

SYMETRIC



GENECODE

ユーザマニュアル

テンプレート構文リファレンス

第 2 版

最終更新日 2013/1/10

目次

第 1 章	概要	3
第 2 章	GeneCode タグ一覧	4
2-1	<gc-script>タグ	4
2-2	<gc-include>タグ	4
2-3	<gc-log>タグ	5
2-4	<gc-each>タグ	6
2-5	<gc-parts>タグ	6
2-6	<gc-paramList>タグ	7
2-7	<gc-preprocess>タグ	8
2-8	<gc-function>タグ	8
2-9	<gc-param>タグ	9
2-10	<gc-view>タグ	9
第 3 章	出しわけ設定	11
3-1	cond 属性	11
第 4 章	テンプレートコメント	12
第 5 章	meta タグ	13
5-1	CSS 変換機能の指定	13
第 6 章	JavaScript 仕様	14
6-1	JavaScript バージョン	14
6-2	ブラウザオブジェクト	14
6-3	DOM	14

第1章 概要

GeneCode 変換テンプレートでは GeneCode タグを用いて様々な機能を実現しています。GeneCode タグは XML 構文に準拠しており、空タグで記述する場合は /> を記述します。

各タグは GeneCode に正しく解釈された場合はサーバ上で処理されるので、ブラウザ上には表示されません。また、処理の結果エラーが発生しても GeneCode タグは消えてなくなります。

また、テンプレートファイルのキャラクタエンコーディングは UTF-8 固定です。

テンプレート中に記述する GeneCode 用 JavaScript で問題が発生した場合は、サーバの GCHtmlLogFile ディレクティブで設定したファイルにエラーが出力され、PC サイトが変換されない状態で出力されます。

第2章 GeneCode タグ一覧

2-1 <gc-script>タグ

属性	なし
終了タグ	必須
空タグ	不可
説明	<p>gc-script はテンプレート中に GeneCode ランタイム上で実行させる JavaScript を記述するために用います。gc-script の終了タグまでを JavaScript として認識するので、他の GeneCode タグとの入れ子になるような記述はできません。</p> <p>gc-script 内で記述できる JavaScript の仕様については、第 6 章を参照してください。</p>
例	<p>【テンプレート】</p> <pre><div> <gc-script> var msg = "JavaScript 上で生成した文字列です"; gcutil.out(msg); </gc-script> </div></pre> <p>【実行結果】</p> <pre><div> JavaScript 上で生成した文字列です </div></pre>

2-2 <gc-include>タグ

属性	<ul style="list-style-type: none"> ● path: テンプレートファイルパス (必須)
----	---

	<ul style="list-style-type: none"> ● cond: 実行される条件判定関数名 (任意) ● name: 名前。cond を使う場合に使用する (任意)
終了タグ	不可
空タグ	可
説明	<p>テンプレートディレクトリ以下に配置されている別のテンプレートを読み込み、展開します。</p> <p>読み込むテンプレートは path 属性に記述し、カレントからのパスで記述した場合は、読み込み元のテンプレートディレクトリから見たパスとなり、「/」から始まるパスで記述した場合は、テンプレートディレクトリからのパスとなります。</p> <p>読み込まれた外部テンプレートは、通常のテンプレートと同じく全ての GeneCode タグが処理されます。cond 属性については、第 3 章を参照してください。</p>
例	<code><gc-include path="footer.html" /></code>

2-3 <gc-log>タグ

属性	<ul style="list-style-type: none"> ● level: 出力するログレベルを 1(error)、2(debug)、3(trace)のいずれかで指定します。省略時は 2 とみなされます。 ● category: 出力するログカテゴリを 1(user)もしくは 2(system)のいずれかで指定します。省略時は 1 とみなされます。 ● class: HTML で出力する際に付与するクラス属性を指定します (任意)。
終了タグ	不可
空タグ	可
説明	<p><code>gcutil.error()</code>、<code>gcutil.debug()</code>、<code>gcutil.trace()</code>関数で出力するログの内容をブラウザのコンソールに出力したい場合に記述します。</p> <p>ブラウザにコンソールがない場合は、HTML タグでログ内容を出力します。level 属性と category 属性で出力内容をフィルタリングすることができます。</p>

	各ログ出力関数を用いてログを出力する場合は、user カテゴリ(1)で出力されます。
例	<pre> <gc-script> gcutil.debug(\$(".body").get(0).outerHTML); </gc-script> <gc-log level="2" category="1" /> </pre>

2-4 <gc-each>タグ

属性	<ul style="list-style-type: none"> ● selector: 繰り返す対象となるセレクタ(必須) ● dh: 無視する先頭要素数(任意) ● dt: 無視する末尾要素数(任意)
終了タグ	必須
空タグ	不可
説明	終了タグまでに記述されているテンプレート・スクリプト・パーツを selector 属性で指定したセレクタで抽出された要素の数だけ繰り返します。
例	<pre> <gc-each selector=""> </gc-each> </pre>

2-5 <gc-parts>タグ

属性	<ul style="list-style-type: none"> ● name: 配置したパーツ名(必須)。 ● cond: 実行される条件判定関数名(任意) ● gcid: GeneCode ビルダークラから付与される ID(任意) ● view: GeneCode ビルダークラで選択したビューID(任意)
----	---

	<ul style="list-style-type: none"> ● cond: 実行される条件判定関数名 (任意)
終了タグ	必須
空タグ	不可
説明	GC パーツを配置します。gc-parts の子要素には gc-paramList、gc-preprocess、gc-view を含める必要があります。基本的にはこのタグは直接記述せず、GC ビルダ一を使いパーツを配置します。cond 属性については、第 3 章を参照してください。
例	<pre> <gc-parts gcid="gcpart-BASICIMAGE01-1" name="BASICIMAGE01" view="001"> <gc-paramList> <gc-param name="selector">body&gt;div&gt;div&gt;img</gc-param> <gc-param name="imgWidth"></gc-param> <gc-param name="imgHeight"></gc-param> <gc-param name="unit">px</gc-param> <gc-param name="clipX1">0</gc-param> <gc-param name="clipY1">0</gc-param> <gc-param name="clipX2">177</gc-param> <gc-param name="clipY2">40</gc-param> </gc-paramList> <gc-preprocess></gc-preprocess> <gc-view> <div class="BASICIMAGE01" gcp="auto"> PC サイト上の単一の IMG の調整・変換を行う </div> </gc-view> </gc-parts> </pre>

2-6 <gc-paramList>タグ

属性	なし
終了タグ	必須

空タグ	不可
説明	gc-param タグのコンテナです。基本的にはこのタグは直接記述せず、GCビルダーを使いを配置します。
例	<pre><gc-paramList> <gc-param name="selector">.main #list</gc-param> </gc-paramList></pre>

2-7 <gc-preprocess>タグ

属性	なし
終了タグ	必須
空タグ	不可
説明	gc-function タグのコンテナです。基本的にはこのタグは直接記述せず、GCビルダーを使いを配置します。
例	<pre><gc-preprocess> <gc-function name="func"></gc-function> </gc-preprocess></pre>

2-8 <gc-function>タグ

属性	<ul style="list-style-type: none"> name: 呼び出す preprocess 関数名
終了タグ	オプション
空タグ	可
説明	パーツで処理する要素に対して事前処理を行う関数を指定するタグです。基本的にはこのタグは直接記述せず、GCビルダーを使いを配置します。

例	<pre><gc-preprocess> <gc-function name="func"></gc-function> </gc-preprocess></pre>
---	---

2-9 <gc-param>タグ

属性	<ul style="list-style-type: none"> name: パラメータ名 (gc-paramList の子要素で指定する場合は必須)。
終了タグ	必須
空タグ	不可
説明	GC パーツへのパラメータを指定します。gc-paramList の子要素として記述する場合は、name 属性にパラメータ名を指定し、テキストノードで値を指定します。gc-function の子要素で指定する場合は、name 属性は記述せず、テキストノードで値だけを指定します。基本的にはこのタグは直接記述せず、GC ビルダ―を使いを配置します。
例	<pre><gc-param name="selector">.main #list</gc-param></pre>

2-10 <gc-view>タグ

属性	なし
終了タグ	必須
空タグ	不可
説明	GC パーツが使用するビューです。基本的にはこのタグは直接記述せず、GC ビルダ―を使いを配置します。
例	<pre><gc-view> <div class="foo" gcp="step:step1"> </div></pre>

	</gc-view>
--	------------

第3章 出しわけ設定

GeneCode タグのうち、gc-parts タグと gc-include タグは、実行させる条件を cond 属性を使って指定することができます。

3-1 cond 属性

cond 属性には JavaScript 関数名を指定します。指定する関数は、この属性を記述するテンプレートから参照できる場所で定義されている必要があります。例えばテンプレート内の<gc-script>タグや、外部の JS ファイルをロードして定義します。cond 属性で指定する JavaScript 関数は、以下のプロトタイプで定義します。

```
bool function(name);
```

引数 name には GeneCode タグで指定される name 属性の値が文字列が渡されます。戻り値は該当する GeneCode タグを実行するかどうかの真偽値を返却します。GeneCode タグの name 属性が未設定の場合は、undefined が渡されます。

例

header.html は読み込まれないが、footer.html は読み込まれる例

```
<gc-script>
function isActive(name) {
    if(name == "footer") {
        return true;
    }
    return false;
}
</gc-script>
<gc-include path="header.html" name="header" cond="isActive" />
<gc-include path="footer.html" name="footer" cond="isActive" />
```

第4章 テンプレートコメント

GeneCode テンプレート上でコメントを記述する場合は、以下の書式を用います。

```
<%-- テンプレートコメント --%>
```

<%-- から --%>までの間に存在する HTML タグ、GeneCode タグ、JavaScript などはいずれも評価・実行されず、レスポンス HTML にも出力されません。

※HTML コメント(<!-- -->)を用いて GeneCode タグをコメントにしようとしても、GeneCode 上では評価・実行されてしまいますので、注意してください。

第5章 meta タグ

5-1 CSS 変換機能の指定

CSS 変換シートによる全体変換と@genecode ブロックによる部分変換のそれぞれに対し、変換の有効・無効を meta タグで設定することができます。

書式	説明
<code><meta name="gc-css-convert" content="no"></code>	CSS 変換機能をすべて無効にします。CSS 変換シートおよび@genecode ルールの展開が行われません。
<code><meta name="gc-css-convert" content="conversion-sheet"></code>	CSS 変換シートによる変換のみを行います。
<code><meta name="gc-css-convert" content="atrule"></code>	@genecode ルールの展開のみを行います。
meta タグ指定なし	CSS 変換機能をすべて有効にする。CSS 変換シートおよび@genecode ルールの展開を行います。

第6章 JavaScript 仕様

6-1 JavaScript バージョン

ECMAScript 5 (ECMA-262) に準拠しています。

6-2 ブラウザオブジェクト

document オブジェクトのみ使用できます。ただし、document オブジェクトは実行されているテンプレートの HTML ではなく、変換元コンテンツ(PC サイト)の HTML を指していることに注意してください。

GeneCode ランタイムでは、ブラウザ上の JavaScript とは異なり、window や navigator 等のブラウザオブジェクト(document 以外)は使用できません。

6-3 DOM

以下の仕様をサポートしています。

- DOM Level 3 Core
- DOM Level 2 HTML

その他、Element インタフェースの innerHTML, outerHTML プロパティに対応しています。

※DOM Level 3 Event, DOM Level 3 Load and Save, DOM Level 3 Validation, DOM Level 3 Xpath には対応していません。

※DOM Level 2 Style (Element インタフェースの style プロパティ)には対応していません。