2004-09-05 23:48
206 Last Modify: 2004-11-14 13:19
207 */
208 public $DOM; // It stores the object references. Parsing Tree, it has many
209 // nodes, each node may have many child node, and so on.
210 public $DocumentAll = Array(); // It stores the object references. array(div, text(hello
211 // world), hr, text(test1 ), span, text(test2), text(test3) ) for <div>hello world<hr>test1
212 // <span>test2</span>test3</div>
213
214 private $ContainerTreeAry = Array(); // It stores the object references. array(root, div, span)
215 // for <div>hello world<hr>test1 <span>test2</span>test3</div>
216 private $HTMLSourceCode; // the html code you want to parse.
217
218 public function __construct( $SourceCode='' )
219 {
220     global $HTMLTagsForREG;
221     global $HTMLParser_SingleTagAry;
222     global $HTMLParser_DoubleTagsAry;
223
224     $SourceCode = preg_replace( '<script[<]*?>.*?</script[<]*?>/ism' , '' , $SourceCode );
225     $SourceCode = preg_replace( '<!--.*?-->/ism' , '' , $SourceCode );
226     $SourceCode = preg_replace( '/&/' , '&amp;' , $SourceCode );
227     // $SourceCode = preg_replace( '/\[/ ' , '&#91;' , $SourceCode );
228     // $SourceCode = preg_replace( '/]/ ' , '&#93;' , $SourceCode );
229
230     $this->HTMLSourceCode = $SourceCode;
231     $this->DOM = new HTMLParser_Root_Class();
232     $RegMatchs = Array();
233     /*
234     $RegMatchs[1] : value
235     $RegMatchs[2] : begin or end tags
236     $RegMatchs[4] : attribute
237     */
238     preg_match_all( '/(.*?)<\/?((' . $HTMLTagsForREG . '))(\s.*?|)>/ism' , $this->HTMLSourceCode ,
239     $RegMatchs , PREG_OFFSET_CAPTURE );
240
241
242
243
244     $nodeValue = NULL;
245     $nodeName = NULL;
246     $tagName = NULL;
247
248     $Depth = 1;
249     $DA_Index = 0;
250     $REGAry_Index = 0;
251     $this->ContainerTreeAry[] = 'root';
252
253
254     $TagsAry = &$RegMatchs[2];
255     $REGAryLen = count( $TagsAry );
256     $AttrAry = &$RegMatchs[4];
257     $ValuesAry = &$RegMatchs[1];
258
259     $ValuesAry[ $REGAryLen ] = Array();
260     $ValuesAry[ $REGAryLen ][0] = strip_tags( substr( $this->HTMLSourceCode ,
261     $RegMatchs[0][ $REGAryLen-1 ][1] + strlen( $RegMatchs[0][ $REGAryLen-1 ][0] ) ) );
262
263     $NodeValue = strip_tags( $ValuesAry[ $REGAry_Index ][0] );
264     if( $NodeValue != '' )
265     {
266         $this->CreateTextElement( $NodeValue , $DA_Index++ , $Depth );
267         array_shift( $ValuesAry );
268     }
269
270
271     while( true )
272     {
273         $nodeName = strtoupper( $TagsAry[ $REGAry_Index ][0] );
274         $nodeValue = strip_tags( $ValuesAry[ $REGAry_Index ][0] );
275
276         if( in_array( $nodeName , $HTMLParser_DoubleTagsAry ) )
277         {
278             $this->CreateHTMLElement( $nodeName , $DA_Index++ , $Depth++ , false ,
279             $AttrAry[ $REGAry_Index ][0] );
280             if( $nodeValue != '' )

```

Page 5 of 8

440

```

441
442
443
444
445 public function DOMToUBB( &$str ) // UBB? we have the DOM now, you can do whatever you want.
446 {
447     $len = count( $this->DOM->childNodes );
448     for( $i=0; $i<$len; $i++ )
449         $this->DumpFromObj( $this->DOM->childNodes[$i], $str );
450 }
451
452
453
454
455
456 public function DumpFromObj( &$obj, &$str )
457 {
458     global $HTMLParser_DoubleTagsAry;
459     if( $obj->nodeName == 'text' )
460         $str .= $obj->nodeValue;
461     else
462     {
463         if( count( $obj->Attr ) )
464             $Attr = ' '.implode( ' ', $obj->Attr );
465         else
466             $Attr = '';
467
468         if( count( $obj->CSS ) )
469             $CSS = ' STYLE="'.implode( ';', $obj->CSS ).'";';
470         else
471             $CSS = '';
472
473         $str .= '<'. $obj->nodeName . $Attr . $CSS . '>';
474
475         $len = count( $obj->childNodes );
476         for( $i=0; $i<$len; $i++ )
477             $this->DumpFromObj( $obj->childNodes[$i], $str );
478         if( in_array( $obj->nodeName, $HTMLParser_DoubleTagsAry ) )
479             $str .= '</'. $obj->nodeName . '>';
480     }
481 }
482 }
483
484
485
486
487
488 ?><META Http-Equiv="Content-Type" Content="text/html; charset=utf-8">
489 <title>dwin.net - web application - html parser (php5)</title>
490 Written by David NG @ <a href="http://dwin.net">http://dwin.net</a><BR>
491 Copyright(c) 1998-2004 dwi.net all rights reserved<BR><BR>
492
493 <a href="html_parser_src_php5.zip">download - html parser (php5) - source code</a><BR>
494
495 <font color=blue><xmp><?php
496
497
498
499
500
501 $Content = $_POST['Content'];
502 if( isset( $Content ) )
503 {
504     $DOM = new HTMLParser_Class( $Content );
505
506     $Str = '';
507     $DOM->DOMToUBB( $Str );
508     echo $Str;
509 }
510
511
512
513
514
515 ?></xmp></font>
516 <form method="post" >
517 <textarea name="Content" style="width:500;height:100"><a disabled title=" hello
518 world" style="width:200; height:expression(200);background:red" 4534 CHARSET = en href="#" dir = 'rtl'
519 disabled >gogo</a><p style='background: url("images/aardvark.gif"); background-repeat: repeat-y; '
520 title=good size=5 style='width:20;height:50;'>hi <b color="red">test<b>
521 how <a href="javascript:http:dwin.net" border=1>are<font color=red>you</a>
522 ?</hello>
523 tell </p>me
524 <big>

```

```
523     you <div>love<span> me</span>
524 </big>end</textarea><BR>
525 <input type="submit">
526 </form>
```