SYMMETRIC CO., LTD.

GeneCode を使用した制作の流れ

株式会社シンメトリック

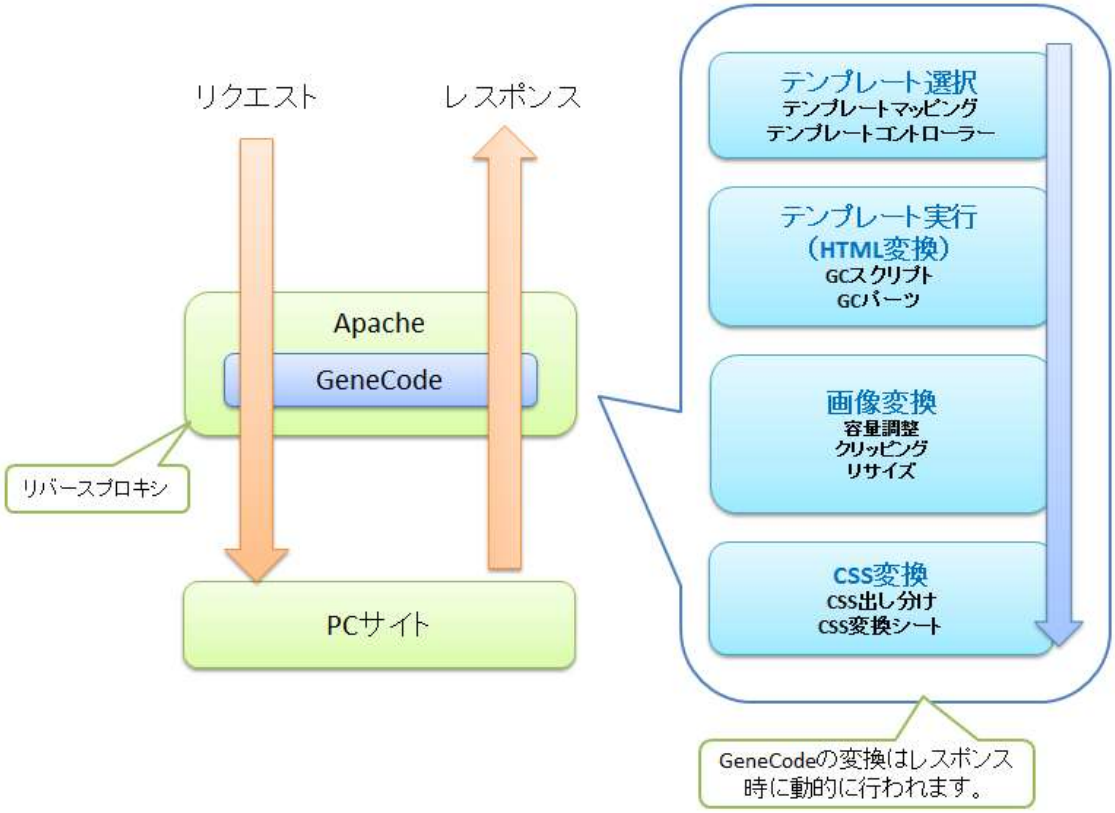
2013/07/02

1. GeneCode とは

GeneCode とは既存の PC サイトとスマートフォンの間に立ち(リバースプロキシ)、PC サイトのレスポンスから必要な要素を抽出・変換することでスマートフォンやタブレットなどマルチデバイスへの表示に最適化するツールです。

PC サイトのリソースを再利用し、見た目だけをスマートフォンなどに最適化し、バックグランドのアプリケーションをそのまま流用することができるためコストを大きく削減することができます。

図 1 GeneCode 処理概要



2. 変換の仕組み

GeneCode の変換は「テンプレート」と呼ばれるファイルをベースに行われます。テンプレートは通常の HTML＋いくつかの独自タグで構成されます。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>sample</title>
<body>
<!-- gc-script タグ  -->
<gc-script>
  gcutil.out($("#someContents").html());
</gc-script>
<!-- gc-parts タグ  -->
<gc-parts gcid="xxx" name="xxx" view="001">
  <gc-paramlist>
    <gc-param name="selector">#someContents</gc-param>
  </gc-paramlist>
  <!--  詳細省略  -->
</gc-parts>
</body>
</html>
```

gc-script タグ内には GeneCode サーバー上で実行する JavaScript を記述できます。

CSS セレクタ

gc-parts タグは GeneCode ビルダーから GC パーツを挿入した際に追加されます。
GC パーツについては後述します。

基本的な変換処理の流れは以下のようになります。

① CSS セレクタで、PC コンテンツから取得したい箇所を特定する。
② GC スクリプトまたは GC パーツで PC コンテンツの要素を加工・出力する。

なお「gc-」で始まる独自タグ以外（通常の HTML タグやクライアントサイド用の script タグなど）はそのまま出力されます。完全な独自タグの一覧は以下をご参照ください。

参考：『テンプレート構文リファレンス』
http://developer.genecode.jp/manuals/gcruntime/1.2.0/gc_template_syntax.pdf

テンプレートは普段お使いのテキストエディタでも編集可能ですが、GeneCode ビルダーを使用することでより効率よく作成することができます。

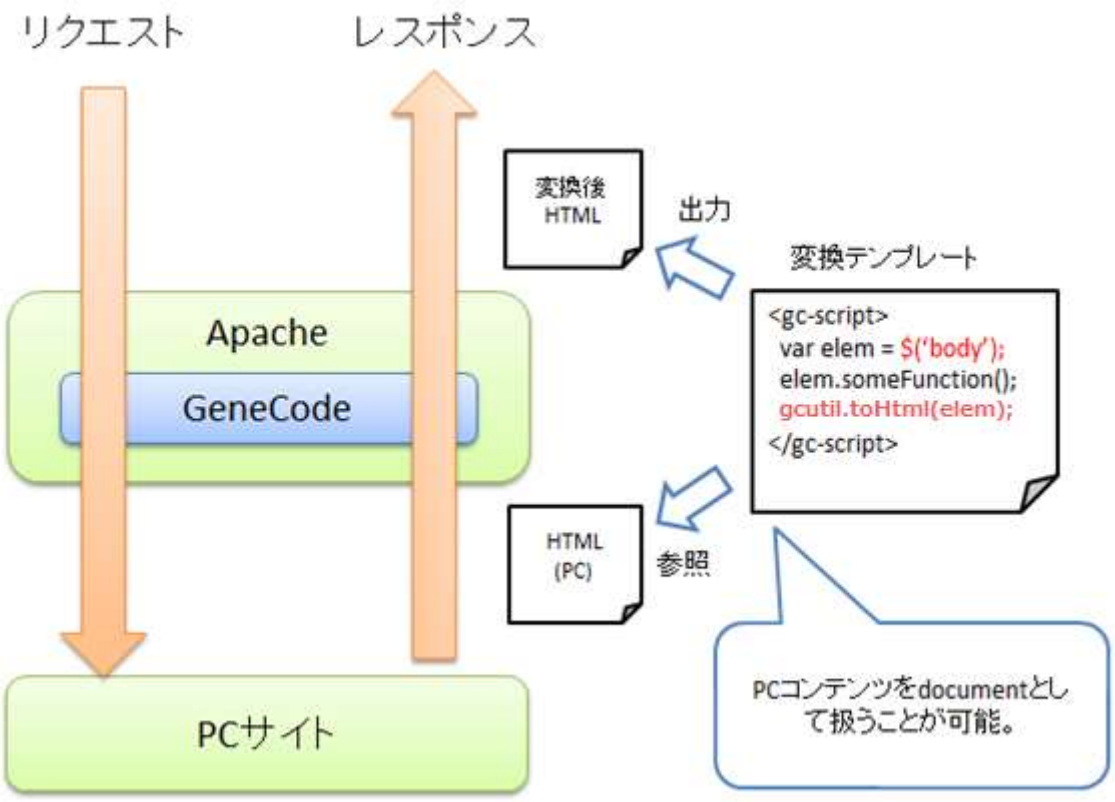
2.1. GC スクリプト

2.1.1. GC スクリプトとは

GC スクリプトは gc-script タグ内に記述する、**サーバーサイド JavaScript** です。通常の JavaScript に GeneCode 独自の API を一部追加しているほか、jQuery ライブラリを標準で読み込んでいます。

GC スクリプト内では、PC コンテンツの DOM を自由に編集して、テンプレート変換後の HTML として出力することができます。

図 2 テンプレート処理概要



2.1.2. 主な GC スクリプト API

GC スクリプトは基本的に通常の JavaScript と同じですが、入出力やロギングなどについて、独自の API を使用することができます。以下に頻繁に利用するであろうものについて一部紹介します。

表 1 主な GC スクリプト API

API	説明
gcutil.out(str)	引数で指定された文字列をレスポンスとして出力します。
gcutil.toHtml(elem)	引数で指定された jQuery オブジェクトをブラウザに出力します。
gcutil.toDebugHtml(elem)	引数で指定された jQuery オブジェクトを debug レベルでログに出力します。
gclog.error(str) / gclog.debug(str) / gclog.trace(str)	エラー / デバッグ / トレース の各レベルでログ出力を行います。またテンプレート上で gc-log タグを記述することでブラウザ上でログを確認できます。 ※gc-log タグについては『テンプレート構文リファレンス(pdf)』を参照してください。
gcutil.getParameter(name)	リクエスト URL のクエリ文字列から指定されたパラメータ名の値を取得します。
gcutil.getHeader(name)	リクエストヘッダ値を表す文字列を返します。リクエストヘッダが存在しない場合は undefined を返します。

完全な API の一覧は以下をご参照ください。

参考：『JavaScriptAPI リファレンス』
http://developer.genecode.jp/manuals/gcruntime/1.2.0/gc_javascript_api.pdf

2.2. GC パーツ

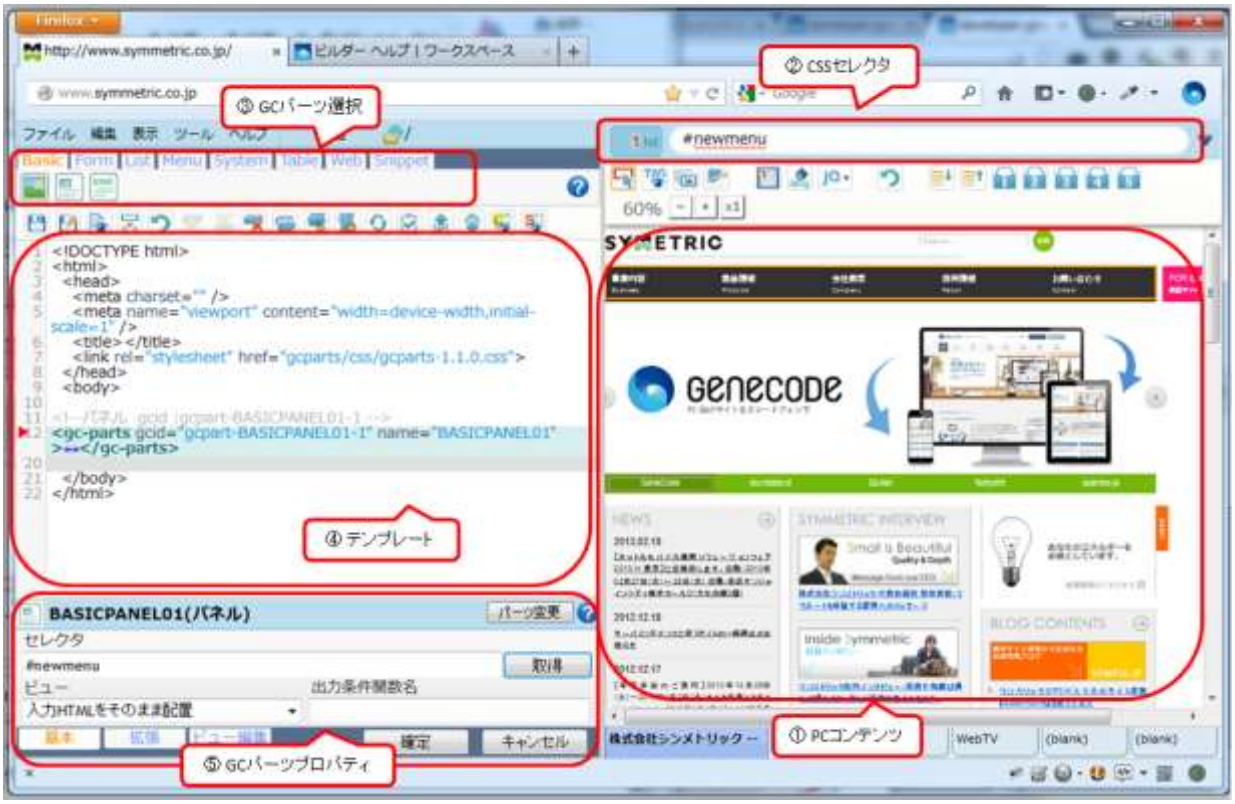
2.2.1. GC パーツとは

GC パーツとは、指定した PC コンテンツを特定の HTML 構造に変換して出力するツールです。GC パーツは GeneCode ビルダーから選択してマウス操作で使用できるため、GC スクリプトよりも手軽に変換を行うことができます。

GeneCode ビルダー自体の使い方等は以下を参照してください。

参考：『GeneCode ビルダーヘルプ』
http://developer.genecode.jp/manuals/gcbuilder/1.2.0/

図 3 GeneCode ビルダー



2.2.2. GC パーツ使用手順

1. 「PC コンテンツ」エリアから取得したい箇所をクリックします。
2. 「CSS セレクタ」エリアに選択した箇所の CSS セレクタが表示されます。
3. 「GC パーツ」エリアに表示されている GC パーツから使用したいパーツを選択します。
4. 「テンプレート」エリアに選択したパーツのタグが書き込まれます。
5. 「GC パーツプロパティ」の「取得」ボタンをクリックします。
   「GC パーツプロパティ」の「セレクタ」欄に「CSS セレクタ」エリアに表示されているセレクタの値がコピーされます。
6. 「GC パーツプロパティ」の「確定」ボタンをクリックします。
7. パーツの表示を確認するには、テンプレートファイルの保存、アップロード、プレビューを行います。

2.2.3. 主な GC パーツ

ここでは頻繁に使われるであろう基本的なパーツの概要についてのみ説明します。各パーツのオプションや詳細な仕様、および完全なパーツの一覧は下記を参照してください。

> 参考：『GeneCode パーツヘルプ』
> http://developer.genecode.jp/manuals/gcparts/

表 2 主な GC パーツ

パーツ	説明
パネル	選択した PC コンテンツの HTML 構造を変えずにそのまま出力します。 ただし不要な属性を削除することで元コンテンツ側のデザインへの依存がなくなり、新しいデザインによる装飾・配置の調整が可能になります。
画像	選択した HTML 断片中に含まれる A 要素・IMG 要素を取り出し出力します。 横幅・高さの属性値設定と、画像クリッピング変換の指示パラメータの指定を行うことができます。
テキストリスト	テキストやリンクで構成されたリスト状のコンテンツを、タッチ・視認しやすいリストとして出力します。
画像付きリスト	画像とテキストやリンクで構成されたリスト状のコンテンツを、タッチ・視認しやすいサムネイル画像付きのリストとして出力します。
テキストメニュー	サイトメニューやカテゴリ一覧の HTML 領域を選択し、テキスト形式のメニューで表示します。 入れ子構造のメニューにも対応しています。
背景付きメニュー	サイトメニューやカテゴリ一覧の HTML 領域を選択し、アイコン画像（背景）とテキストのメニューで表示します。 デフォルトでは製品標準の画像が背景として使われますが、画像リソースを差し替えたり、CSS で背景画像を変更、制御したり、元コンテンツで使用している背景画像リソースを使うこともできます。

2.3. PC サイトのマーキング

2.3.1. マーキングとは

GeneCode では PC サイトにあらかじめ目印となる属性を記述しておくことで、より手軽に変換を行うことができます。これをマーキングと呼びます。HTML 構造に依存しない目印によって、変更にも強くなります。主に以下のようなシーンで役に立ちます。

- PC サイトとモバイルサイトの同時開発
  （PC コンテンツの構造がまだ定まってない）
- PC コンテンツの構造が頻繁に変わってしまう
- PC コンテンツから取得したい範囲の一部分をピンポイントで変換したい

2.3.2. マーキングの種類

マーキングには以下の 4 つのタイプがあります。それぞれ独自属性（data-gc-XXX）の形式と class 属性の形式の 2 通りで記述できます。

2.3.2.1. マーク

GC パーツや GC スクリプトから参照したい要素に追加します。

```
■属性表記
<ul data-gc-mark="news">
  <li>・・・・</li>
</ul>

■class 属性表記
<ul class="GCMARK_news">
  <li>・・・・</li>
```

2.3.2.2. ラベル

マークを付けて参照した要素のうち、さらに別の処理をしたい要素に指定します。GC パーツでは各パーツ毎にラベルを付けた要素に変更を行います。
（例）サムネイルにする画像を指定する、アコーディオンの見出しとコンテンツを指定する、など

```
■属性表記
<ul data-gc-mark="news">
  <li data-gc-label="item">・・・・</li>
</ul>

■class 属性表記
<ul class="GCMARK_news">
  <li class="GCLABEL_item">・・・・</li>
</ul>
```

2.3.2.3. ラベルインデックス

ラベルを付けた要素を順番に処理したい場合に試用します。ラベルインデックスを使用しない場合は元コンテンツ上の出現順に処理します。

```
■属性表記
<ul data-gc-mark="news">
  <li data-gc-label="item" data-gc-labelindex="0">・・・・</li>
</ul>

■class 属性表記
<ul class="GCMARK_news">
  <li class="GCLABEL_item GCLABELINDEX_0">・・・・</li>
</ul>
```

2.3.2.4. コマンド

要素の変換や削除を指示するマーキングです。現在 3 種類のコマンドがサポートされています。

removeall

この値を指定した要素とその子孫要素が削除されます。

remove

この値を指定した要素だけを削除します。

rename

この値を指定した要素が、指定された要素名に変更されます。

■属性表記

```
<div data-gc-cmd="rename:ul">
   <span data-gc-cmd="rename:li">・・・</span>
   <div data-gc-cmd="removeall">
      ・・・
   </div>
</div>
```

■class 属性表記

```
<div class="GCCMD_rename_ul">
   <span class="GCCMD_rename_li">・・・</span>
   <div class="GCCMD_removeall">
      ・・・
   </div>
</div>
```

3. GeneCode を使用した制作の流れ

以降では実際に GeneCode を使用して制作する際の流れと注意点を示します。
大まかに以下の流れで制作を行います。

3.1 PC サイトから持ってきたい箇所を決める
3.2 スマートフォンサイトの画面イメージ作成
3.3 テンプレート作成 & プレビュー
3.4 スタイルの適用
3.5 動作確認

3.1. PC サイトから持ってきたい箇所を決める

基本的に PC サイト上で表示されているものは取得可能ですが、以下のケースについては注意が必要です。

- JavaScript
  ブラウザでの描画後に JavaScript で動的に取得・表示している要素は、GeneCode で取得することはできません。別の表現を考えるか、PC サイトの JavaScript を必要に応じて移植する必要があります。
- FORM 系
  画面上に表示されていない（hidden）のものが PC サイトの動作に関わっているケースがあります。その場合は忘れずに取得するようにしてください。
- IFRAME
  IFRAME の中身のコンテンツと外側のコンテンツは実際は別の HTML ですので、それぞれ別の変換テンプレートを作成する必要があります。GeneCode ビルダーを使用する際も中身のコンテンツの URL に直接アクセスして使用する必要があります。
- CSS による見た目
  GeneCode のテンプレートからは CSS のプロパティは取得できないため、例えば background-image で指定した背景画像などで表現されたボタンなどは、取得しただけでは見えない（HTML タグは取得できていても）場合があります。この場合はあとで別途 CSS を当てなおす必要があります。
- 動画や Flash
  タグ自体は取得できますが、動作するかは端末依存になります。

3.2. スマートフォンサイトの画面イメージ作成

上記で決定した抽出箇所を元に、スマートフォンで表示したいイメージをまとめます。その際、CSS は PC のものは流用できないので、新しくスマートフォン用の CSS を作成する想定でイメージを決めてください。

3.3. テンプレート作成 ＆ プレビュー

使用するパーツやその他の手法が決まったら、実際にテンプレートを作成していきます。GeneCode ビルダーのプレビュー機能を使用してください。

この時点ではデザインは気にせず PC から持ってきたい要素が過不足なく取得できることを目標にしてください。

またこの時点で PC サイトにマーキングが必要な（あるいは追加したほうが便利な）場合は PC コンテンツの変更を検討してください。

3.4. スタイルの適用

テンプレートで変換した結果はそのままではGCパーツデフォルトのCSSが適用されています。独自の CSS を作成して、画面イメージのデザインを実装してください。

このとき GC パーツを多用していれば、もとの PC サイトでの構造がどうであれ、HTML の構造が均一になるためスタイルの使いまわしがしやすくなります。

CSS のコーディング時には以下の点にご注意ください。

3.4.1. パーツデフォルトの CSS

GeneCode ビルダーで gc-parts を使用してテンプレートを作成したい際に、各パーツ独自のデフォルトのスタイル（gcparts-x.x.x.css）が当たっています。この CSS ファイルはあくまでパーツの動作確認のためのものです。そのまま利用して頂いても構いませんが、適用したいデザインによっては邪魔になることもありますので、gcparts-x.x.x.css を読み込んでいる link タグを除去しすることをお勧めします。

3.4.2. ベンダープレフィックスの記述方法

CSS のプロパティの中には、様々なブラウザに対応するために「-webkit-」「-moz-」「-o-」などのブラウザベンダーごとのプレフィックスを付ける必要がある場合があります。GeneCode では「-webkit-」のみを記述すればその他のプレフィックスに自動で展開します。プレフィックスが必要な場合は「-webkit-」のみを記述してください。

3.4.3. グラデーションの記述方法

CSS3 のグラデーション機能を使用する場合は以下のように記述してください。

■-webkit-gradient の書式は以下のみをサポート
※下記の書式以外は変換されずにそのまま出力されます。

①-webkit-gradient(linear, left top, left bottom, from(色), to(色));
②-webkit-gradient(linear, left top, left bottom, color-stop(xxxx%, 色),
color-stop(xxxx%, 色));
③-webkit-gradient(linear, left top, left bottom, color-stop(xxxx%, 色),
color-stop(xxxx%, 色), color-stop(xxxx%, 色));

※ 色の指定は #RGB #RRGGBB rgb(r,g,b) rgba(r,g,b,a) および色の名前が使える。
※ 中間色の位置は、0-1 の数値ではなく、%で指定する。
※ 「,」の後は 1 つの半角スペースで、それ以外に改行やスペースを入れない
※ -box-shadow で色を指定する場合は、5 個目のパラメータとして指定する必要がある。
（例）-webkit-box-shadow: 0px 1px 2px 0px rgba(0,0,0,0.3);

3.5. 動作確認

シンメトリックで通常以下のパターンで実機テストを行っています。

- Android2.3 系
- Android4.0 系
- iOS5 系
- iOS6 系

見た目の確認のほか、FORM や PC サイトから持ってきたスクリプトなどは特に動作確認を行ってください。

4. PC コンテンツ取得時の注意点

GeneCode で PC サイトから要素を抽出するには GeneCode ビルダー上で GC パーツを使用する方法と、gc-script タグを用いて GC スクリプトを記述する方法の 2 通りの方法が用意されています。

いずれの方法でも CSS セレクタを利用して PC コンテンツの箇所を指定することになりますので、CSS セレクタの知識は必須です。

> 参考：CSS セレクター一覧
> http://memopad.bitter.jp/w3c/jQuery/jquery_ref_selectors.html

4.1. GC パーツを使用する場合

GC パーツを使用する方法は、GeneCode ビルダーの機能をフル活用できるため、もっとも生産性が高い方法です。

ただし PC サイトはサイトによって様々な構造を持っているため、汎用的なパーツでうまく処理できないケースもあります。パーツで PC コンテンツが上手く取得できない場合には以下の手段か、GC スクリプトを使用することができます。

4.1.1. セレクタを変えてみる

例えば「リストやメニューの一部分だけを利用したい」などの場合に、CSS セレクタの書き方を工夫するだけでうまくいくケースが多々あります。

（例）リストの一部だけを使用したい場合

このケースではセレクタを「#newmenu>ol>li」から「#newmenu>ol>li:gt(0):lt(4)」などに変更することで、リスト中の使用したい部分だけを取得することができます。

他にもとびとびになっている要素を出力したい場合などは Ctrl を押しながら選択することで複数選択を行うことなども可能です。

4.1.2. 事前に DOM 構造を変更する

例えば FORM タグなど、まとめて取得しないと意味をなさない部分に、間に余計な要素が挟まっているケースや、スマートフォンサイトではメニューの順番を変えたいなどのケースがあります。

このような「並べ替え」や「不要な要素の除去」は GC パーツが実行される前に DOM 構造を変更することで実現できます。

マーキングを利用する

マーキングを使用する 【製品版のみ】

単純な子要素の除去、タグ名の変換程度であれば PC サイトにマーキングをすることで解決可能です（2.3 PC サイトのマーキング）。単純な子要素の除去、タグ名の変更などでは足りない場合、または PC コンテンツの変更が一切許されないような状況では、次の 2 つの方法が利用できます。

GC パーツより前に GC スクリプトを記述する

GC パーツより先に GC スクリプトを記述し、PC コンテンツの DOM 構造を変更することができます。ただしこの方法の場合 PC コンテンツがグローバルに書き換わってしまい、当該パーツ以外の箇所で予期しない動作をする恐れがあります。

プリプロセスを使用する

GC パーツにはプリプロセスという、GC パーツ実行前に任意の JavaScript 関数を実行させる仕組みが備わっています。この手法では GC パーツ実行前に PC コピーされるため、グローバルに書き換わってしまうことはありません。

ただし現状 GeneCode ビルダーの補助がほぼない手法であり、自前で関数を用意し、さらに直に GC パーツのタグを編集する必要があります。できればセレクタの工夫や、GC スクリプトで解決することをお勧めします。

プリプロセス定義例

```
<gc-script>
// プリプロセス関数の定義例
// 引数には CSS セレクタで指定した箇所が jQuery オブジェクトで渡ります。
gcproc.someFunction = function someFunction (elem) {
  //
  // 何らかの処理
  //
  // 必ず最後に操作後の jQuery オブジェクトを返します。
  return elem;
};
</gc-script>
```

プリプロセス使用例

```
<!-- プリプロセスの呼び出し例 -->
<gc-parts gcid="xxx" name="xxx" view="001">
  <gc-paramList>
    <gc-param name="selector">form:eq(1)</gc-param>
  </gc-paramList>
  <gc-preprocess>
    <gc-function name="gcproc.someFunction">
    </gc-function>
  </gc-preprocess>
  <gc-view>
    <!-- 中略 -->
  </gc-view>
</gc-parts>
```

※試用版ではプリプロセスの定義と使用は同じテンプレートファイル内に記述する必要があります。製品版では外部 JS ファイルに分けることも可能です。

4.2. GC スクリプトを使用する場合

GC スクリプトの実態はサーバー上で動作する JavaScript です。通常ブラウザで使用する JavaScript と同じ構文で自由に DOM を操作・取得できます。さらに jQuery ライブラリが標準で使用できますのでより簡潔に DOM 操作を記述することができます。

GC スクリプトはもっとも柔軟性が高い手法ですが、コーディングが必要な点と、テンプレートが使いまわしづらくなりやすいことから、GC パーツが適用できない場合やごく小さな部分を単純に出力した場合などの限定したケースで利用することを推奨しています。

※gc-script タグ内の JavaScript は GeneCode サーバー上で実行されます。ブラウザ上で実行したい通常の JavaScript は記述しないでください。

4.3. 共通の注意点

GC パーツ、GC スクリプトのどちらを使うにしても、抽出箇所を指定するセレクタは出来るだけ短く記述してください。たとえば「body>div>div.classA」と書くよりも「div.classA」で特定できるならば後者で記述してください。こうすることで PC サイトの HTML 構造を変更に強くすることができます。