

# jQuery Mobile

スマートフォンサイト・デザイン

Webクリエイターが身につけておくべき  
新・100の法則。

石原 悠 著



## スマホサイト制作に 手間をかけていませんか？

上手に活用して  
サイト制作を効率化

動くプロトタイプを  
手間なく伝える

コンポーネントの  
デザインを改善する

フラットデザイン対応の最新バージョン1.4対応！

サンプルデータ  
ダウンロード



## 著者プロフィール

### 石原 悠 (いしはら ゆう)

会社員として複数の会社でディレクター、デザイナーの経験を経て、2012年1月にフリーランスに。Webサイト、iOS、Androidアプリのデザインや実装、ロゴや印刷物の制作など幅広い業務を大小問わず請け負っており、jQuery Mobileなどのフレームワークを用いた構築の実績も多い。

デザインだけでなく、プログラミングまですべてを手がけたiOSアプリを複数リリースするなど実装方面の制作も得意としており、Webサイトやアプリの構築に関してさまざまな角度から得た経験と知識を生かした柔軟でスピーディーな対応を得意としている。

●normo(個人事業Webデザイン、iPhoneアプリ制作請負) <http://normo.jp>



本書は2014年5月現在の情報をもとに解説しています。本書の発行後に各Webサイトの内容や仕様、jQuery Mobileのバージョンなどが変わっていることがあります。あらかじめご了承ください。  
本文中の製品名およびサービス名は、一般に各開発メーカーおよびサービス提供元の商標または登録商標です。なお、本文中にはTMおよび©マークは明記していません。

## はじめに

Webサイトの運営をされている方なら、アクセス数の推移を見てすでにご存知かもしれませんが、近年Webサイトへのアクセスの比重がPCからモバイルのほうへと推移しています。

2014年には「モバイルファースト」というキーワードをよく耳にするようになりました。

これまでWebサイトの制作業務では、まずPC用のWebサイトを制作して、余った時間や予算でモバイルサイトを構築する、という流れで進められてきました。それが「まずモバイルで使えるサイトを重点的に作る」という新しい流れも生まれてきているようです。

そうした状況に応じて、モバイル向けのサイト制作が容易になるツールとしてjQuery Mobileをはじめ、Twitter Bootstrapやレスポンシブデザインなど便利で高機能なフレームワークや技術が日々進化してきています。

この本を手にとられた方は、おそらくこれから新しくWebサイトを制作されるにあたってその中からjQuery Mobileの導入を検討されているかと思います。

jQuery Mobileは数あるモバイルとPCの両方のWebサイトを構築するフレームワークの中で、学習コストも低く、効率的で柔軟性にも優れたWebサイトが開発できる非常に便利なフレームワークになっています。そのサイト制作にこの本が少しでもお役に立てたら幸いです。

2014年5月

石原 悠

# 目次

はじめに	3
本書の読み方	8
序章	9

## 第1章 jQuery Mobile の基本を知る法則

法則		
基本的な使い方を知る	1 ページの設定と構成要素を理解する	18
	2 Ajaxを使ったページ遷移を理解する	20
	3 Ajaxによるページ遷移の不具合を解消する	22
ボタン要素を理解する	4 jQuery Mobileに用意されているボタンを把握する	23
	5 ボタンにアイコンを表示する	25
	6 ページのヘッダに「戻る」ボタンを表示する	27
	7 複数のボタンをグループ化してユーザビリティを高める	29
ツールバーを理解する	8 ヘッダを固定して最大限に有効活用する	31
	9 フッタのレイアウトを調整して見やすくする	33
	10 複数ページにまたがるコンテンツを自由に行き来する	35
リストで情報を整理する	11 リストを使ってコンテンツを階層化する	37
	12 読み込み専用のリストを使用する	41
	13 リストに自動で区切りを挿入する	42
	14 新着リストに更新時間を表示する	44
	15 画像と簡単な情報をまとめた商品一覧ページを作成する	45
	16 下層ページの情報量をリストに表示してユーザーのストレスを軽減する	46
スマートフォンのフォームを理解する	17 状況に合わせた文字入力フォームを選択する	47
	18 スマートフォンらしいタッチパネルを生かしたフォームを利用する	49
	19 選択入力可能なメニューを利用する	51
フォームを活用する	20 必須項目が入力されたら「送信」ボタンをアクティブに変更する	52
	21 基本要素を理解してユーザーフレンドリーなアンケートページを作成する	54
	22 買い物カートを利用したECサイトを作成する	56
章末コラム	iPhoneとAndroidの違い	60

## 第2章 基本的なUIパーツを使いこなす法則

法則		
フォームを使いやすくする	23 検索フォームにプレースホルダーとフィルタリング機能を使う	62
	24 オートコンプリート機能を利用してユーザーの入力をサポートする	63

ダイアログボックスを  
カスタマイズする

ポップアップを  
カスタマイズする

ページレイアウトを  
工夫する

章末コラム

25	フォームの入力内容をあらかじめ指定する	64
26	ダイアログボックスの閉じ方やトランジションをカスタマイズする	66
27	ダイアログボックスの「閉じる」ボタンの位置を変更する	68
28	サイトが単調にならないようにダイアログボックスを使い分ける	69
29	ポップアップウィジェットを使用する	71
30	ポップアップウィジェットの位置や見栄えを変える	75
31	オーバーレイポップアップウィジェットを使用する	78
32	パネルウィジェットを使用して複雑なサイト構造を一覧できるようにする	80
33	パネルウィジェットにリストを設置してより多くのリンクを表示させる	82
34	グリッドレイアウトを使った緩急のあるサイトを作る	83
35	レスポンシブグリッドでモニタサイズや端末の向きに応じたレイアウトにする	85
36	レスポンシブテーブルでモニタサイズや端末の向きに応じたレイアウトにする	87
37	カラムトグルのチェックボックスを使ってコンテンツの表示・非表示を切り替える	89
38	開閉式コンテンツを利用して情報を効率的に格納する	90
39	開閉式コンテンツを複数並べてアコーディオンメニューにする	92
40	開閉式コンテンツのアイコンを変更する	93
	CSSのマルチクラスの記述方法	94

## 第3章 ページレイアウトをカスタマイズする法則

法則

カスタマイズの  
基本を理解する

ヘッダとフッタを  
活用する

画像を使ってカスタマイズする

スタイルシートを使って  
カスタマイズする

41	複雑になりがちなるリストを見やすくする	96
42	初期状態の2つのテーマを理解する	97
43	自分で新しいテーマを作成してページに反映する	98
44	ページ遷移のエフェクトを理解して状況によって使い分ける	99
45	ヘッダとフッタを画面外に移動させ、フルスクリーンモードで読みやすくする	100
46	サイト全体に共通でヘッダ、フッタを固定表示する	101
47	ヘッダに複数のボタンを設置してページ全体の情報量を増やす	102
48	レイアウトを工夫してフッタを活用し、サイトをより見やすくする	104
49	ヘッダメニューをプルダウンメニューに変更してボタン数を増やす	106
50	プルダウンメニューに一手間加えて操作感を向上させる	109
51	ページ遷移時に表示されるローダーをオリジナルのものにする	110
52	省略されて「...」と表示される長いテキストを省略しないようにする	112
53	「ホーム画面に追加」をした際のサイトのアイコンをカスタマイズする	114
54	背景にパターン画像を敷いてjQuery Mobileサイトの見栄えを変える	116
55	タブナビゲーションに通知件数アイコンを入れる	117
56	端末の種類や回転方向、解像度を識別してスタイルを反映する	119

JavaScriptを使って  
カスタマイズする

57	オリジナルアイコンを作成してサイトのオリジナリティを高める	122
58	CSS3のグラデーションを利用した立体的なボタンを作成する	124
59	CSS3のシャドウを利用した立体的なボタンを作成する	126
60	グローバル設定を利用してサイト全体のレイアウトを統一する	128
61	すべてのページにTwitterとFacebookへの投稿ボタンを設置する	130
62	デフォルトの角丸とドロップシャドウを削除してシンプルなページを作成する	132
63	ページトランジションを自分でカスタマイズする	134
章末コラムCSSやCSS3の活用		136

## 第4章 スマートフォンならではの ジェスチャーを生かす法則

イベント取得時の  
カスタマイズをする

法則			
64	スマートフォン独自のジェスチャーに対するイベントを取得する	138	
65	画面の回転のタイミングを取得して縦長画像と横長画像を入れ替える	141	
66	背景色をスクロール位置に応じて変える	143	
67	ページのロードが完了した結果で成功と失敗のメッセージを表示する	145	
68	ページを移動するたびにランダムに画像を切り替える	147	
69	スクロール位置が変わったときにページトップボタンを表示する	149	
70	どのリンクから移動してきたかによってページ上の文章を変える	151	
71	複数の開閉パネルを同時に開いたり閉じたりする	154	
章末コラム		マウスで操作するサイトとタッチ端末で操作するサイト	156

## 第5章 フレームワークやプラグインを活用する法則

フレームワークや  
プラグインを使う

オリジナルの  
プラグインを作成する

テーマをカスタマイズする

既存サイトを  
ブラッシュアップする

法則			
72	スマートフォンサイトでGoogleマップを表示する	158	
73	Googleマップにマーカーを付けたりクロースアップしたりする	160	
74	端末をシェイクして画面を変える	162	
75	スワイプで次の画像に移動するフォトギャラリーを作る	164	
76	入力された郵便番号をもとに住所を自動表示する	166	
77	リスト要素の右肩にバッジを追加する	169	
78	データロードの成功時と失敗時に状況に合わせたアラートを表示する	171	
79	Webサイト上で効果を見ながらjQuery Mobileのテーマを作成する	173	
80	サードパーティ製のデザインを探してページに反映する	177	
81	Twitter Bootstrap風のデザインにする	178	
82	jQuery Mobile独自のデザインを利用しないようにする	179	
章末コラム		オープンソースのプラグインの選び方	180

## 第6章 WordPress & EC-CUBE を カスタマイズする法則

	法則	181
RSSリーダー作成の応用	<b>83</b> YouTube RSSフィードからJSONをロードして一覧表示する	182
	<b>84</b> 検索結果の一覧にサムネイル画像を表示する	184
	<b>85</b> 作成したリストをタブで切り替えられるようにする	185
	<b>86</b> 通知プラグインを使用して、RSSのロードに失敗したらアラートを表示する	186
WordPressのカスタマイズ	<b>87</b> WordPressで構築したブログを端末に合わせて振り分ける	188
	<b>88</b> WordPressに対応したjQuery Mobileテーマを作成する	190
	<b>89</b> 検索サイトからリンクしてきたときにもトップページに戻れるようにする	192
	<b>90</b> 「前の記事へ」「次の記事へ」リンクをテキストからボタンに変更する	193
	<b>91</b> パネルウィジェットに固定リンクや外部リンクを格納する	194
	<b>92</b> 縦に長くなりがちなページをコンパクトにまとめる	196
	<b>93</b> jQuery MobileでGoogleアナリティクスを使ってアクセス解析を行う	197
	<b>94</b> ブログをソーシャルメディアに対応させ、SNSのボタンを設置する	199
	<b>95</b> WordPressとjQuery Mobileで作ったブログに自作テーマを適用する	201
EC-CUBEのカスタマイズ	<b>96</b> EC-CUBEのメニューをスマートフォン向けにコンパクトにまとめる	202
	<b>97</b> ポップアップウィジェットを使用してログイン関連のメニューをコンパクトにまとめる	205
	<b>98</b> ECサイトにローテーションバナーを設置して訪問者の視線を誘導する	207
	<b>99</b> 「おすすめ商品」をスライドバナーからリストビューに変更する	209
	<b>100</b> 限定商品に在庫数が表示されたバッジを付けて目立たせる	210
章末コラム	SassとCompassで作成するjQuery Mobileテーマ	211
用語集		212
索引		217

# 本書の読み方

## ●法則のタイトル

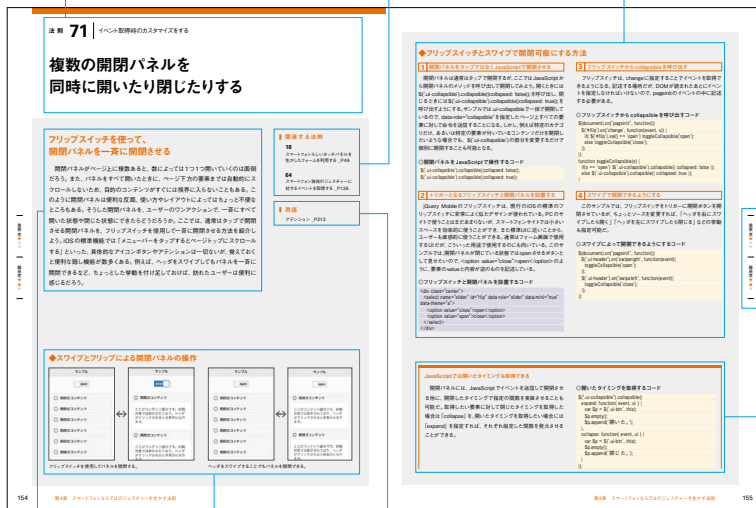
やるべきことがすぐにわかる  
タイトル付けをしています。

## ●関連する法則

解説している法則と密接にかかわる  
他の法則を紹介しています。続けて  
読むことで、理解が深まります。

## ●Webサイトの制作手順

Webサイトの制作手順を、詳細な  
説明とともにサンプルコードを掲載  
しています。



## ●重要度・難易度

法則の重要度と難易度を3段階  
に分類しています。法則を実践  
するときの目安にしてください。

## ●ポイント

法則に関連する知って  
おくべき知識について  
説明しています。

## ●解説

法則をしっかり習得で  
きるように、ていねい  
に解説しています。

## ●画面例

豊富な画面例によっ  
てわかりにくいところの  
理解が深まります。

## ●用語

わかりにくい用語は  
巻末の用語集で調べ  
られます。

## 本書掲載のソースコードについて

本書で説明しているサンプルコードと画像ファイルは、本書のサポートページからダウンロードできます。

### ◆本書サポートページ

<http://www.impressjapan.jp/books/1111101131>

本書では、jQuery MobileによるWebサイトの構築を実践するうえで必要な100のノウハウを紹介しています。まずはじっくりと解説を読んで、それから図解や実例、具体的な記述方法などを読み進めることで、確実に理解を深められます。

※ここで紹介している紙面はイメージです。実際の本書紙面とは異なります。



# 序章

2014年にバージョン1.4として公開されたjQuery Mobileの特徴や主な変更点、ダウンロードしてから開発するまでの環境の構築方法などを解説する。

## デザインが刷新された新バージョン1.4

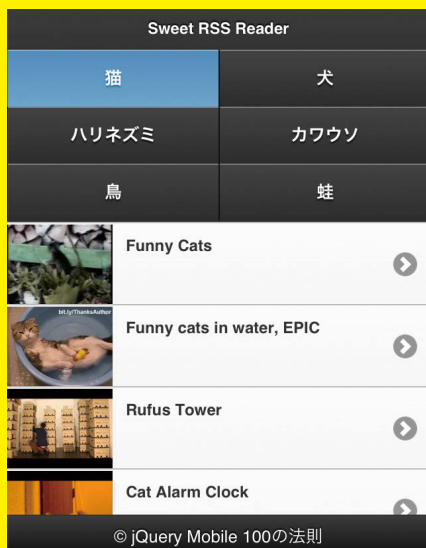
### ●2014年1月にバージョン1.4がリリースされた

jQuery Mobileは2010年に公開されて以来バージョンアップを繰り返し、本書の執筆時点ではバージョン1.4が公開されている。

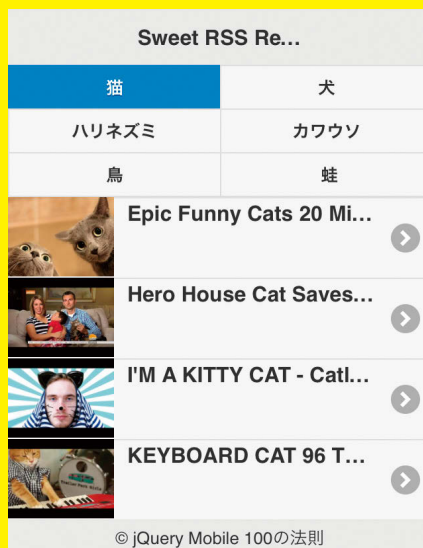
バージョン1.3まではバージョンアップごとに新しくウィジェットなどの機能面が追加、強化されることが多かったが、2014年の1月に公開されたバージョン1.4で

は、デザインの刷新と使い方の変更が大きかった。

デザイン面では、iOS 7をイメージしたフラットなデザインに変更され、また使い方の面では、前バージョンまではデータ属性でスタイルを指定していたが、新しいバージョンからは直接クラスセレクタを指定してデザインを反映させるようになった。



バージョン1.3までのカラフルなデザイン。



iOS 7を想起させるフラットなデザイン。

## ●フラットデザインの採用

これまでは、スキューモーフィックデザインといわれる、現実世界に実際にあるものをモチーフにした立体感や光沢があるデザインが流行していた。それが2013年あたりから、次第にシンプルでフラットなデザインが採用されることが増えてきた。

jQuery Mobileの1.3までのバージョンアップでも、ところどころデザインの見直しがされていてその兆しはあったが、バージョン1.4で大幅にほぼすべてのパーツにおいてフラットデザインが採用されることとなった。

新しいフラットなデザインのjQuery Mobileは、これまでデフォルトで用意されていた5つのテーマを使い分けていたのが、白と黒の2つのテーマのみとなった。また、バージョン1.3までのテーマAは黒ベースのものだったが、バージョン1.4から白ベースのテーマがAとして採用されている。

## ●デザインの指定方法の変更

これまでは、UIにデザインを指定する際、データ属性を使用して「data-role="button"」などとしてボタン要素として指定していた。バージョン1.4ではこの指定方法が撤廃されて、クラスセレクタを直接使用するようになった。

バージョン1.3までは、複数のスタイルを指定したいときにはデータ属性を「data-role="button"」「data-mini="true"」などと複数記述しなければならなかったのだが、バージョン1.4からは、「class="ui-btn ui-btn-mini"」など、

直接クラスセレクタを指定してデザインを適用することが可能になった。これにより、HTMLファイルが随分すっきりして、可読性が上がった。

◎バージョン1.3以前のボタンの指定方法のコード

```
<a href="#" data-role="button" data-inline="true">ボタン</a>
```

◎バージョン1.4からの新しいボタンの指定方法のコード

```
<a href="#" class="ui-btn ui-btn-inline">ボタン</a>
```

## ●アイコンの描画がSVG形式に変更

jQuery Mobileは、デフォルトの状態では数多くのプリセットのアイコンが用意されていて、簡単な指定で使えるのが魅力の1つだったのだが、バージョン1.4からはPNG形式とSVG形式の2種類を状況によって使い分けられるような仕様に変更になった。

SVG形式で描画されたアイコンは、PNG形式と異なり拡大してもぼやけないので、拡大して大きく扱う場合でもきれいなエッジで描画することが可能だ。今後、Retinaディスプレイ以上の解像度の画面が登場したときでも問題ないだろう。

他にもバージョン1.4では、グローバル設定やメソッドの呼び出し方が変わったり、依存しているjQueryのバージョンが上がったりして使い勝手が変わっているので、本書を参考にいろいろと試してほしい。今後は、階層メニューやポップアップウィジェットなどが非推奨になる可能性もあり、次第に廃止されていくと考えられる。

## ECサイト構築のために最低限必要なファイル

### ●ソースコードをダウンロードする

jQuery Mobileの公式サイトにアクセスし、「Download jQuery Mobile」の「Latest stable」をクリックしてフルパッケージのソースコードをダウンロードする。本書執筆時点（2014年5月）では、jQuery Mobileの公式サイトからは安定版のバージョン1.4.2がダウンロードできる。

- ・「Latest stable」：安定版の最終リリース版
- ・「Custom download」：必要な機能のみを選択してダウンロード可能

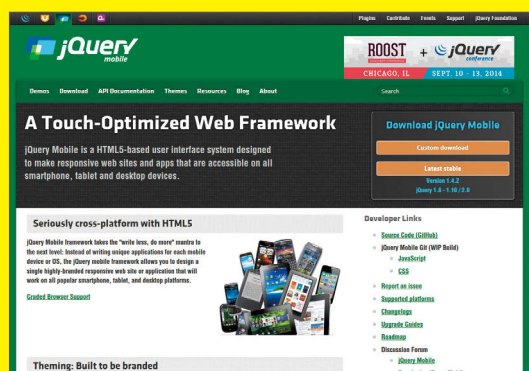
本書ではフルパッケージのバージョン1.4.2を使用する。バージョン1.4.2では、いくつかの新機能や専用のアイコン群が追加され、ユーザーにとってより快適なサイト制作ができるようになっている。また、トップページの「Download」をクリックすると、さまざまなダウンロードリンクがあり、それぞれ次のような内容のファイルが用意されている。

- ・ CDN-Hosted JavaScript : JS ファイル
- ・ CDN-Hosted CSS : CSS ファイル
- ・ Copy-and-Paste Snippet for CDN-hosted files: CDNを使用する場合のコピーペースト用のスニペット
- ・ ZIP File : デモを含めたファイル一式

ダウンロードできるファイルには、それぞれにUncompressed

版とMinified版があり、選べるようになっている。Uncompressed版は、デバッグ用なのでソースコードにコメントが付いていて可読性が高く、ソースコードを解読して内容を理解しやすい。

また、業務などで実際に使用する場合は、軽くてサーバーやクライアントPCへの負荷が低いMinified版や、すでにインターネット上で配信されているCDN版を使用したほうがいいだろう。



◆jQuery Mobile公式サイト <http://jquerymobile.com/>

### ●ファイルの中身を確認する

本書では、トップページにある「Latest stable」をクリックしてすべてのファイルをダウンロードしたうえで、順を追って説明していく。ZIPファイルをダウンロードして展開すると、次の図のような構成でファイルが格納されている。

```
- demosディレクトリ
- imagesディレクトリ
- jquery.mobile-1.4.2.css
- jquery.mobile-1.4.2.js
- jquery.mobile-1.4.2.min.css
- jquery.mobile-1.4.2.min.js
- jquery.mobile-1.4.2.min.map
- jquery.mobile.external-png-1.4.2.css
- jquery.mobile.external-png-1.4.2.min.css
- jquery.mobile.icons-1.4.2.css
- jquery.mobile.icons-1.4.2.min.css
- jquery.mobile.inline-png-1.4.2.css
- jquery.mobile.inline-png-1.4.2.min.css
- jquery.mobile.inline-svg-1.4.2.css
- jquery.mobile.inline-svg-1.4.2.min.css
- jquery.mobile.structure-1.4.2.css
- jquery.mobile.structure-1.4.2.min.css
- jquery.mobile.theme-1.4.2.css
- jquery.mobile.theme-1.4.2.min.css
```

なおjQuery Mobile本体のJavaScriptファイルとして、次の2つがある。

```
jquery.mobile-1.4.2.js
jquery.mobile-1.4.2.min.js
```

ファイル名に「.min」という文字が入っているものが、前述したMinified版にあたる圧縮されたソースコードで、入っていないものがUncompressed版にあたる圧縮前のソースコードである。

そして、その他にjQuery Mobileサイトを装飾する次のCSSが入っている。

```
jquery.mobile.external-png-1.4.2.css
jquery.mobile.icons-1.4.2.css
jquery.mobile.inline-png-1.4.2.css
jquery.mobile.inline-svg-1.4.2.css
jquery.mobile.structure-1.4.2.css
jquery.mobile.theme-1.4.2.css
jquery.mobile-1.4.2.min.map
```

このうちファイル名に「.structure」という文字が入っているものは、ページ遷移など基本となる動作部分のみが記載され、ナビゲーションバーのカラーリングやアイコンなどの装飾部分が省かれたファイルである。通常は「jquery.mobile-1.4.2.css」ファイルのみ使用すればいいが、自作のテーマを使うときなどは各テーマのスタイルが記述されている「.theme」と「.structure」を合わせて使用することになる。

また、バージョン1.4.2から新たに追加されたアイコン用のファイルも入っており、SVGやPNG形式のファイルがそれにあたる。またミニファイされたJavaScriptファイルをデバッグするために使用するmapファイルなども追加されている。

このように多くのファイルがあるのだが、実際に仕事な

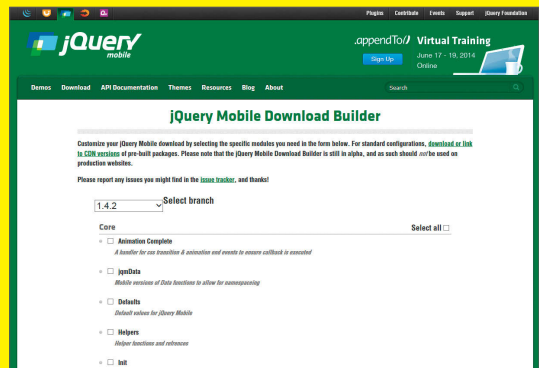
どで使用する場合はすべてのファイルが必要なわけではない。本書では最低限必要とされる次の3つのファイルを使用して話を進めていく。

images ディレクトリ  
jquery.mobile-1.4.2.css  
jquery.mobile-1.4.2.js

また、Mobile版ではないjQuery 本体の「jquery.js」も必要となる。jQuery Mobile 1.4.2では、jquery.jsのバージョンは1.10.2が使用されているが、1.8～1.10/2.0などの幅広いバージョンがサポートされているようだ。

ダウンロードページには、最新のバージョンである1.4以外にも、以前のバージョンもいくつかダウンロードできるようになっている。ただし、対応しているjQueryのバージョンが異なることに注意が必要だ。

jQuery Mobileサイトには、必要なモジュールを選択してダウンロードできる「jQuery Mobile Download Builder」が用意されている。



jQuery Mobile Download Builderを使うと、ダウンロードすべきファイルが整理されているので非常に便利だ。

## オフラインの開発環境の構築

### ● オフライン環境に必要なファイルをヘッダに記述する

jQuery Mobileを使ったサイト制作に必要なファイルを読み込むため、HTMLファイルのヘッダ部分を修正する。この記述方法は2つあるので、用途に合わせて選択してほしい。1つ目の方法では、ダウンロードしたjQuery Mobile関連のファイルをローカルPCのディレクトリ内に格納して使用する。オフラインでのサイト制作の際には、後述する2つ目の方法では動作確認ができないので、この方法を選択しなければならない。

次に、HTMLファイルが入っている階層に「/jquerymobile」ディレクトリを作成し、その中に次に示す3ファイルを格納し、「/images」ディレクトリを作成する。

```
jquery.mobile-1.4.2.min.css
jquery.mobile-1.4.2.min.js
jquery.js
/imagesディレクトリ
```

jQuery Mobileは基本的にHTML5で記述するためtype属性が省略でき、type="text/css"やtype="text/javascript"を記載する必要はない。

### ◎ オフライン環境の場合のコード

```
<link rel="stylesheet" href="../jquerymobile/jquery.mobile-1.4.2.css" />
<script src="../jquerymobile/jquery.js"></script>
<script src="../jquerymobile/jquery.mobile-1.4.2.js"></script>
```

### ● オンライン環境では、CDNで提供されているファイルを使用する

必要なファイルをヘッダに記述する2つ目の方法は、コンテンツデリバリーネットワーク (Contents Delivery Network、CDN) で提供されているファイルを使用して、サーバーから直接ソースコードを読み込む方法だ。なお、コンテンツデリバリーネットワークとは、デジタルコンテンツをインターネット経由で配信するために最適化されたネットワークのことである。

この方法は、準備が不要でレスポンスもいいため、公式サイトではこちらが推奨されている。筆者もディレクトリに余分なファイルを入れたくないのでCDNのファイルを使用することがほとんどだが、オフライン環境では使えないため、本書では、オフライン環境で使える1つ目に紹介した方法で解説する。

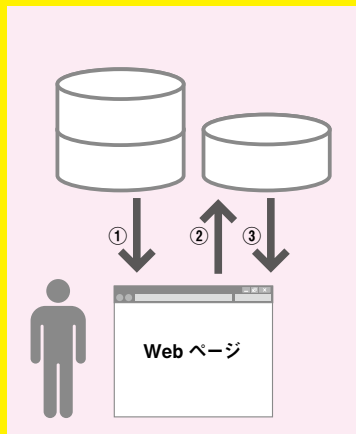
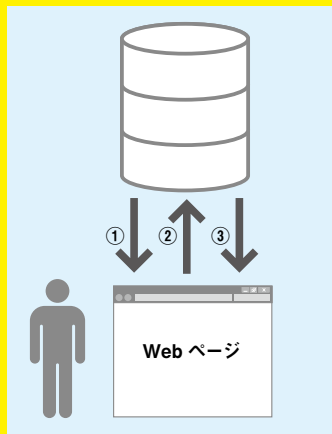
### ◎ オンライン環境の場合のコード

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.2/jquery.mobile-1.4.2.min.css" />
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="http://code.jquery.com/mobile/1.4.0/jquery.mobile-1.4.2.min.js"></script>
```

## ●CDNを利用しない場合と利用する場合のサイト閲覧の違い

通常は左図のようにサーバーと1対1でリクエストとレスポンスが繰り返される。しかし、CDNを利用して、右図のようにサイト内で使用する一部のデータを地理的およびバックボーン的に分散させた別のサーバーから受信すれば、負荷が分散されより効率のいい安定したサイトの配信が可能になる。

◎ 通常のネットワークを使用したサイトの閲覧（左図）と、CDNを使用したサイトの閲覧（右図）



- ①サーバーからHTMLデータが送信される
- ②HTMLを解析し、画像やJavaScriptなどの別ファイルのデータをサーバーにリクエストする
- ③リクエストのあった画像やJavaScriptが送信される



## jQuery Mobileの基本を知る法則

jQuery Mobileを使用するにあたって必要になる知識として、jQuery MobileによるサイトでのHTMLの書き方から、ページ遷移などの仕組み、ボタンやリストなどの基本的なUIの使い方までを解説する。

## 法 則

- 1 ページの設定と構成要素を理解する ..... 18
- 2 Ajaxを使ったページ遷移を理解する ..... 20
- 3 Ajaxによるページ遷移の不具合を解消する ..... 22
- 4 jQuery Mobileに用意されているボタンを把握する ..... 23
- 5 ボタンにアイコンを表示する ..... 25
- 6 ページのヘッダに「戻る」ボタンを表示する ..... 27
- 7 複数のボタンをグループ化してユーザビリティを高める ..... 29
- 8 ヘッダを固定して最大限に有効活用する ..... 31
- 9 フッタのレイアウトを調整して見やすくする ..... 33
- 10 複数ページにまたがるコンテンツを自由に行き来する ..... 35
- 11 リストを使ってコンテンツを階層化する ..... 37
- 12 読み込み専用のリストを使用する ..... 41
- 13 リストに自動で区切りを挿入する ..... 42
- 14 新着リストに更新時間を表示する ..... 44
- 15 画像と簡単な情報をまとめた商品一覧ページを作成する ..... 45
- 16 下層ページの情報量をリストに表示してユーザーのストレスを軽減する ..... 46
- 17 状況に合わせた文字入力フォームを選択する ..... 47
- 18 スマートフォンらしいタッチパネルを生かしたフォームを利用する ..... 49
- 19 選択入力可能なメニューを利用する ..... 51
- 20 必須項目が入力されたら「送信」ボタンをアクティブに変更する ..... 52
- 21 基本要素を理解してユーザーフレンドリーなアンケートページを作成する ..... 54
- 22 買い物カートを利用したECサイトを作成する ..... 56

# ページの設定と構成要素を理解する

## head 要素に必要な項目を記述する

まずは単体のページを作成して、ヘッダに必要な項目をリストアップする。  
<head>～</head>には、序章で読み込む設定をしたjQuery Mobileのプログラム以外に、次の3つの項目の記述が必要になることを覚えておこう。

- ・ 文字コードの指定
- ・ ページタイトル
- ・ Viewportの設定

ここで選択する文字コードは、ソースコード本体やjQuery Mobileの公式サイトでも使用されているのと同様にUTF-8を使用するため、<meta charset="utf-8">と記述する。

Viewportという項目では、スマートフォンでページを開いた際のページ幅や、ピンチ操作などで拡大可能な最大幅を指定する。jQuery Mobileを使用したサイトの制作では、横幅をスマートフォンのブラウザにフィットさせるため、このViewportの記述1行を加える必要がある。

### 関連する法則

88

WordPressに対応したjQuery Mobile  
テーマを作成する\_P190

### 用語

Viewport\_P213  
データ属性\_P215

◎<head>～</head>に記述するコード

```
!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
<title>jQuery Mobile 100の法則：サンプルプログラム</title>

<link rel="stylesheet" href="../jquerymobile/jquery.mobile-1.4.0.css" />
<script src="../jquerymobile/jquery.js"></script>
<script src="../jquerymobile/jquery.mobile-1.4.0.js"></script>
</head>
```

## ボディに必要な項目をリストアップする

次にbody要素内部だが、最初のdata-role属性に下のHTMLソースの①のように"page"と指定し、jQuery Mobileサイト内での1ページ単体のくくりとする。そして、この中にページを構成するさまざまな要素を配置してページを作成していく。

jQuery Mobileでは、HTML内に記述したひとまとまりの要素を、HTML 5の独自データ属性「data-」を使用して「data-role="〇〇"」と指定することで、各パーツとしての機能を付与させる仕組みになっている。ここでは、data-role="page"を指定することで、jQuery Mobileがその中の要素をまとめて1つのページとして認識するように指定している。

◎<div>～</div>に記述するコード

```
<div data-role="page">
  <div data-role="header">～</div>           ←②
  <div role="main" class="ui-content">～</div> ←③
  <div data-role="footer">～</div>          ←④
</div>
```

### ◆ヘッダ、コンテンツ、フッタの役割



ヘッダ部分には上のHTMLソースの②のように、div要素にdata-role="header"というデータ属性を指定して記述する。このheader属性はサイト全体で常に表示される共通のパーツであり、「コンテンツタイトル」「ホームボタン」「戻るボタン」などを設置するために使用するエリアとなる。

コンテンツ部分には上のHTMLソースの③のように、div要素に<div role="main" class="ui-content">という値をそれぞれ指定し、その内容を記述する。ここでは「タイトル」「本文」「画像」「リスト」「子階層へのリンク」など、実際のコンテンツの内容を配置する。本書掲載の多くの要素は、この中で使用するものとなる。

フッタ部分には上のHTMLソースの④のように、div要素にdata-role="footer"というデータ属性を指定してその内容を記述する。フッタはページ下部に表示されるエリアで、一般的には「コピーライト」「利用条件」「免責事項」「PCサイトなどへのリンク」を設置することが多い。ただスマートフォンの特性上、表示面積が狭くあまり多くのパーツが設置できないので、コピーライトのみを記載しているサイトも少なくない。また、タブナビゲーションを使用するサイトでは、フッタ自体が使われないこともある。

### スマートフォンの画面に合わせてページを表示する Viewport

この法則で登場した「Viewport」は、jQuery Mobileを使用しない場合でもスマートフォンサイトの制作でよく使用する設定になる。PCサイトをスマートフォンのブラウザで閲覧した際に、ページ全体が小さく表示されてしまう経験をした人もいると思うが、これは横幅の指定されていないページを閲覧した際に、ブラウザがページの横幅をデフォルトのViewport設定に合わせて980pxで表示してしまうために起きる現象だ。これはあらかじめmeta要素内のViewportの設定で横幅を指定して

おくことで回避できる。

jQuery Mobileでは<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">を指定しているが、width(幅)やinitial-scale(倍率の初期値)、maximum-scale(倍率の最大値)などを調整することで、ページを拡大縮小したり、任意のサイズのページを指定したりすることも可能になるので覚えておきたい。

# Ajaxを使ったページ遷移を理解する

## 複数のページを別々のHTMLファイルに記述してサイトを作成する

法則1で説明したように、jQuery Mobileではbody要素内にdata-role="page"を指定することで1ページとして認識される。そのため、例えば1つのHTMLファイル内にdata-role="page"を3つ指定すれば、全部で3ページのサイトということになる。このページの考え方はjQuery Mobileの大きな特徴の1つだ。しかし本書では通常のサイトのページの考え方と同様に、個別のHTMLファイルによる単体ページを複数作って、それらを相互に遷移させるサイトの作成方法を覚えてほしい。なぜなら、実際にスマートフォンサイトを制作する場合、階層が深くページ数の多いサイトを制作するケースが多く、主にこの方法を使うことになるからだ。このサイト設計については、初期段階でいろいろ迷うことが多いのだが、別々のHTMLファイルで制作すれば、コードの可読性も上がり、分業も可能になるので、迷った場合はこの構成でサイトを制作しておくことをおすすめする。

## ページ遷移にはAjaxを使って通信効率を向上させる

内部的な処理の話だが、jQuery Mobileではスムーズなページ遷移を実現して通信効率を向上させるために、通常「Ajax」という非同期処理を用いてHTMLファイルをロードし、コンテンツ部分だけDOM(Document Object Model)を書き換える方式をとっている。

Ajaxとは、Webページをインターネットからダウンロードして表示するまでの処理方式の1つだ。通常Webページは、ハイパーリンクがクリックされると、次に表示させたいHTMLファイルをサーバーからダウンロードしてきて、ブラウザ全体の表示をすべて破棄してから次のHTMLファイルの表示を始める。多くのサイトの場合、ヘッダ、フッタ、グローバルナビやサイドメニューなど一部共通のHTMLを表示している部分があると思うが、通常のページ遷移ではこれらもいったん破棄して再度同じものを描画している場合が多い。しかし、Ajaxを使ったページ遷移を行うと、ページ上にすでに表示されている要素で変更の必要のないものはそのまま利用し、表示を切り替えたいコンテンツ部分のみを入れ替えることが可能となる。

### ■ 関連する法則

#### 3

Ajaxによるページ遷移の不具合を解消する\_P22

#### 68

ページを移動するたびにランダムに画像を切り替える\_P147

### ■ 用語

Ajax\_P212

DOM\_P212

## フォームなどでは注意が必要なケースもある

jQuery Mobileではこのページ遷移の方法を採用することで、ページの切り替えごとにダウンロードされたり再描画されたりする部分の処理負荷や転送量が軽減され、快適なページ閲覧が可能になる。

しかし、Ajaxでの遷移を採用する場合にフォームを使用すると、ページ全体が切り替わらず、通常のページ遷移の際に受け渡されていたformの値の情報などの扱いが大きく変わってくるので、その部分だけAjaxを無効化させて運用することも多い。このAjaxによるページ遷移の不具合を解消する方法については次の法則で解説する。

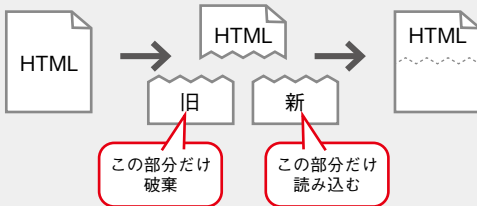
### ◆通常のページ遷移とAjaxのページ遷移の違い

#### ◎通常のページ遷移



通常のページ遷移では、ページ遷移をするたびに新たにHTMLファイルの情報を読み込む。

#### ◎Ajaxのページ遷移



Ajaxのページ遷移では、共通部分はそのまま利用し、表示を切り替えたい部分のみを読み込む。

### 複数のページを1つのHTMLファイルに記述する方法

複数のページ構造をもつサイトを1つのHTMLファイルで制作するには、単一ページのdiv要素であるdata-role="page"の塊を複数作り、それぞれにidを使用して各ページを指定すればいい。

ソースコードのように、id="index"、id="news"、id="about"と指定しておけば、アンカーリンクから#indexなどにリンクすることで、それぞれのページに遷移させることが可能になる。また、HTMLファイルの一番上に記述されたdata-role="page"の要素が最初のページとして表示される。2~3ページぐらいの簡単な構造のサイトを作成するケースなら、この方法で十分だろう。

この方式のメリットとしては、複数のHTMLファイルにまたがらないので、ヘッダやフッタを何度も記述せずに済むことや、一度のアクセスでHTMLファイルをすべてダウンロードできるので、ページ遷移の際にロード処理に時間がかからないことが

挙げられる。

デメリットとしては、1ページ内にすべての要素を記述するため、内容が多くなればなるほど可読性が損なわれ、また1回のアクセスでダウンロードするファイルサイズが大きくなってしまいうことだ。回線が遅いスマートフォンのサイトでは、あまりに大きなサイトを1つのHTMLファイルで制作してしまうと致命的になるおそれがあるので、ページ数が多いならこの法則で解説した複数のHTMLファイルに記述する方法を選んでほしい。

#### ◎複数のページを1つのHTMLファイルに記述するHTML5のコード

```
<div data-role="page" id="index">~</div>…1 ページ  
(トップページなど)  
<div data-role="page" id="news">~</div>…2 ページ  
(ニュースリリース、新着情報など)  
<div data-role="page" id="about">~</div>…3 ページ  
(会社概要など)
```

# Ajaxによるページ遷移の不具合を解消する

## Ajaxを使わずに複数ページを相互に遷移させる

法則2で解説した Ajax を使ったページ遷移の方法には注意点がある。Ajax を使うと、最初に読み込まれたページに head 要素が依存してしまうため、例えば「個別の CSS を使用する下層ページがある」などの理由で Ajax 読み込みを行いたくない場合も出てくるだろう。そのような場合、ある特別な処理を実行すれば、Ajax 処理を回避してページごとにロードしなおして表示させることが可能となる。

多くの場合デフォルトの Ajax の遷移を使うことになると思うのだが、サーバーの要件や、JavaScript イベントのハンドリングの方法によって、Ajax の遷移ではデータのやり取りに不具合が生じてしまう場合がある。このような場合、非 Ajax 処理を利用するのだが、それには以下で紹介する 4 通りの方法がある。状況に応じて使い分けよう。

### ■ 関連する法則

2

Ajax を使ったページ遷移を理解する\_P20

68

ページを移動するたびにランダムに画像を切り替える\_P147

### ■ 用語

Ajax\_P212

## ◆ Ajax を非使用にする処理

◎ ページ初期化時に Ajax を非使用にするコード

```
<script language="JavaScript">
  $(document).bind("pageinit", function(){
    $.mobile.ajaxEnabled = false;
  });
</script>
```

「\$.mobile.ajaxEnabled = false;」と記述すれば、jQuery Mobile がページを初期化した際に「すべてのリンクにおいて非同期処理を行わない」という指定が可能となる。一度にすべてのリンクに対して指定できるので、サイト全体で Ajax での遷移をまったく行わない場合はこの方法を使用する。

◎ a 要素で Ajax を非使用にするコード

```
<a href="/index2.html" rel="external">次のページ</a>
```

リンクごとに Ajax の使用／非使用を指定したい場合は、a 要素を使って非使用のリンクに rel="external" を記述する。サイト全体では Ajax を使用するが、特定のページのみ非同期処理を行いたくない場合などには、この方法を使用する。

◎ 特定のフォームだけ Ajax の遷移を回避するコード

```
<form data-ajax="false"></form>
```

ある特定のフォームだけ Ajax で遷移させたくない場合は、そのフォームに対して <form data-ajax="false"> と指定する。