

2007年 1月 1日 01:49	FormatString.class.php	Page 1
1	<?php /* vim: set fdm=marker: */	
2	/** 日記に入力された文字列を表示するときに、	
3	* 入力規則に従って、整形するクラス。	
4	*/	
5	class FormatString {	
6	var \$string;	
7	/** コンストラクタ	
8	*/	
9	function FormatString(\$str) {	
10	\$this->string = \$str;	
11	}	
12		
13	/** stringの値を返す。	
14	* @return string 現在の整形データ	
15	*/	
16	function getString() {/**{{{*/	
17	return \$this->string;	
18	}	
19	/**}}}}*/	
20		
21	/** stringの値を入力する。	
22	* @param \$str 文字列	
23	*/	
24	function putString(\$str) {/**{{{*/	
25	\$this->string = \$str;	
26	}	
27	/**}}}}*/	
28		
29	/** PHP5からは、コンストラクタは、__constructというメソッドで、	
30	* 定義される。よって、将来対策。	
31	*/	
32	function __construct(\$str) {	
33	\$this->FormatString(\$str);	
34	}	
35		
36	/** + 対策.*/	
37	function formatOL() {	
38	/**{{{*/	
39	//データは、\$nで区切られていると思ってよい。	
40	\$array = explode(" \$n", \$this->string);	
41	\$ret = ""; //変換された後のデータ。	
42	\$isInOL = false; //OL状態にあったのか、そうでないのか。	
43	for (\$iCounter = 0; \$iCounter < count(\$array); \$iCounter++) {	
44	\$temp = \$array[\$iCounter]. " \$n";	
45	if (\$temp[0] == "+") {	
46	//OLの対称である。今はじめて+に出くわしたのか、前の行も+があったんかをチェックする	
	必要あり。	
47	if (\$isInOL == false) {	
48	//初めてOLを使うので、+を \$n\$1\$1<i />に置換すればOK	
49	// \$temp = ereg_replace(" [^\$+]", " \$n\$1\$1<i />", \$temp);	
50	\$temp = ereg_replace(" [/^\$+]", " \$n<i />", \$temp);	
51	\$isInOL = true;	
52	} else {	
53	// の中にある.+を<i />に置換すればOK	

2007年 1月 1日 01:49	FormatString.class.php	Page 2
54	\$temp = preg_replace("[^\$+]", "", \$temp);	
55	}	
56	} else {	
57	//OLの対称ではないが、+状態から、抜け出たのか、何も関係ないのかをチェックする 必要あり。	
58	if (\$isInOL == false) {	
59	//もともと何にも関係ないので、そのままスルー	
60	} else {	
61	//+状態から抜け出したので、末尾に、 を加える必要あり。	
62	\$temp .= " ";	
63	\$isInOL = false;	
64	}	
65	}	
66	\$ret .= \$temp;	
67	}	
68	\$this->string = \$ret;	
69	}	
70	/**}}}	
71		
72	/** - 対策 */	
73	function formatUL () {	
74	/**{{{*/	
75	//データは、 で区切られていると思ってよい。	
76	\$array = explode(" ", \$this->string);	
77	\$ret = ""; //変換された後のデータ。	
78	\$isInUL = false; //OL状態にあったのか、そうでないのか。	
79	for (\$iCounter = 0; \$iCounter < count(\$array); \$iCounter++) {	
80	\$temp = \$array[\$iCounter]. " ";	
81	if (\$temp[0] == "-") {	
82	//ULの対称である。今はじめて+に出くわしたのか、前の行も+があったんかをチェックする 必要あり。	
83	if (\$isInUL == false) {	
84	//初めてULを使うので、+を に置き換えればOK -にヒットさせる場合は、 文末におく。	
85	\$temp = preg_replace("[^\$-]", " ", \$temp);	
86	\$isInUL = true;	
87	} else {	
88	// の中にある.+をに置き換えればOK	
89	\$temp = preg_replace("[^\$-]", "", \$temp);	
90	}	
91	} else {	
92	//ULの対称ではないが、+状態から、抜け出たのか、何も関係ないのかをチェックする 必要あり。	
93	if (\$isInUL == false) {	
94	//もともと何にも関係ないので、そのままスルー	
95	} else {	
96	//+状態から抜け出したので、末尾に、 を加える必要あり。	
97	\$temp .= " ";	
98	\$isInUL = false;	
99	}	
100	}	
101	\$ret .= \$temp;	
102	}	
103	\$this->string = \$ret;	