

2007年 1月 1日 01:49	FormatString.class.php	Page 3
104	} /**)}}*/ /** [http://URL:youtube] 対策 * @param \$sys "PC"->計算機 "MOBILE"->携帯 * */ function formatYouTube (\$sys) { /**{{{*/ \$pattern_youtube = '/*[(https? ftp)(://[_!@*~a-zA-Z0-9-;?&@=&\$%#]+) (:youtube)*/'; \$ary = explode("¥n", \$this->string); //print_r (\$ary); \$ret = ""; for (\$iCounter = 0; \$iCounter < count (\$ary); \$iCounter++) { if (preg_match (\$pattern_youtube, \$ary[\$iCounter], \$array)) { \$url = \$array[1].\$array[2]; \$url = str_replace("?", "/", \$url); \$url = str_replace("=", "/", \$url); \$replace = "<object width=¥\"425¥\" height=¥\"350¥\">¥n¥t<param name=¥\"movie¥\" value=¥\".\$url.¥\"></param><param name=¥\"wmode¥\" value=¥\"transparent¥\"></param><embed src=¥\".\$url.¥\" type=¥\"application/x-shockwave-flash¥\" wmode=¥\"transparent¥\" width=¥\"425¥\" height=¥\"350¥\"></embed></object>"; if (\$sys == "PC") { \$ret .= preg_replace (\$pattern_youtube, \$replace, \$ary[\$iCounter]); } else if (\$sys == "MOBILE") { \$ret .= preg_replace (\$pattern_youtube, " [YouTube動画]", \$ary[\$iCounter]); } else { \$ret .= \$ary[\$iCounter]; \$ret .= "¥n"; } } \$this->string = \$ret; } /** }}}*/ /** [TeXの数式:tex] 対策. */ function formatTeX() [/**]]*/ \$pattern_tex = '/*[(.+)(:tex)*/'; \$ary = explode("¥n", \$this->string); //print_r (\$ary); \$ret = ""; for (\$iCounter = 0; \$iCounter < count (\$ary); \$iCounter++) { if (preg_match (\$pattern_tex, \$ary[\$iCounter], \$array)) { // print_r (\$array); \$texGif = ""; \$texGif = ""; \$ret.=preg_replace (\$pattern_tex, \$texGif, \$ary[\$iCounter]); } else { \$ret .= \$ary[\$iCounter]; } } }	104

2007年 1月 1日 01:49	FormatString.class.php	Page 4
153	}	
154	\$this->string = \$ret;	
155	}	
156	/**}}*/	
157		
158	/** *H3対策 */	
159	function formatH3 () {	
160	/** {{{*/	
161	\$array = explode(" \n", \$this->string);	
162	// \$array = explode(" \n", \$this->string);	
163	\$buffer = "";	
164	// print_r (\$array);	
165	\$pattern_H3 = ' / (^ *) ([^ \n] ^ *) ([\r] ^ \n) * \$ / ';	
166	for (\$iCounter = 0; \$iCounter < count (\$array); \$iCounter++) {	
167	if (preg_match (\$pattern_H3, \$array [\$iCounter], \$xArray)) {	
168	// print \$array [\$iCounter];	
169	\$buffer .= " ". \$xArray [2]. " ". "\n";	
170	} else {	
171	\$buffer .= \$array [\$iCounter]. "\n";	
172	}	
173	}	
174		
175	\$this->string = \$buffer;	
176	}	
177	/**}}*/	
178		
179	/** *H4対策 */	
180	function formatH4 () {	
181	/** {{{*/	
182	\$array = explode(" \n", \$this->string);	
183	\$buffer = "";	
184	\$pattern_H4 = ' / (^ *) ([^ \n] ^ *) ([\r] ^ \n) * \$ / ';	
185	for (\$iCounter = 0; \$iCounter < count (\$array); \$iCounter++) {	
186	if (preg_match (\$pattern_H4, \$array [\$iCounter], \$xArray)) {	
187	\$buffer .= " ". \$xArray [2]. " ". "\n";	
188	} else {	
189	\$buffer .= \$array [\$iCounter]. "\n";	
190	}	
191	}	
192	\$this->string = \$buffer;	
193	/**}}*/	
194	}	
195		
196	/** *H5対策 */	
197	function formatH5 () {	
198	/** {{{*/	
199	\$array = explode(" \n", \$this->string);	
200	\$buffer = "";	
201	\$pattern_H5 = ' / (^ *) ([^ \n] ^ *) ([\r] ^ \n) * \$ / ';	
202	for (\$iCounter = 0; \$iCounter < count (\$array); \$iCounter++) {	
203	if (preg_match (\$pattern_H5, \$array [\$iCounter], \$xArray)) {	
204	\$buffer .= " ". \$xArray [2]. " ". "\n";	
205	} else {	
206	\$buffer .= \$array [\$iCounter]. "\n";	