

```
1  #!/usr/bin/perl
2  #!d:/perl/bin/perl.exe
3
4  $strUrlBase = '<base href="http://deskpro/LinZhou/" />';
5  $strUrlBase = '<base href="http://perlclass.misterperl.com/LinZhou/" />';
6
7  #####
8  #
9  # ProgramName: Solt Machine ©1998-2001, Lin Zhou
10 # Description: Perl scripts, as a CGI, to emulate a slot machine, but with fun!
11 #
12 #     functions list:
13 #         &DecodeForm, &evalprompt($), &slot_debug(@)
14 #
15 #     variables list -
16 #     :calc: $forminput, $buffer, $name, $value,
17 #             $maxpoints, $start, $ratio, $failed, $times, $numA, $numB, $numC,
18 #             $points, $coins, $invest, $earned, $codes,
19 #     :html: $coA, $coB, $coC, $bgA, $bgB, $bgC, $disprompt, $discoins, $units,
20 #             $scroll, $onload, @image, @color, $colorOnceMore,
21 #             $reset, $addto, $play, $action, $disable,
22 #             $sounds, $snd_music, $snd_start, $snd_click, $snd_mover,
23 #     :disp: $prompt, $message, %messages, %tips, @bonus,
24 #
25 #####
26
27 ### use Term::ReadKey; ($wchar, $hchar, $wpixels, $hpixels) = GetTerminalSize();
28
29 print "Content-type: text/html\n\n";
30
31 ### Begins Decoding the Data ###
32
33     sub DecodeForm;     ## prototype, decode form input names and values and save in %forminput
34
35     %forminput = &DecodeForm;
36
37 ### End of Decoding the Data ###
38
39 ### Initialization, before import the form input values, for every new page ####
40
41     sub evalprompt($); ## prototype, function evals prompt string from %prompts to $prompt
42     sub slot_debug(@); ## prototype, function evals strings to variables and print out
43
44     %messages = (
45         "start", "Win ME, Win e<u>X</u>tra <u>P</u>oints: Welcome to join!",
46         "great", "Congratulations! You Win A Great Deal.",
47         "vgood", "Good Fortune! Let's Try Mr. Lucky Seven Next...",
48         "seven", "How do you like her? Click it to have a look...",
49         "worth", "Hello, Mr. Lucky Seven! You Earned A Worth.",
50         "weird", "Pretty lucky today? See fortune smiled on you?",
51         "magic", "Cool...! You will become a millionaire soon!",
52         "lucky", "Would these be your lucky numbers?",
53         "cross", "Hah, some loss... Did you cross your fingers?",
54         "nobig", "Not a big deal... Let's beat it up!",
55         "retry", "A Nice Try. Just Don't Give Up!",
56         "cheer", "Cheer up! Get more money and win it back!",
57         "clues", "Please add more coins to keep your investment.",
58         "hints", "Hint: More coins, and more points to win more!",
59         "2wins", "We do have a win-win strategy.",
60         "greet", "Ehhh o~oh... soo..., how are you doing today?",
61     );
62
63     %tips = (
64         "ctrla", "Wish to win? Press Ctrl+A to see your secret code.",
65         "aleft", "Press Alt+Left to browse one page back, if you really hate the result...",
66         "dream", "Ever dreamed to win? Hah, not with us! (Ask yourself who wrote it...)",
67         "smart", "We are glad you made a smart choice - to fill up our moneybag!",
68         "twin7", "One of a hundred chances to get a twin. Am I right, or not?",
69         "warns", "Not enough yet? You would lose all you have!",
70         "whati", "What did I said?! You could have left with fortune ...",
71         "codes", "Did you get a lower secret code on this bet?",
72         "fresh", "Tips: Did you know pressing F5 can refresh the page?",
73         "never", "Tips: Never use your own money! We mean it.",
74         "might", "Tips: Might be a good time to take money back as you can ...",
75         "music", "Tips: No music? Check with your browser settings and enable JavaScript.",
76         "pages", "Tips: Browse page back to take a repentant medicine.",
77         "plays", "Tips: Best play on a full screen of 800x600 or up. (Press F11, F5)",
78     );
79
80     $maxpoints = 90000000;
81
82     $ratio = 100;     ## decide how fast you will earn or lose: larger for faster.
83     $start = 10;     ## coins presented as premiere bonus only used the 1st time.
```

```
81
82 $earned = $coins = $points = $invest = $failed = 0;
83
84 $numA = $numB = $numC = 0;
85
86 @bonus = ( 1000000, 50000, 30000, 20000, 17500, 15000, 12500, 10000,    ## 0~7,  earn points
87           -8000, -7500, -7000, -6500, -6000, -5500, -5000, -4500,    ## 8~15, lose points
88 );
89 @color = ("red", "teal", "navy", "chocolate", "hotpink", "skyblue", "magenta", "salmon", "lime");
90 @image = ('./images/slot_face.gif',      ## "00", "777"
91           './images/slot_banana.gif',    ## "11", "666", "888"
92           './images/slot_cherry.gif',    ## "22", "555", "999"
93           './images/slot_grapes.gif',    ## "33", "444"
94           './images/slot_lemon.gif',     ## "44", "333"
95           './images/slot_orange.gif',    ## "55", "222"
96           './images/slot_strawberry.gif', ## "66", "111"
97           './images/slot_bar.gif',      ## "7x", "000"
98           './images/slot_watermelon.gif', ## "88",
99           './images/slot_winme.jpg',     ## "99", "77x"
100 );
101
102 $snd_music = "./midi/labamba.mid";      $snd_winme = "./midi/musica.wav";
103 $snd_start = "./midi/chimes.wav";
104 $snd_click = "./midi/click.wav";
105 $snd_mover = "./midi/mover.wav";
106
107 $scroll = "no"; $onload = "";
108 $reset = "hidden"; $colorOnceMore = "seagreen";
109 $bgA = $bgB = $bgC = ' ';
110
111
112 ### Getting the form input values, from previous running #####
113 #
114 # $value = $forminput{'name'}; ## $value from %forminput ##
115 #
116 $times = ($forminput{'times'}) ? ($forminput{'times'}) : 0;
117 $codes = ($forminput{'codes'}) ? ($forminput{'codes'}) : 0;
118 $coins = ($forminput{'coins'}) ? ($forminput{'coins'}) : 0;
119 $addto = ($forminput{'addto'});
120
121 $invest = $forminput{'invest'} + $coins;
122 $points = $forminput{'points'} + ($coins * $ratio);
123 $earned = $forminput{'earned'};
124
125 # test!
126 # &slot_debug("coins points invest earned codes times"); # test to see if got the values.
127 # test!
128
129 #
130 ### initial color of numbers, make all deferent and random: #####
131 #
132 do { $coA = $color[(int(rand(7))+1)]; } until ($coA ne "");
133 do { $coB = $color[(int(rand(7))+1)]; } until ($coB ne $coA);
134 do { $coC = $color[(int(rand(7))+1)]; } until ($coC ne $coB) && ($coC ne $coA);
135 #
136 if ($forminput{'submit'} eq "Reset") { # clear points to restart if "Reset"
137     $points = 0;
138     $onload = "history.go(-$times);";
139     $reset = "hidden";
140     $addto = "none";
141 }
142 $times++;
143
144 if ($addto eq "Add Coins") { # show input if "Add Coins"
145     $reset = "submit";
146     $disprompt = "To add";
147     $units = "coins";
148     $discoins = "text";
149     $dispoints = "hidden";
150     $addto = "none";
151     $disable = "DISABLED";
152     $play = "submit";
153     $action = "Continue";
154     if ($invest != 0) {
155         $prompt = evalprompt("hints");
156         $message = $messages{'clues'};
157     } else {
158         $prompt = evalprompt("warns");
159         $message = $messages{'hints'};
160     }
161 }
```

```
161     $onload= "self.status='$tips{'smart'}';";
162 }
163 elsif ($coins == 0 && $points == 0) { # if starting a new game
164     $disprompt = "To bet";
165     $units = "coins";
166     $discoins = "text";
167     $dispoints = "hidden";
168     $addto = "none";
169     $disable = "DISABLED";
170     $reset = "hidden";
171     $play = "submit";
172     $action = "Start Play";
173     $points = $start * 100;
174     $earned = int($points / 100);
175     $prompt = evalprompt("bonus");
176     $message = $messages{'start'};
177     $colorOnceMore = "darkred";
178     if ($times == 1) {
179         $sounds = $snd_start;
180         if(!int(rand(3))) {
181             $onload = "self.status='$tips{'never'}';";
182         } else {
183             $onload = "self.status='$tips{'music'}';";
184         }
185     }
186 }
187 else { # else playing the game
188     $numA = int(rand(10));
189     $numB = int(rand(10));
190     $numC = int(rand(10));
191 ##
192 ## secret: a number set for enough times to win!
193 ## set default $codes=1000000 if you are a shareholder of this slot machine :)
194 ##
195     $codes = 10; # by default, ten times to win!
196     while (($codes <= 2) || ($coins > 0)) {
197         $codes += int(rand(10) + 1);
198         $coins = int($coins/10);
199     }
200     $numA = $numB = $numC if (!(($times % $codes));
201     local $comes = abs($times % $codes - $codes);
202
203     $secret = "<br /><font class='secret'>Your secret code is $codes on this bet;
204             $comes times to win!<br />Better have the status bar displayed.</font>";
205
206     if(!int(rand(2))) {
207         $onload = "self.status='$tips{'plays'}'";
208         $onload = "self.status='$tips{'fresh'}'" if (!(($times % ($codes+1)));
209     } else {
210         $onload = "self.status='$tips{'ctrla'}'";
211         $onload = "self.status='$tips{'codes'}'" if ($codes <= 10);
212     }
213 ##
214 ## end secret.
215 ##
216     if (($codes <= 2) || (!(($times % 4))) {
217         $onload = "self.status='$tips{'dream'}'";
218     } elsif (!(($times % 7)) {
219         $onload = "self.status='$tips{'aleft'}'";
220     }
221     $coins = 0;
222
223     if (($numA == $numB) && ($numB == $numC)) { ## got "nnn"
224         local $index = abs(7 - $numA);
225         local $lucky = (defined($codes) && $codes <= numA && (int(rand($numA))));
226         $bgA = $bgB = $bgC = $image[$index];
227         $coA = $coB = $coC = $color[$index];
228         $points += $bonus[$index];
229         if ($numA == 7) { ## got "777"
230             $message = $messages{'great'};
231             $message = $messages{'weird'} if($lucky);
232             $snd_music = $snd_winme;
233         } else {
234             $message = $messages{'vgood'}; ## got "nnn", but n !=7
235             $message = $messages{'magic'} if($lucky);
236         }
237         $onload= "self.status='$tips{'might'}'";
238     }
239     elsif ($numA == 7 || $numB == 7 || $numC == 7) { ## got one or two 7's
240         local $index = 7;
```

```
241     ++$index if ($numA == 7);
242     ++$index if ($numB == 7);
243     ++$index if ($numC == 7); $index = 7 if ($index < $#image);
244     if ($numA == 7) {
245         $points += int($bonus[7]/$ratio);
246         $bgA = $image[$index];
247         $coA = $color[0];
248     }
249     if ($numB == 7) {
250         $points += int($bonus[7]/$ratio);
251         $bgB = $image[$index];
252         $coB = $color[0];
253     }
254     if ($numC == 7) {
255         $points += int($bonus[7]/$ratio);
256         $bgC = $image[$index];
257         $coC = $color[0];
258     }
259     if ($index == 7) {                                     ## got one 7
260         $message = $messages{'worth'};
261         goto same2;
262     } else {                                             ## got two 7's
263         $message = $messages{'seven'};
264     }
265     $onload= "self.status='$tips{'twin7'}'";
266 }
267 elseif (($numA==$numB)||($numB==$numC)||($numC==$numA)) { ## got two of the same
268     $points += int($bonus[7]/$ratio);
269 same2: $bgA = $bgB = $image[$numA] if ($numA==$numB);
270        $bgB = $bgC = $image[$numB] if ($numB==$numC);
271        $bgC = $bgA = $image[$numC] if ($numC==$numA);
272        $onload= "self.status='$tips{'warns'}'";
273 }
274 else {                                                 ## not match: lose points
275     local $index = int(rand(7)) + 1;
276     $points -= int($points/2) if ($invest == 0);
277     $points += int($bonus[$index+7] / $maxpoints * $points * $ratio * $times);
278     $points = 0 if ($points < 10);
279     $failed = 1;
280 }
281
282 $disprompt = "You have";                               ## post-preprocessing to display the prompt
283
284 if ($points <= 0) {                                     ## lost all points
285     $earned = $points = 0;
286     if ($invest == 0) {
287         $prompt = evalprompt("nobonus");
288     } else {
289         $prompt = evalprompt("alllost");
290     }
291     $units = "points left";
292     $message = $messages{'cheer'};
293     $disable = "DISABLED";
294     $play = "hidden";
295     if (int(rand(2))) {
296         $onload = "self.status='$tips{'whati'}'";
297     }
298     else {
299         $onload = "self.status='$tips{'pages'}'";
300     }
301 }
302 else {
303     if ($points < ($start * $ratio / 2)) {             ## have some points left
304         $addto = "addto";                             ## need more coins
305         $disable = "";
306     } else {
307         $addto = "none";
308         $disable = "DISABLED";
309     }
310     $points = $maxpoints if ($points >= $maxpoints);
311     $earned = (int($points / 100)) - $invest;
312     $prompt = evalprompt("total");
313
314     if ($earned >= 0) {                                 ## earn, so far
315         $prompt .= "earned";
316         $units = "points";
317         local $random = int(rand(3));
318         if ($failed) {                                  ## lost, this time
319             if ($random == 0) {
320                 $message = $messages{'greet'};

```

```
321         } elsif ($random == 1) {
322             $message = $messages{'cross'};
323         } else {
324             $message = $messages{'nobig'};
325         }
326     } elsif (!defined($message)) {          ## earn, this time, and no $message
327         if ($random) {
328             $message = $messages{'2wins'};
329         } else {
330             $message = $messages{'lucky'};
331         }
332     }
333 } else {                                     ## lost, so far, but might earn this time
334     $units = "points left";
335     $message = $messages{'retry'} if (!defined($message));
336     $prompt .= "<font color='red'>lost</font>";
337 }
338
339 if ($invest != 0) {
340     $prompt .= evalprompt("coins");
341 }
342
343 $play = "submit";
344 $action = "Once More";
345 }
346
347 $dispoints = "text";
348 $discoins = "hidden";
349 $reset = "submit";
350 }
351
352
353 ### Creating the HTML page #####
354
355 print <<EndOfHTML;
356
357 <html>
358 <head><title>Slot Machine &copy;1998-2001, Lin Zhou</title>
359
360     $strUrlBase<!--         base href address         -->
361
362 <style type=text/css><!--
363
364 body {
365     margin: 5% 1%; padding: 0; cursor: wait;
366     background: black; font: 14pt/150% "Times New Roman";
367 }
368 br, p, table {
369     margin: 0; padding: 0;
370 }
371 h1, h2 {
372     margin: 0px; padding: 5px 0px 20px 0px; background: #fffaf0;
373     border-width: 0px; border-style: solid;
374     border: 10 solid; border-color: #ccff99 #338800 #336600 #ccffbb;
375     font: bold 32px/32px "Arial","palatino", "bookman", "book antiqua", "times new roman";
376     line-height: 40px; width: 664px; text-align: center; vertical-align: middle;
377     color: purple; background-color: ivory;
378 }
379 font.message { color: orange; font: bold 11pt/12pt "times new roman"; }
380 font.prompt { color: orchid; font: bold 11pt/11pt "times new roman"; }
381 font.number { color: green; }
382 font.secret { color: black; font: bold 10pt/12pt "times new roman"; }
383
384 td { background: transparent none no-repeat fixed center;
385     margin: 0px; padding: 0px;
386 }
387 td.slota {
388     background: lightpink url("$bgA") repeat fixed center; color: $coA;
389     border-top: 10px; border-right: 10px; border-bottom: 10px; border-left: 10px;
390     border-style: outset; border-color: pink;
391     border: 10px outset; text-align: center; vertical-align: middle;
392     font: bold 240pt/200pt Arial; width: 200px; height: 100px;
393 }
394 td.slotb {
395     background: palegreen url("$bgB") repeat fixed center; color: $coB;
396     border-top: 10px; border-right: 10px; border-bottom: 10px; border-left: 10px;
397     border-style: outset; border-color: pink;
398     border: 10 outset; text-align: center; vertical-align: middle;
399     font: bold 240pt/200pt Arial; width: 200px; height: 100px;
400 }
```

```
401 td.slotc {
402     background: khaki url("$bgC") repeat fixed center; color: $coC;
403     border-top: 10px; border-right: 10px; border-bottom: 10px; border-left: 10px;
404     border-style: outset; border-color: pink;
405     border: 10 outset; text-align: center; vertical-align: middle;
406     font: bold 240pt/200pt Arial; width: 200px; height: 100px;
407 }
408 td.data {
409     font: bold 15pt/15pt Arial; color: blue; border: 0;
410 }
411 .copyright { color: dimgray; font: bold 8pt Arial; text-align: right; }
412 input { height: 20pt; vertical-align: middle; font: bold 12pt/12pt Arial; }
413 input.button { height: 20pt; color: $colorOnceMore; }
414 input.disable { height: 20pt; color: honeydew; }
415 input.points { text-align: center; padding: 4; color: greenyellow; background: linen; }
416 input.input { text-align: left; padding: 4; }
417 input.reset { height: 20pt; color: deeppink; }
418 //--></style>
419
420 <script language="JavaScript"><!-- // manipulates sound on mouse actions
421     function snd_mover() {
422         document.all.id_mover.src="$snd_mover";
423     }
424     function snd_mdown() {
425         document.all.id_mdown.src="$snd_click";
426     }
427 //--></script>
428
429 </head>
430
431 <body scroll="$scroll" onload="$onload; document.body.style.cursor='default';"><center>
432 <!-- ## background sound and music, manipulated by JavaScript functions #####
433 -->
434 <bgsound id='id_sound' volume='-1500' loop='1' src="$sounds">
435 <bgsound id="id_music" volume="-1850" loop="-1" src="$snd_music">
436 <bgsound id="id_mdown" volume="-1500" loop="1" src="">
437 <bgsound id="id_mover" volume="-100" loop="1" src="">
438
439 <table border="0" width="100%" height="100%"><tr><td>
440 <form action="$ENV{'SCRIPT_NAME'}" method="post"><center>
441
442 <table><tr>
443 <td colspan="3" align="center" valign="middle">
444 <table><tr><td align="center">
445
446 <h1>
447 <span style="position:relative;left:0px;top:0px;z-index:10;visibility:visible;">
448 <font color="silver">Win Me, Win eXtra Points<sup>&reg;</sup>!</font>
449 <span style="position:absolute;left:-1px;top:-1px;z-index:100;visibility:visible;">
450 <font color="hotpink">Win Me,</font>
451 <font color="blue">Win</font>
452 e<font color="blue">X</font>tra
453 <font color="blue">P</font>oints<sup>&reg;</sup>!
454 </span>
455 </span>
456 </h1>
457
458 </td></tr></table>
459 <table border="0">
460 <tr><td class="slotA" onClick="this.innerHTML='&nbsp;';"
461 onMouseOut="this.innerHTML=$numA">$numA</td>
462 <td class="slotB" onClick="this.innerHTML='&nbsp;';" onMouseOut="this.innerHTML=$numB">$numB</td>
463 <td class="slotC" onClick="this.innerHTML='&nbsp;';" onMouseOut="this.innerHTML=$numC">$numC</td>
464 </tr>
465 </table>
466 </td></tr>
467 <tr><td class="data" width=50% height="50" align="left" valign="top"><hr>
468 <font class="prompt">
469 $prompt
470 </font>
471 </td><td class="data" width=50% height="50" align="right" valign="top"><hr>
472 
473 </td></tr>
474 <tr><td class="data" width="50%" align="left" valign="top">
475 $disprompt
476 <input class="input" type="$discoins" name="coins" value="$coins" size="10" maxlength="5">
477 <input class="points" type="$dispoints" name="points" value="$points" size="10" DISABLED>
478 $units ...<br /><br />
479 <font class="message">
480 $message
```

```
481     </font>
482 </td><td class="data" width="50%" align="right" valign="top">
483     <input class="reset" type="$reset" name="$reset" value="Reset"
484         onFocus="this.blur();" onMouseOver="snd_mover()" onMouseDown="snd_mdown()">
485     <input class="button" type="$play" name="play" value="$action"
486         onFocus="this.blur();" onMouseOver="snd_mover()" onMouseDown="snd_mdown()">
487     <input class="button" type="submit" name="$addto" value="Add Coins"
488         onFocus="this.blur();" onMouseOver="snd_mover()" onMouseDown="snd_mdown()" $disable>
489     $secret
490     <input type="hidden" name="times" value="$times">
491     <input type="hidden" name="codes" value="$codes">
492
493     <input type="hidden" name="earned" value="$earned">
494     <input type="hidden" name="invest" value="$invest">
495     <input type="hidden" name="points" value="$points">
496 </td></tr>
497 </table>
```

```
499 <p class="copyright">&copy; 1998-2001, Lin Zhou &nbsp;  </p>
500 </form>
501 </table>
502 </center>
503 </body>
504 </html>
```

505 EndOfHTML

509 ### sub to eval prompt hashes #####

```
511 sub evalprompt($) {
512     local $value = abs($earned);
513     %prompts= (
514         "bonus", "You have $points premiere bonus. ",
515         "total", "Totally <font color='lightyellow'>$value</font> coins ",
516         "coins", " with <font class='number'>$invest</font> invested. ",
517         "hints", "$invest coins invested but $points points left ",
518         "warns", "$points points left - You're using up the bonus ",
519         "nobonus", "Sorry, your $start bonus coins ran out. ",
520         "alllost", "Sorry, all $invest coins lost. ",
521     );
522     return $prompts{"$_[0]"};
523 }
524 }
```

526 ### end sub evalprompt #####

529 ### sub to debug any \$scalar, @list, and %hash #####

```
531 sub slot_debug(@) {
532     #
533     print "<p>$#_ arguments received \n";
534     print "\n<p>";
535     foreach (@_) {
536         foreach (split(/ /, $_)) {
537             local $space = " ";
538             print "\n<p>";
539             #
540             if (defined eval("\$_$_")) {
541                 print "\$_$_=";
542                 print eval("\$_$_");
543                 print $space;
544             }
545             elsif (defined eval("@$_$_")) {
546                 local @list = ("");
547                 eval("@list=@$_$_");
548                 #
549                 print "(@):";
550                 print "@list";
551                 print "[list]\n";
552                 print $space;
553             }
554             elsif (defined eval("%$_$_")) {
555                 local %hash = ("");
556                 eval("%hash=%$_$_");
557                 #
558                 print "(%):";
559                 print %hash;
560                 print "[hash]\n";
561                 print $space;
562             }
563         }
564     }
565 }
```

```
561             print "$_ ";
562             print $space;
563         }
564     #         print " |\n</p> ";
565     }
566     }
567     print " ;\n<p />";
568 }
569
570 ### end sub slot_debug #####
571
572
573 ### sub to decode form input data #####
574
575 sub DecodeForm {
576     my ($buffer, $name, $value, %forminput);
577     #
578     # test!
579     #     print "$_=$ENV{$_}\n <br />" foreach (sort keys %ENV);     # test environment var's;
580     # test!
581     #
582
583     =pod    alt_form_input_reader
584     my ($env_content_length, $env_query_string);
585     $env_content_length = $ENV{"CONTENT_LENGTH"};
586     $env_query_string   = $ENV{"QUERY_STRING"};
587     if($env_content_length > 0) {
588         # read POST data from STDIN
589         read STDIN, $buffer, $env_content_length;
590         # add in the GET data
591         $buffer .= "&".$env_query_string if length $env_query_string;
592     } else {
593         # read only the GET data
594         $buffer = $env_query_string;
595     }
596     =cut    alt_form_input_reader
597
598     $ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
599     if($ENV{'REQUEST_METHOD'} eq "GET") {
600         $buffer = $ENV{'QUERY_STRING'};
601     } elsif($ENV{'REQUEST_METHOD'} eq "POST") {
602         read STDIN, $buffer, $ENV{'CONTENT_LENGTH'};
603     } else {
604         ### for command-line debug:
605         ### arguments used as name1=value1&name2=value2&...
606         $buffer = (grep(!/^-/ , @ARGV))[0];     ### from the first argument without a switch (-)
607         $buffer =~ s/\\&/&/g;
608     }
609     #
610     # test!
611     #     print "$buffer \n";     # test what it read from the form input;
612     # test!
613
614     foreach (split(/&/ , $buffer)) {
615         # separate the names and values
616         ($name, $value) = split(/=/ , $_);
617
618         # translate '+' to spaces
619         $name =~ tr/+// ;
620         $name =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
621
622         # translate %xx codes to characters
623         $value =~ tr/+// ;
624         $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
625
626         # remove potentially dangerous commands
627         $value =~ s/<!--(.\n)*-->//g;
628         $value =~ s/<([\^]|\\n)*>//g;
629
630         $forminput{$name} = $value;
631     }
632     #
633     # test!
634     #     print "<p>$_=$forminput{$_}\n" foreach (sort keys %forminput);     # test decoded results;
635     # test!
636     #
637     %forminput;
638 }
639 ### end sub DecodeForm #####
640
```