



GENECODE

ジーンコード設計・要件定義 ガイド

第 1 版

最終更新日 2016/03/31

目次

第 1 章	開発フロー	4
第 2 章	要件定義	5
(1)	概要	5
(2)	変換による制約	5
(3)	デバイスによる制約	6
(4)	端末の保証についての考え方	6
(5)	デザイン	7
(6)	デザインコンポーネントで PC サイトのアトミックデザイン化	7
(7)	弊社内で持っているコンポーネントのご紹介	8
(8)	注意すべき PC サイトの要素など	8
第 3 章	設計	9
(1)	設計の概要	9
(2)	HTML	9
(3)	パーツ	10
(4)	画像	10
(5)	スタイルシート	11
(6)	JavaScript	11
(7)	HEAD 内要素や META 要素など	12

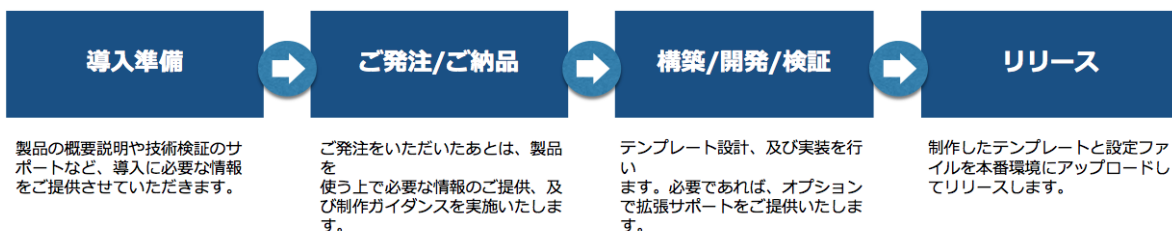
改訂履歴

版数	発行日	改訂内容
第 1 版	2016 年 3 月 31 日	・初版発行

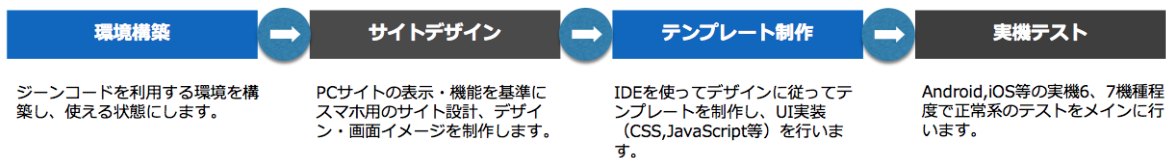
第1章 開発フロー

ジーンコードを使用した開発フローでも、通常の開発フローと同じ流れで開発を進めていきます。

ただし、変換によってコンテンツを作り出す方式のため、省略が可能な項目だけでなく、実装上の制約もあることに注意が必要です。



構築/開発/検証の流れ



第2章 要件定義

(1) 概要

- 対象範囲(URL)の確定
 - PC サイトから変換してスマホサイトと表示する対象ページの確定作業
- ネットワーク設計・監視体制
 - ジーンコードを導入する際のネットワーク設計と監視設計
- URL 設計
 - PC 変換して構築したスマホサイトの URL について(主にドメインの考え方)
- 対象デバイスタイプ(スマホ or タブレット) OS・ブラウザ確定
 - スマホサイトの動作対象端末の剪定
- PC 連動箇所指定
 - スマホサイトと連動したい箇所の洗い出し
- デザイン提案
 - 連動箇所に基づくデザイン作成作業

(2) 変換による制約

変換の都合上、次のような制約があります。

- スマホページと変換元となる PC ページとの関係は 1 対 1 です(脚注 1)。
- 複数の PC ページから 1 つのスマホページを作成することはできません。
- フレーム/IFRAME を使用したページの場合も、各フレーム/IFRAME ごとにスマホ用に変換します。
- スマホの画面フローは PC の画面フローと同じになります(脚注 2)。
- PC サイト側に存在しない新機能は実装できません(脚注 3)。
- フォームコンポーネントの種類(テキストボックス・ラジオボタン等)も、原則として PC ページと同じ種類のコンポーネントを使用します(脚注 4)。

脚注:

1. 変換元ページが存在しない場合に、スマホ独自ページ(静的ページ)を配置することは可能です。

2. 自動で Submit するコードを埋め込んだページに変換することで、ページをスキップすることはできません。
3. PC サイト側を改修し、ジーンコード経由で PC サイトにアクセスした際の動作・表示項目を変更すれば実現可能です。
4. 機能上はコンポーネントの種類変更も可能ですが、既存の JavaScript コードやアプリケーションに影響が出る可能性があります。

(3) デバイスによる制約

機能上はコンポーネントの種類変更も可能ですが、既存の JavaScript コードやアプリケーションに影響が出る可能性があります。

- Flash コンテンツは表示できません
- VBScript/DHTML/Java アプレットは動作しません
- zip ファイルなど圧縮ファイルの展開ができません
- 印刷ができません
- IP アドレスが接続ネットワーク/利用場所等により変動します
- UserAgent ヘッダがスマートフォンの UserAgent になります

(4) 端末の保証についての考え方

ジーンコードは、各 OS やブラウザの表現や挙動の違いを吸収するシステムではなく、すでにあるウェブサイトをマルチデバイス対応させる仕組みを提供する情報ハブとしての役割を果たします。

ジーンコード内部で配布されているパーツは、各 OS やブラウザでの表示に対応しておりますので、パーツを利用するのであれば、保証はされています。

ただ、上級なウェブページを作る上で、パーツよりもよりカスタム要素の強いサイトを構築するのであれば、そのデザインに付随する CSS や JavaScript コードについては、テンプレート開発者が対応 OS やブラウザを設定し自身で対応する必要があります。

(5) デザイン

デザインについては、次のように大枠のデザインポリシーを決めることをお勧めします。

- ヘッダ／フッタ: スマートフォン向けのヘッダ／フッタとして新たにデザイン定義します。メニューはスライドメニュー(フライアウト)、またはドロップダウンメニューとし、表示するメニュー項目は PC サイトと同じメニュー項目を使います。
- 見出し: 全てテキスト化します
- コンテンツ: メインコンテンツ内のすべての情報を漏れなく表示します。画像一覧はカルーセル化します。また、リスト表示部分では詳細説明文を省き、リストの一覧性を向上させます。右サイドフレームの情報は一部を除き、表示しません。
- デザイン: CSS で再現出来る表現を基本とし、フラットなデザインを心がけること。
- 画面構成: 複雑な表組み・レイアウトはさけ、シンプルな構成とすること。

(6) デザインコンポーネントで PC サイトのアトミックデザイン化

- アトミックデザイン
 - これ以上分解できないところまで要素を分解する = Atom
 - ちょっとした組み合わせ(ログインフォーム、検索フォーム、フォームでいう姓名など) = Molecule
 - 有機的な何か(ヘッダーやフッターなど) = Organisme
- 定義
 - Atom = 原子
 - ボタン
 - リンク
 - 文字
 - Molecule = 分子
 - 検索ボックス
 - ログインフォーム
 - Organism = 微生物・単細胞生物
 - カルーセル
 - フライアウトメニュー
 - など
 - Templates = ジーンコードでいうパターンファイルに近い
 - Pages

対象となるページをすべてデザインするのではなく、似たようなデザインを探し、リファレンスを作る方式で、作業の効率化を図ります。

(7) 弊社内で持っているコンポーネントのご紹介

別紙「GGUI デザインコンポーネント」を参照

(8) 注意すべき PC サイトの要素など

- 画面幅 400px (iPhone 6 相当) を標準的なスマートフォンの画面幅とした場合、画面内に収まらない要素は、構造を大きく変化させる必要がある
 - テーブル
 - バナー画像
- クライアントサイドで構築されている DOM 構造 / AJAX など
- 文字を含む大きな画像は、小さく表示することで文字の可読性が著しく低くなってしまいます
- ID やクラス属性が適切に振られておらず、jQuery セレクターで要素の特定が難しい場合
- IFRAME
- モーダル
- ポップアップウィンドウ (スマホではタブになるため)
- window.opener は iOS の Chrome で使用できません

第3章 設計

(1) 設計の概要

ジーンコードを使用するにあたって必要な設計とは、主に以下 2 点を決定していく作業になります。

- PC サイト連動箇所: デザインを実現するにあたり、PC ページのコンテンツの内、スマートフォン側で利用するコンテンツブロック/フォーム/画像/JavaScript コード/meta タグ等はどれか? また、どのように変換するか?
- スマートフォン固有箇所: デザインを実現するにあたり、PC ページ側に情報が存在しないために、スマートフォン向けに別途作成する必要があるか? 利用したりするスタイルシート/画像/HTML ブロック等は何か?



(2) HTML

スマートフォン用 HTML をどのように生成するかを決定します。スマートフォン用 HTML は主に以下の 4 つの HTML ブロックを組み合わせて出力します。

- スマートフォン用 HTML タグ(直接記述)
- 対応する PC ページから取得した HTML ブロック
- 対応する PC ページから取得し、パーツで変換した HTML ブロック
- 対応する PC ページから取得し、スクリプト(gc-script)で変換した HTML ブロック

また、PC ページの HTML ブロックはセレクトア(※1)を利用して指定するため、変換対象ブロックを特定できるセレクトアが存在するかを調査し、セレクトアの書き方を決定します(※2)。

これらは最終的にはテンプレートに記述する内容になります。

フレーム/IFRAME の場合は、各フレームの HTML 毎に生成方針を検討します。

※1 変換対象の参照方法としては、jQuery のセレクトア構文を使用します。

※2 変換対象を特定するための id 属性/class 属性が存在しない場合は、PC ページに目印となる属性を追加することも検討します。

(3) パーツ

パーツは、PC サイトの構造とスマートフォンサイトで使用する CSS フレームワークを考慮し、どのパーツを使用すれば変換できるかを決定します。代表的なページに対し、事前にプロトタイプ作成を行うことをお勧めします。

(4) 画像

画像については以下のいずれの変換方式をとるかを検討します。

- リサイズ: PC と同じ画像ファイルを使用しますが、CSS で大きさの調整を行います。
- 置き換え: IMG 要素の src 属性を変換し、別途作成したスマホ用の画像ファイルに置換する方法です。
- テキスト化: img 要素の alt 属性を利用してテキストに置き換え、CSS でデザインする方法です。見出し画像やボタンに適した方法です。

- 削除: スペーサー画像などに適した方法です。

その他、画像変換でクリッピングする方法もあります。

(5) スタイルシート

利用・作成するスマートフォン向け CSS を検討します。CSS は PC サイトからの変換ではなく、スマートフォン向けに準備・作成した CSS を利用します。

ジーンコードでは以下の CSS を標準提供しています。

- `gcreset.css`: 各種プロパティをリセットする CSS です。
- `gccommon.css`: フォーム等に基本的なデザインを適用する CSS です。
- `gcparts.css`: パーツで変換した部分に適用する CSS です。
- `gcbs-color.css`: Bootstrap パーツのカラーテーマです。

この他、独自の CSS を利用することができます。

また、使用する CSS は `link` タグでテンプレートに記述します。

なお、Bootstrap パーツや jQueryMobile パーツを使用すると、各フレームワークの書式に則った構文で HTML 出力されるため、Bootstrap のテーマ等の利用も可能です。

(6) JavaScript

JavaScript については以下の観点で JavaScript の扱いを検討します。

- PC サイトの JavaScript (外部 JS ファイル、埋め込み `script` タグ、`onclick` 属性等) は基本的にスマートフォンサイトでも全て利用します。
- ジーンコードでは jQuery 1.7.2 を使用します。PC サイト側で使用する jQuery のバージョンと異なる場合は、どちらか一方に合わせるか、共存を検討します。

- JavaScript コードの内容によっては、ポーティングもしくは JS 変換 / PostJS を検討します。代表的なケースは以下の通りです。
 - PC サイトとスマートフォンサイトの URL が異なる環境で、JavaScript コードに PC サイトの URL が記載してある場合
 - HTML 変換に伴い、JavaScript コードに記載しているセレクタではオブジェクトが参照できなくなる場合
 - JavaScript コードが PC サイト向けのデザインを出力する場合

(7) HEAD 内要素や META 要素など

サイトの方針にもかなり左右されますが、以下のことにも考慮をする必要があります。

- Script タグなど読み込むスクリプトを考慮する: HEAD タグ内のみではなく、BODY などにもあるかをしっかりチェックすべき
- Google Analytics などサイトの分析用のスクリプトを入れている場合、そのタグを本サイトと共有するか、別に管理するかを検討ください。
- SEO 向けに入れている要素の検討
- EFO スクリプト: フォームの要素やタグ構成が変わらないケースでは、EFO スクリプトはそのまま流用することができますが、構造が変わってしまう場合には、EFO 業者に確認が必要になります。
- ドメインが本サイトと異なる場合、アプリケーション内で絶対 URL を出力しているケースはアプリケーションの修正が必要な場合もあります。
- IFRAME やモーダル表示などは注意が必要です。