SYMETRIC

ジーンコード ver 3

ユーザマニュアル

テンプレート構文リファレンス

第 1 版

最終更新日 2018/12/20

目次

改訂履歴

版数	発行日	改訂内容
第 1 版	2018 年 12 月 20 日	・初版発行

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<gc-include />のように記述します。

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	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。 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。
例	【テンプレート】 　　<gc-script> 　　　　var msg = "JavaScript 上で生成した文字列です"; 　　　　gcutil.out(msg); 　　</gc-script> 【実行結果】 　　JavaScript 上で生成した文字列です

2-2　<gc-include>タグ

属性	● path：テンプレートファイルパス（必須）

	● 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	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。 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。 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。
例	<gc-include path="footer.html" />

2-3　<gc-log>タグ

属性	● 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	gcutil.error()、gcutil.debug()、gcutil.trace()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。 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。

	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	<gc-script> gcutil.debug($("body").get(0).outerHTML); </gc-script> <gc-log level="2" category="1" />

2-4 <gc-each>タグ

属性	● selector：繰り返す対象となるセレクタ（selector もしくは mark 必須） ● mark:繰り返す対象となるマーク（selector もしくは mark 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、もしくは mark 属性で指定されたマークに対し、抽出された要素の数だけ繰り返します。
例	<gc-each selector="div.itemList < div"> <gc-parts>...</gc-parts> </gc-each>

2-5 <gc-parts>タグ

属性	● 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 ビルダーから付与される ID（任意） ● view：GeneCode ビルダーで選択したビューID（任意） ● 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 ビルダー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	<gc-parts gcid="gcpart-BASICIMAGE01-1" name="BASICIMAGE01" view="001"> <gc-paramList> <gc-param name="selector">body>div>div>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> PC サイト上の単一の IMG の調整・変換を行う <img gcp="step:step1"> </gc-view> </gc-parts>

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 ビルダーを使いパーツを配置します。
例	<gc-paramList> 　<gc-param name="selector">.main #list</gc-param> </gc-paramList>

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 ビルダーを使いパーツを配置します。
例	<gc-preprocess> 　<gc-function name="func"></gc-function> </gc-preprocess>

2-8　<gc-function>タグ

属性	● 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 ビルダーを使いパーツを配置します。
例	<gc-preprocess> 　<gc-function name="func"></gc-function> </gc-preprocess>

2-9　<gc-param>タグ

属性	● 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 ビルダーを使いパーツを配置します。 なお、gc-param タグで name="selector"もしくは name="mark"の場合の書式については、第 4 章を参照してください。
例	<gc-param name="selector">.main #list</gc-param>

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 ビルダーを使いパーツを配置します。
例	<gc-view> 　 　 </gc-view>

2-11　<gc-convertScript>タグ

属性	なし
終了タグ	必須
空タグ	不可
説明	PC のページ内に埋め込まれている<script>タグから JavaScript 文字列リテラルを置換するための置換ルール用タグです。このタグは直接記述せず、IDE から挿入・編集を行います。

例	<gc-convertScript>
	　{
	　　"replaceStringLiteral": [
	　　　{
	　　　"source": "/resources/image/logo.gif",
	　　　"dest": "/template/img/logo.png"
	　　　}
	　　]
	　}
	
	</gc-convertScript>

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　　セレクタとマークの構文

<gc-param>タグでパーツのセレクタやマークを指定する場合、以下の構文で記述します。

4-1　　セレクタ書式

セレクタには、jQuery 1.7 相当の jQuery セレクタ書式が使用できます。以下に例を示します。

セレクタ式	説明
div#foo	id 属性が foo に等しい div 要素が選択されます。
form:has(textarea)	子孫要素に textarea 要素を含んでいる form 要素が選択されます。

詳細は jQuery 公式 http://api.jquery.com/ を参照してください。

4-2　　マーク書式

マークの書式は使用するマーク名をそのまま指定します。複数のマークを結合した結果を利用する場合は、カンマ「,」区切りで指定します。以下に例を示します。

マーク式	説明
itemList	マーク itemList が付与された要素を結合した要素が選択されます。
headerItemList,footerItemList	マーク headerItemList が付与された要素の後に、マーク footerItemList が付与された要素を結合した要素が選択されます。

4-3　　インデックス変数

<gc-each>タグでセレクタを指定した場合、<gc-each>内のパーツのセレクタでインデックスを表す変数が利用できます。インデックスは 0 始まりの整数で、%GCIDX%で参照できます。

例: gc-each 内でパーツを使用する例

```
<gc-each selector="div.itemList">
  <gc-parts name="xxx">
    <gc-param name="selector">div.itemList:eq(%GCIDX%) p</gc-param>
  </gc-parts>
</gc-each>
```

4-4　スコープ

<gc-each>タグ内でパーツを使用する場合、セレクタ式もしくはマーク式の先頭にスコープ指定子を指定することができます。スコープ指定子を指定する場合、直後には 1 つ以上のスペースが必要です。

スコープ指定子	説明
未指定	<gc-each>タグ内で使用するパーツで、スコープ指定子・インデックス変数のいずれも使用しないセレクタ式・マーク式場合は、@THIS が指定されているものとみなされます。
@THIS	<gc-each>タグで繰り返されるエントリのスコープを示します。例えば、@THIS は繰り返されるエントリを指し、@THIS > div は繰り返されるエントリの直下の div 要素を指します。
@GLOBAL	グローバルスコープを示します。<gc-each>タグで繰り返されるエントリのスコープを使用しない場合に指定します。

例:

```
<gc-each selector="div.itemList">
  <gc-parts name="xxx">
    <gc-param name="selector">@THIS p</gc-param>
  </gc-parts>
</gc-each>
```

※スコープ指定は<gc-each>タグにのみ使用でき、JavaScript API では使用できません。

第5章　　ノードパスの構文

パーツでラベルパスを指定する場合、コンテンツと連動する場所の指定方法としてノードパスと呼ばれる書式を使用します。ノードパスの書式は以下の通りです。

/タグ名 1[インデックス 1]/タグ名 2[インデックス 2] ・・・

ノードパスは基点となる要素を基準とし、対象となる要素までのタグ名とインデックスをすべて記述します。インデックスは同一階層内でそのタグ名が何番目に出現したかを 0 始まりで指定します。ノードパスには基点となる要素自身も含める必要があります。また、タグ名・インデックスは省略できません。

例えば、以下の例で基準となる要素が article の場合、最後の p 要素へのノードパスは
/article[0]/div[0]/p[1] となります。

<article>
　┗
　　┣<h4>
　　┣<p>
　　┗<p>

第6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7章　　meta タグ

7-1　　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
<meta name="gc-css-convert" content="no">	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。
<meta name="gc-css-convert" content="conversion-sheet">	CSS 変換シートによる変換のみを行います。
<meta name="gc-css-convert" content="atrule">	@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8章　　JavaScript 仕様

8-1　　JavaScript バージョン

ECMAScript 5（ECMA-262）に準拠しています。

8-2　　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。

8-3　　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。

8-4　　jQuery

jQuery 1.7.2 に準拠しており、$関数でインスタンスを作成できます。なお、jQuery 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）のドキュメントを指していることに注意してください。