

## HtmlTemplateタグ一覧 (04/04/01)

仕様決定: 年 月 日

HtmlTemplateバージョン:

デザイン担当:

プログラム担当:

	使用	タ グ	意 味	\$out配列を渡す場合
基本 7 タグ		{val foo}	改行コードを に置き換えてfooの内容を表示する	\$out['foo']
		{rval foo}	fooの内容をそのまま表示する	\$out['foo']
		<!--[def hoge]--> ～<!--[/def]-->	hogeが定義済みである場合に、タグの間の～部分を表示する。	\$out['foo']
		<!--[ndef hoge]--> ～<!--[/ndef]-->	hogeが未定義である場合に、タグの間の～部分を表示する。	\$out['foo']
		<!--[vdef hoge]--> ～<!--[/vdef]-->	hogeが未定義である場合に、タグの間の～部分を表示する。 (<!--[each hoge]-->～<!--[/each]-->内で使用)	\$out['foo']
		<!--[each hoge]--> ～<!--[/each]-->	hogeに関して、タグの間の～部分をくり返す。 ～部分に値を埋め込む例: {val hoge/foo}	\$out['hoge'][]['foo']
ユ ー ザ ー 定 義 タ グ		<!--[comment]--> ～<!--[/comment]-->	タグの間の～部分をコメントとして扱い、出力しない。	配列は渡されません
		{def hoge:～}	hogeが空でない場合に、タグ内の～の部分を表示する。(単タグ)	\$out['foo']
		{ndef hoge:～}	hogeが空の場合に、タグ内の～部分を表示する。(単タグ)	\$out['foo']
		<!--[switch hoge:val]--> ～<!--[/switch]-->	hogeがvalに等しい場合に、タグの間の～部分を表示する。	\$out['foo']
		{switch hoge:val:～}	hogeがvalに等しい場合に、タグ内の～部分を表示する。(単タグ)	\$out['foo']
		{set_comma foo}	fooにコンマを追加して表示する(数値のみ)	\$out['foo']
旧 タ グ		<!--[ifdef hoge]--> ～<!--[/ifdef]-->	hogeが定義済みである場合に、タグの間の～部分を表示する。	\$out['foo']
		<!--[ifndef hoge]--> ～<!--[/ifndef]-->	hogeが未定義である場合に、タグの間の～部分を表示する。	\$out['foo']

Memo:

## HtmlTemplateタグ一覧 (04/04/01)

仕様決定:      年      月      日

HtmlTemplateバージョン:

デザイン担当:

プログラム担当:

	使用	タ グ	意 味	\$out配列を渡す場合
基本 7 タグ		{val foo}	改行コードを に置き換えてfooの内容を表示する	\$out['foo']
		{rval foo}	fooの内容をそのまま表示する	\$out['foo']
		<!--{def hoge}--> ~<!--/{def}-->	hogeが定義済みである場合に、タグの間の~部分を表示する。	\$out['foo']
		<!--{ndef hoge}--> ~<!--/{ndef}-->	hogeが未定義である場合に、タグの間の~部分を表示する。	\$out['foo']
		<!--{vdef hoge}--> ~<!--/{vdef}-->	hogeが未定義である場合に、タグの間の~部分を表示する。 (<!--{each hoge}-->~<!--/{each}-->内で使用)	\$out['foo']
		<!--{each hoge}--> ~<!--/{each}-->	hogeに関して、タグの間の~部分をくり返す。 ~部分に値を埋め込む例: {val hoge/foo}	\$out['hoge'][]['foo']
		<!--{comment}--> ~<!--/{comment}-->	タグの間の~部分をコメントとして扱い、出力しない。	配列は渡されません
ユ ー ザ ー 定 義 タ グ		{def hoge:~}	hogeが空でない場合に、タグ内の~の部分を表示する。(単タグ)	\$out['foo']
		{ndef hoge:~}	hogeが空の場合に、タグ内の~部分を表示する。(単タグ)	\$out['foo']
		<!--{switch hoge:val}--> ~<!--/{switch}-->	hogeがvalに等しい場合に、タグの間の~部分を表示する。	\$out['foo']
		{switch hoge:val:~}	hogeがvalに等しい場合に、タグ内の~部分を表示する。(単タグ)	\$out['foo']
		{set_comma foo}	fooにコンマを追加して表示する (数値のみ)	\$out['foo']
旧 タ グ		<!--{ifdef hoge}--> ~<!--/{ifdef}-->	hogeが定義済みである場合に、タグの間の~部分を表示する。	\$out['foo']
		<!--{ifndef hoge}--> ~<!--/{ifndef}-->	hogeが未定義である場合に、タグの間の~部分を表示する。	\$out['foo']

Memo:

HtmlTemplateタグ一覧 (04/04/01)

仕様決定:        年    月    日

HtmlTemplateバージョン:

デザイン担当:

プログラム担当:

	使用	タ グ	意 味	\$out配列を渡す場合
基本 7 タ グ		{val foo}	改行コードを</br>に置き換えてfooの内容を表示する	\$out['foo']
		{rval foo}	fooの内容をそのまま表示する	\$out['foo']
		<!--{def hoge}--> ～<!--{/def}-->	hogeが定義済みである場合に、タグの間の～部分を表示する。	\$out['foo']
		<!--{ndef hoge}--> ～<!--{/ndef}-->	hogeが未定義である場合に、タグの間の～部分を表示する。	\$out['foo']
		<!--{vdef hoge}--> ～<!--{/vdef}-->	hogeが未定義である場合に、タグの間の～部分を表示する。 (<!--{each hoge}-->～<!--{/each}-->内で使用)	\$out['foo']
		<!--{each hoge}--> ～<!--{/each}-->	hogeに関して、タグの間の～部分をくり返す。 ～部分に値を埋め込む例: {val hoge/foo}	\$out['hoge'][][ 'foo']
		<!--{comment}--> ～<!--{/comment}-->	タグの間の～部分をコメントとして扱い、出力しない。	配列は渡されません
ユ ー ザ ー 定 義 タ グ		{def hoge:～}	hogeが空でない場合に、タグ内の～の部分を表示する。(単タグ)	\$out['foo']
		{ndef hoge:～}	hogeが空の場合に、タグ内の～部分を表示する。(単タグ)	\$out['foo']
		<!--{switch hoge:val}--> ～<!--{/switch}-->	hogeがvalに等しい場合に、タグの間の～部分を表示する。	\$out['foo']
		{switch hoge:val:～}	hogeがvalに等しい場合に、タグ内の～部分を表示する。(単タグ)	\$out['foo']
		{set_comma foo}	fooにコンマを追加して表示する(数値のみ)	\$out['foo']
旧 タ グ		<!--{ifdef hoge}--> ～<!--{/ifdef}-->	hogeが定義済みである場合に、タグの間の～部分を表示する。	\$out['foo']
		<!--{ifndef hoge}--> ～<!--{/ifndef}-->	hogeが未定義である場合に、タグの間の～部分を表示する。	\$out['foo']

Memo:

HtmlTemplateタグ一覧 (04/04/01)

仕様決定:      年      月      日

HtmlTemplateバージョン:

デザイン担当:

プログラム担当:

	使用	タ グ	意 味	\$out配列を渡す場合
基本 7 タ グ		{val foo}	改行コードを</br>に置き換えてfooの内容を表示する	\$out['foo']
		{rval foo}	fooの内容をそのまま表示する	\$out['foo']
		<!--{def hoge}--> ~<!--/{def}-->	hogeが定義済みである場合に、タグの間の~部分を表示する。	\$out['foo']
		<!--{ndef hoge}--> ~<!--/{ndef}-->	hogeが未定義である場合に、タグの間の~部分を表示する。	\$out['foo']
		<!--{vdef hoge}--> ~<!--/{vdef}-->	hogeが未定義である場合に、タグの間の~部分を表示する。 (<!--{each hoge}-->~<!--/{each}-->内で使用)	\$out['foo']
		<!--{each hoge}--> ~<!--/{each}-->	hogeに関して、タグの間の~部分をくり返す。 ~部分に値を埋め込む例: {val hoge/foo}	\$out['hoge'][][foo]
		<!--{comment}--> ~<!--/{comment}-->	タグの間の~部分をコメントとして扱い、出力しない。	配列は渡されません
ユ ー ザ ー 定 義 タ グ		{def hoge:~}	hogeが空でない場合に、タグ内の~の部分を表示する。(単タグ)	\$out['foo']
		{ndef hoge:~}	hogeが空の場合に、タグ内の~部分を表示する。(単タグ)	\$out['foo']
		<!--{switch hoge:val}--> ~<!--/{switch}-->	hogeがvalに等しい場合に、タグの間の~部分を表示する。	\$out['foo']
		{switch hoge:val:~}	hogeがvalに等しい場合に、タグ内の~部分を表示する。(単タグ)	\$out['foo']
		{set_comma foo}	fooにコンマを追加して表示する(数値のみ)	\$out['foo']
旧 タ グ		<!--{ifdef hoge}--> ~<!--/{ifdef}-->	hogeが定義済みである場合に、タグの間の~部分を表示する。	\$out['foo']
		<!--{ifndef hoge}--> ~<!--/{ifndef}-->	hogeが未定義である場合に、タグの間の~部分を表示する。	\$out['foo']

Memo: