

モバイルアプリをプロデュース

# HTML5で モバイルアプリ開発入門



**今後主流となるHTML5でモバイルアプリを作る入門書!**

かんたんなサンプルを用いて

**HTML5で何ができるのか** を解説

次世代Web技術の可能性を探る

- ※Apple、iPod、iTunes、iPhone、iMac、Safari、App Storeは、米国および他の国々で登録されたApple Inc.の登録商標または商標です。
- ※Android、Google Chrome、Google Play、YouTubeおよびYouTubeロゴは、Google Inc.の商標または登録商標です。
- ※Microsoft、Windows、Windows 7、Windows 8は、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- ※OperaはOpera Software ASAの商標または登録商標です。
- ※Mozilla、Firefoxとそれぞれのロゴは、米国 Mozilla Foundation の米国及びその他の国における商標または登録商標です
- ※Facebook、Facebookロゴは Facebook, Inc.の商標または登録商標です。
- ※Adobe、Adobeロゴ、Adobe Flash Playerは、米国またはその他の国またはその両方においてAdobe Systems Incorporated(アドビシステムズ社)の登録商標または商標です。
- ※その他、本書に記載されている会社名、製品名などは、各社の登録商標または商標です。
- ※本書ではTMおよび®の記載は省略しました。

# はじめに

---

次世代のHTMLと言われているHTML5ですが、いささか噂が先行しすぎて期待が高まっているのは確かだとしても、実際にどのようなものなのかをきちんと把握しようとする、範囲が広すぎてとらえ切れないという問題にぶつかります。

HTML5には、従来のHTML4ではプラグインなしで実現できなかったマルチメディア的な要素(タグや属性)が追加されています。これらによって、ブラウザ上で動く「Webアプリ」と呼ばれるものを作成することが可能になりました。HTML5を従来のHTML4との比較で考えると、単なるタグ付け言語のように思われるかもしれませんが、しかし、実は、HTML5は「仕様」と呼ばれるさまざまな機能の集まりなのです。

そして、HTML5とは、これまでWeb技術として発展してきた「HTML」「CSS」「JavaScript」といった技術を自在に操ることで、さまざまなWebサイトやWebアプリケーションと呼ばれるものを創り出すための「しくみ」でもあります。

本書では、HTML5の新しい機能をちょっとしたサンプルを用いて説明しています。おそらく機能を羅列するよりも直感的にわかりやすいはずです。そして、これらHTML5の技術を組み合わせて何ができるのかという展望についても説明していきます。

本書はまた、iPhoneやAndroidといったモバイル環境に特化した書籍でもあります。HTML5に期待されていることの1つは、モバイルにおけるアプリの開発環境としてのプラットフォームとなることです。HTML5を利用すると何ができるのか、何ができないのか。いまiPhoneやAndroidで動いているアプリとWebアプリはどのように違うのか。

本書では、これらの基本的な疑問に答え、解決策と方向性を提案することを目的の1つとしています。HTML5の全体像を把握し、アプリのさまざまな開発方法を知ること、今後のモバイル環境におけるアプリの可能性が見えてくるはずです。

## 第1章 HTML5によるWebアプリケーションの概要

<b>1-1 Webアプリケーションの基礎知識</b> .....	2
1-1-1 モバイルを取り巻く環境 .....	2
1-1-2 Webアプリケーションとは .....	3
1-1-3 HTML5の役割 .....	4
1-1-4 HTML5のAPI .....	5
1-1-5 HTML5以前のWebアプリケーションとは .....	6
1-1-6 HTML5によるWebアプリケーションとは .....	7
<b>1-2 HTML5でできること</b> .....	8
1-2-1 HTML5で新しくなったこと .....	8
1-2-2 Webアプリケーション 開発 .....	8
1-2-3 ソースの構造化 .....	10
1-2-4 フォーム機能の拡張 .....	11
<b>1-3 モバイルアプリケーション開発の手法</b> .....	12
1-3-1 モバイルアプリケーションの種類 .....	12
1-3-2 「Webアプリケーション」とは? .....	12
1-3-3 「ネイティブアプリケーション」とは? .....	15
1-3-4 「ハイブリッドアプリケーション」とは? .....	16
1-3-5 各アプリケーションごとの開発環境 .....	18
<b>1-4 押さえておきたいHTML5の技術</b> .....	20
1-4-1 HTML5仕様の概略 .....	20
1-4-2 HTML5仕様の機能カテゴリ .....	21
1-4-3 CSSについて .....	24
1-4-4 HTML5のAPIについて .....	27
1-4-5 アプリ開発に役立つフレームワークとツールについて .....	35

## 第2章 開発から公開までの流れ

<b>2-1 企画を考える</b> .....	40
2-1-1 アイデアをだす .....	40
2-1-2 ポジショニングマップ .....	41
2-1-3 クロスポイント抽出法 .....	45
2-1-4 アイデアを整理する .....	50
2-1-5 アプリの仕様を決める .....	51
2-1-6 将来の展開方法を決める .....	52
<b>2-2 開発環境を準備する</b> .....	54
2-2-1 開発に必要な環境 .....	54
2-2-2 safariのデバッグツール.....	55
2-2-3 Google Chromeの デバッグツール .....	57
2-2-4 Operaのデバッグツール .....	58
2-2-5 Firefoxのデバッグツール .....	59
<b>2-3 アプリケーションを作成する</b> .....	63
2-3-1 htmlを作成する .....	63
<b>2-4 アプリケーションをテストする</b> .....	70
2-4-1 動作を確認する .....	70
2-4-2 ブラウザごとに確認する.....	71
<b>2-5 アプリケーションを公開する</b> .....	72
2-5-1 公開する方法 .....	72
2-5-2 自分のサイトで公開.....	72
2-5-3 公式サイトで公開 .....	73

## 第3章 モバイル対応Webアプリを作成してみよう

<b>3-1 時計アプリ</b> .....	<b>76</b>
3-1-1 時計アプリ .....	76
3-1-2 使用するファイルの確認 .....	77
3-1-3 スマートフォンに対応させる .....	78
3-1-4 ソースファイルindex.html .....	80
3-1-5 ソースファイルpetitClock.css .....	82
3-1-6 ソースファイルpetitClock.js .....	85
3-1-7 完成イメージ(パソコンで見ると) .....	87
3-1-8 完成イメージ(iPhone / Android) .....	88
<b>3-2 おえかきアプリ</b> .....	<b>89</b>
3-2-1 おえかきアプリ .....	89
3-2-2 おえかきアプリのCanvas要素 .....	90
3-2-3 使用するファイルの確認 .....	92
3-2-4 スマートフォンに対応させる .....	93
3-2-5 ソースファイルindex.html .....	94
3-2-6 ソースファイルoekaki.js .....	96
3-2-7 完成イメージ(パソコンで見ると) .....	101
3-2-8 完成イメージ(スマートフォンで見ると) .....	102
<b>3-3 メモアプリ</b> .....	<b>103</b>
3-3-1 メモアプリ .....	103
3-3-2 使用するファイルの確認 .....	108
3-3-3 ソースファイルindex.html .....	109
3-3-4 ソースファイルpetitMemo.css .....	110
3-3-5 ソースファイルpetitMemo.js .....	110
3-3-6 完成イメージ(パソコンで見ると) .....	114
3-3-7 完成イメージ(iPhone/Androidで見ると) .....	115

## 3-4 地図アプリ ..... 117

3-4-1 地図アプリ ..... 117

3-4-2 使用するファイルの確認 ..... 122

3-4-3 ソースファイルindex.html ..... 123

3-4-4 ソースファイルpetitMap.css ..... 124

3-4-5 ソースファイルpetitMap.js ..... 124

3-4-6 完成イメージ(パソコンで見ると...) ..... 128

3-4-7 完成イメージ(iPhone/Androidで見ると...) ..... 129

## 3-5 ビデオ ..... 131

3-5-1 ビデオアプリ ..... 131

3-5-2 videoタグの注意点 ..... 135

3-5-3 ブラウザごとの対応状況 ..... 136

3-5-4 videoタグのソース ..... 137

3-5-5 複数のブラウザに対応したソースの記述方法 ..... 139

3-5-6 使用するファイルの確認 ..... 141

3-5-7 ソースファイルindex.html ..... 142

3-5-8 ソースファイルmovie.css ..... 143

3-5-9 完成イメージ(パソコンで見ると) ..... 144

3-5-10 完成イメージ(スマートフォンで見ると) ..... 144

## 3-6 オフラインアプリ ..... 145

3-6-1 オフラインアプリ ..... 145

3-6-2 使用するファイルの確認 ..... 152

3-6-3 ソースファイルindex.html ..... 153

3-6-4 ソースファイルoffline.css ..... 154

3-6-5 ソースファイルoffline.manifest ..... 154

3-6-6 ソースファイル.htaccess ..... 155

3-6-7 完成イメージ(パソコンで見ると) ..... 155

3-6-8 完成イメージ(iPhone/Androidで見ると) ..... 156

3-6-9 進化するオフラインアプリ ..... 159

<b>3-7 電力使用量表示アプリ</b> .....	161
3-7-1 電力使用量表示アプリ .....	161
3-7-2 progress要素と meter要素の使い方 .....	163
3-7-3 ソースファイルindex.html .....	166
3-7-4 ソースファイルstyle.css .....	167
3-7-5 完成イメージ(パソコンで見ると) .....	168
3-7-6 完成イメージ(スマートフォンで見ると) .....	169
3-7-7 対応していないブラウザの表示方法 .....	170
<b>3-8 Webアプリをネイティブアプリに近づける</b> .....	171
3-8-1 ネイティブアプリに 近づける .....	171
3-8-2 アイコンを作成する.....	175
3-8-3 ホーム画面に登録する (iPhone) .....	180
3-8-4 ホーム画面に登録する (Android) .....	183
3-8-5 Webアプリをネイティブに近づける (PC) .....	186

## 第4章 ハイブリッドアプリを作ってみよう

<b>4-1 ハイブリッドアプリを作るための流れ</b> .....	190
4-1-1 ハイブリッドアプリとネイティブアプリの違い .....	190
4-1-2 ネイティブアプリの開発環境 .....	191
4-1-3 ハイブリッドアプリの開発環境 .....	193
4-1-4 ハイブリッドアプリの苦手なこと .....	194
4-1-5 高度なHTML5アプリにする .....	195
<b>4-2 ハイブリッドアプリ作成ツールの 導入方法</b> .....	198
4-2-1 ハイブリッドアプリ作成ツール .....	198
4-2-2 PhoneGapの導入 .....	199
<b>4-3 アプリをネイティブ化する</b> .....	208
4-3-1 PhoneGapでネイティブ化 .....	208

4-3-2 ファイルを準備する	209
4-3-3 Xcodeでネイティブ化する	210
4-3-4 iOSシミュレーターで確認する	218

## 第5章 作ったアプリを公開しよう

<b>5-1 App Storeに公開する</b>	<b>220</b>
5-1-1 App Storeとは	220
5-1-2 アプリ公開の前に知っておくべきこと	221
5-1-3 iOSデベロッパプログラムの登録サポート	222
5-1-4 iPhoneでテスト	223
5-1-5 専用のサポートページを用意する	224
5-1-6 アプリの登録	225
5-1-7 アプリの審査	225
5-1-8 アプリ公開にあたって	226
5-1-9 HTML5がアプリの市場にもたらすもの	227
<b>5-2 Google Play Storeに公開する</b>	<b>228</b>
5-2-1 Google Play Store とは	228
5-2-2 Google Play Storeの特徴	229
5-2-3 アプリ公開までの流れ	230
5-2-4 公開の準備	231
5-2-5 アプリの署名	232
5-2-6 アプリのバージョンング	232
5-2-7 アプリ公開の 具体的な手順	234
5-2-8 Google Play Storeからダウンロードしてみよう	235
<b>5-3 Windows Storeに公開する</b>	<b>236</b>
5-3-1 Windows Storeとは	236
5-3-2 アプリの審査	237
5-3-3 アプリ公開に必要な費用	238

5-3-4 アプリ登録までの流れ .....	239
5-3-5 有償アプリの販売条件 .....	240
5-3-6 Windows Storeの可能性 .....	240
<b>5-4 自分のWebサイトでアプリを公開 .....</b>	<b>242</b>
5-4-1 野良アプリとは .....	242
5-4-2 自分のサイトで公開するメリット .....	242
5-4-3 自分のサイトで公開するデメリット .....	244
5-4-4 情報サイトからアプリを配布 .....	247
5-4-5 Facebookがアプリストアを開設 .....	248
<b>5-5 公開した後に .....</b>	<b>250</b>
5-5-1 アプリを公開 .....	250
5-5-2 アプリでお金を稼ぐ1 .....	251
5-5-3 アプリでお金を稼ぐ2 .....	252
5-5-4 アプリでお金を稼ぐ3 .....	253
5-5-5 アプリ専用サポートサイトの活用1 .....	254
5-5-6 アプリ専用サポートサイトの活用2 .....	255
索引 .....	256

## ■サポートページについて

本書の解説で使用している映像素材のサンプルファイルや、刊行後に発見された誤字および脱字に対する正誤表などの配布は、以下のサポートページで行います。

**本書専用サポートページ**

<http://www.scc-kk.co.jp/scc-books/support/B-359/support.html>

なお、サポートページの内容は必要に応じて随時更新されますので、ご注意ください。

# Memorandum

## 第1章

---

# HTML5による Webアプリケーションの概要

次世代標準の技術であるHTML5の本質を押さえておくことが、Webアプリケーション開発の鍵となります。

この章では、以下について、順次、説明していきます。

- ・Webアプリケーション
- ・HTML5の概念
- ・HTML5の仕様

# 1-1 Webアプリケーションの基礎知識

第1章

## 1-1-1 モバイルを取り巻く環境

2012年のはじめ、前年末の市場調査で、スマートフォンがパソコンを出荷数で大きく上回ったことが報じられました。右の報道が示すように、モバイル化の流れは、人々の想定を超えるスピードで進んでいます。まさに2011年は、本格的なモバイル時代が到来した年だったといえるでしょう。

The screenshot shows the Canalsys website newsroom page. The main headline is "Smart phones overtake client PCs in 2011" with a sub-headline: "- Vendors shipped 488 million smart phones in 2011, compared to 415 million client PCs". The article text states that Canalsys today released its full, detailed Q4 2011 country-level smart phone shipment estimates to clients, so completing the picture for the year. One notable result was that total annual global shipments of smart phones exceeded those of client PCs (including pads) for the first time. The article also includes a table of worldwide smart phone and client PC shipments and growth rates by category for Q4 2011 and full year 2011.

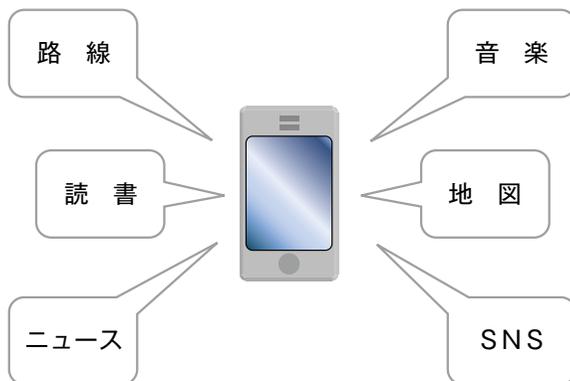
Category	Q4 2011		Full year 2011	
	Shipments (millions)	Growth Q4'11/Q4'10	Shipments (millions)	Growth 2011/2010
Smart phones	158.5	56.6%	487.7	62.7%
Total client PCs	120.2	16.3%	414.6	14.8%
- Pads	26.3	186.2%	63.2	274.2%
- Notebooks	6.7	-32.4%	28.4	-23.3%
- Netbooks	57.9	7.3%	209.4	7.5%
- Desktops	29.1	-3.6%	112.4	2.5%

Source: Canalsys estimates © Canalsys 2012

※イギリスの調査会社「canalsys」のサイト  
<http://www.canalsys.com/newsroom/smart-phones-overtake-client-pcs-2011>

HTML5によるWebアプリケーションの概要

現代のネットユーザーは、ネットワーク端末を、ただの情報流通のための道具と見なしているのではありません。各々の人生の時間を少しでも有効に活用するために欠かせない常備ツールとして、ありとあらゆる生活シーンで、積極的にモバイル端末を役立てています。



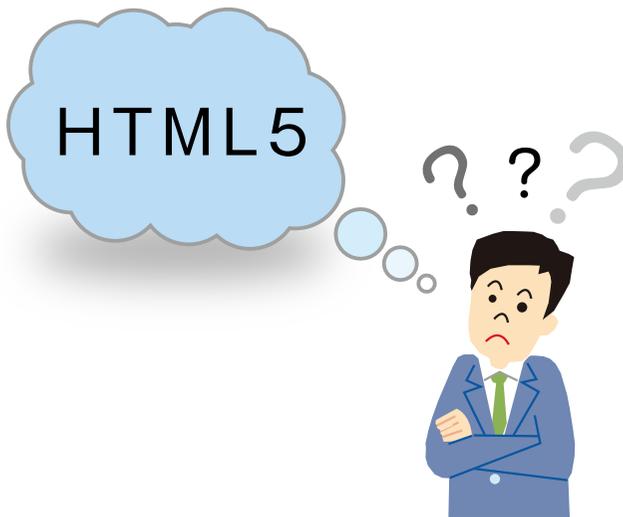
その中で、重要な意味を持つのが「モバイルアプリケーション」、つまり一般的に「アプリ」と呼ばれているソフトウェアです。本書では、モバイルアプリケーションの中でも「Webアプリケーション」を中心に解説しています。

## 1-1-2 Webアプリケーションとは

本書で学ぶ「Webアプリケーション開発」の中心である「HTML5」と呼ばれるマークアップ言語は、時代の要請に応じていく上で、もっとも核心的な要素です。

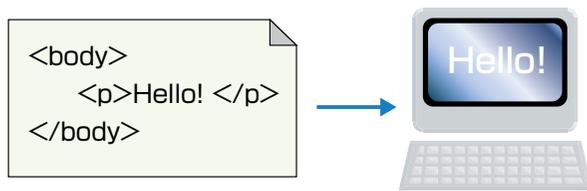
ひとくちに「Webアプリケーション」といっても、実装するための技術はさまざまです。また、モバイル端末やPCなどによっても開発環境が異なります。

まずは、「HTML5」を利用したWebアプリケーションとは何か、何ができ、何ができないのか、どうすれば開発できるのかといったことを理解する必要があります。

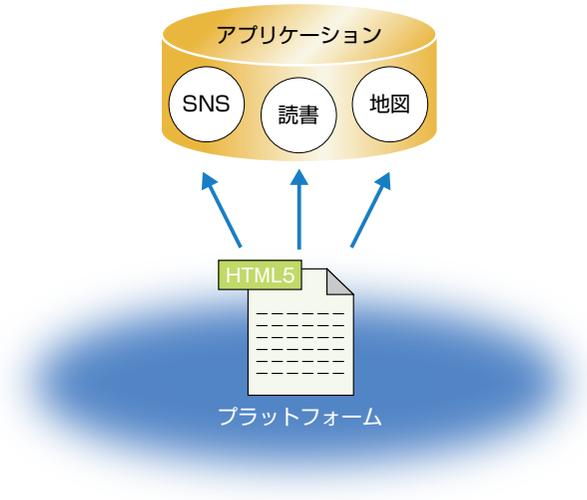


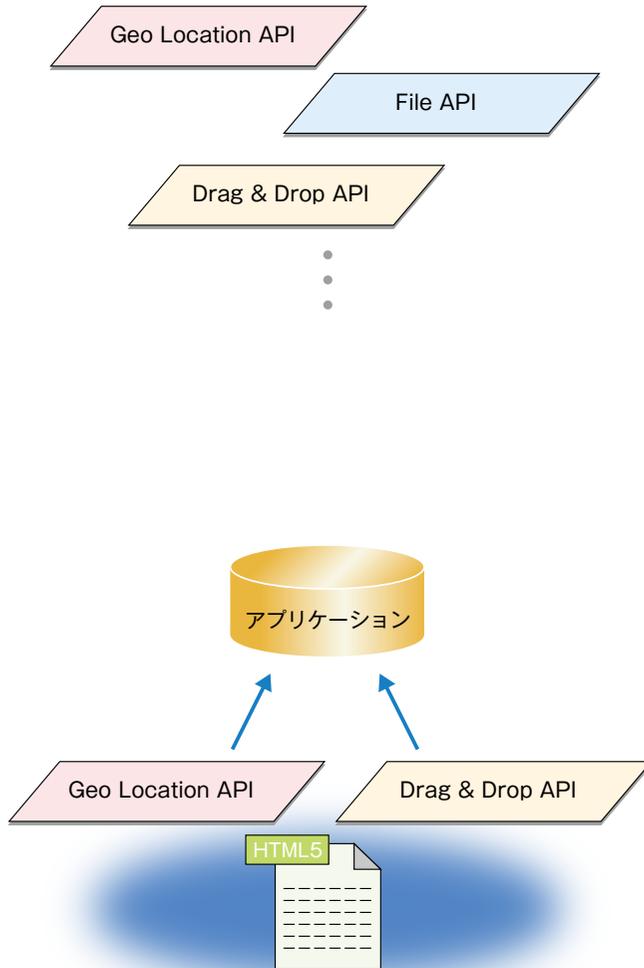
### 1-1-3 HTML5の役割

HTML5には、主に二つの役割があります。まず第一に、旧来のHTMLから継承した文書のマークアップが挙げられます。



そして第二には、アプリケーションのプラットフォームとしての役割です。この部分こそが、旧来のHTMLとの決定的な違いであり、本書の中心的なテーマということになります。





#### 1-1-4 HTML5のAPI

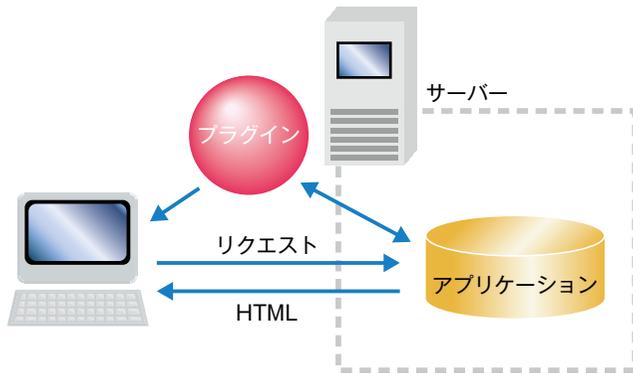
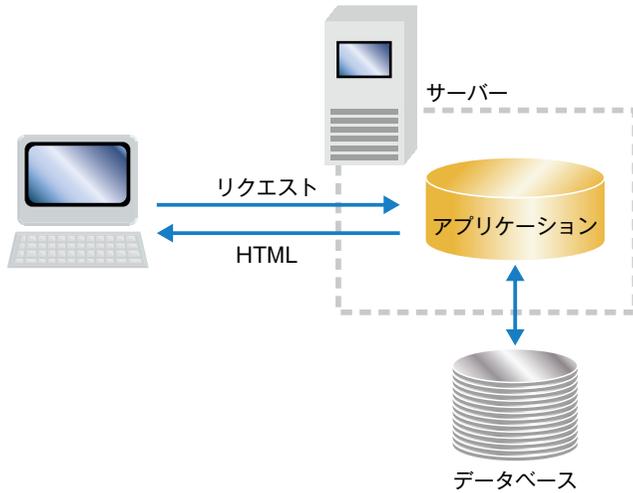
HTML5には、アプリケーションのプラットフォームとしての役割を果たすための多くの機能が追加されています。これらの機能は、ブラウザを通じてさまざまな操作を行うためのAPI (Application Program Interface)として提供されています。

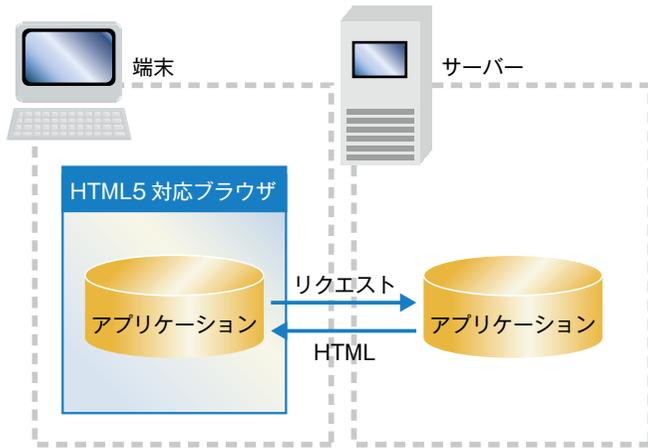
これらをどのように活用すれば、Webアプリケーションとして効率的に機能させられるのか、というアプローチが、これからHTML5に取り組んでいく上での本書の基本姿勢となります。

## 1-1-5 HTML5以前の Webアプリケーションとは

HTML5以前のWebアプリケーションは、多くの場合、インターネット上に分散しているWebサーバーに関連づけられたプラットフォームから提供されていました。旧来のHTMLは、サーバー上のアプリケーションが実行した結果の出力として、ユーザーのブラウザに表示されます。

また、いわゆるリッチコンテンツを提供するWebサービスの場合、利用者が自身の端末に、Flash等のプラグインをインストールすることが動作の前提となっているケースが目立ちました。





## 1-1-6 HTML5による Webアプリケーションとは

HTML5が実現するWebアプリケーションの場合、動的なコンテンツの動作の大部分が、Flash等の特別なプラグインを介さず、サーバー上のアプリケーションとのやり取りをせずに、利用者側のブラウザ上で展開します。

これによって、旧来のWebアプリケーションで生じていたWebサーバーに対する過負荷や通信速度の限界、端末側の実装状況による動作の違いといった課題を、端末側のWebブラウザに分担させることができるようになります。

### One Point

HTML5の考え方と普及の背景には、米Google社が2005年に発表したGoogleMapsというWebアプリケーションの大成功があります。

GoogleMapsでは、特別なプラグインをインストールすることなく、また、画面のスクロールの度ごとに、表示に必要な全データを再読み込みさせることなく動作します。GoogleMapsの登場が、その後のWebアプリケーションに方向性を与えたといえます。

# 1-2 HTML5でできること

第1章

## 1-2-1 HTML5で 新しくなったこと

HTML5で「何ができるか」について、具体的に見ていきましょう。旧来のHTMLと比較して大きく異なるのは3つです。

以下では、それぞれの特徴を、かいつまんで説明します。

- ・Webアプリケーション開発に適している
- ・ソースの構造化が強化されている
- ・入力フォームの機能が拡張されている

第2章

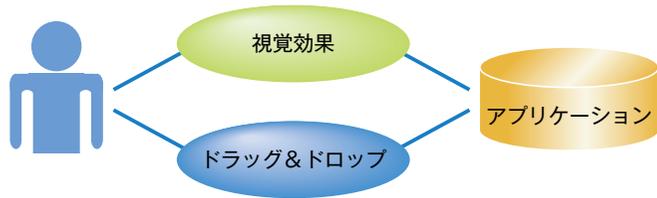
第3章

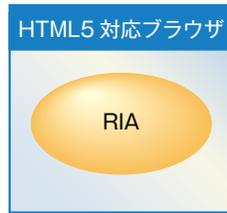
第4章

第5章

## 1-2-2 Webアプリケーション 開発

まず、Webアプリケーション開発について見てみましょう。ユーザーは、UI(ユーザーインターフェイス)を利用して、アプリケーションとやり取りをします。





従って、アプリケーションの開発者は、グラフィカルインターフェースによる描画機能を利用して視覚的な効果を出したり、ドラッグ&ドロップなどの支援機能を用いて操作性を上げることによって、より高機能かつ高性能なアプリケーションを作り上げていく必要があります。

従来は、さまざまなプラグイン (Java、Flash) や、Ajaxなど Web 技術を組み合わせることで、Web アプリケーションを実現してきました。HTML5以降は、ブラウザだけでRIA (リッチインターネットアプリケーション) を実現できるようになっています。

HTML5でWebアプリケーションを開発することは、種々のプラグインを必要としなくなるという点で、利用者の利便に大きく貢献します。

他方、利用者の環境に依存することなくコンテンツの開発に専念できるようになることは、開発者にとっても大きなメリットといえるでしょう。

たとえば、動画共有サービスのYouTubeでは、既存のFlashプレーヤーに加えて、HTML5を利用した動画プレーヤーを提供しています。



※ YouTube HTML5 動画プレーヤーの紹介サイト

<http://www.youtube.com/html5?hl=ja>

### 1-2-3 ソースの構造化

次に、ソースの構造化について見て行きましょう。HTML5には、文章を構造化するための新しい要素が追加されています。これによって、Webページを構成する要素に意味を持たせることができます。たとえば、検索エンジンなどは、構造化タグを参照することで、Webページの内容をより明確に判断することが可能となります。

#### HTML5の構造化タグ

タグ	概要
header	ヘッダー領域を明示する
footer	フッター領域を示す「footer」
nav	ナビゲーション領域を示す
section	ドキュメント類のセクションを明示する
article	記事であることを示す
aside	補足的な情報であることを示す

## HTML5で追加されたtype属性

search	検索
url	URL
email	メールアドレス
number	数値
tel	電話番号
date	日付
month	月
week	週
time	時間
color	色

### 1-2-4 フォーム機能の拡張

最後に、フォーム機能について見ていきます。HTML5では、入力フォームの機能が拡張され、多くの属性が追加されています。

たとえば、「input」要素に指定可能なtype属性が追加されています。これらはフォーム部品の種類を指定する際に使用します。HTML5では、URLや電子メール、検索情報などの入力欄のほかに、日付や時間の入力欄、レンジや色の入力欄などを作成することも可能となっています。

また、HTML5で追加された便利な属性として、テキストの入力フォームにあらかじめ補助情報を表示しておける「placeholder」、自動で入力箇所の手元をフォーカスする「autofocus」や、入力が必要の際に適用する「required」等が挙げられます。

# 1-3 モバイルアプリケーション開発の手法

第1章

第2章

第3章

第4章

第5章

## 1-3-1 モバイルアプリケーションの種類

モバイルアプリケーションの種類、そしてそれらのモバイルアプリケーションを開発するための方法として、以下の3つが挙げられます。

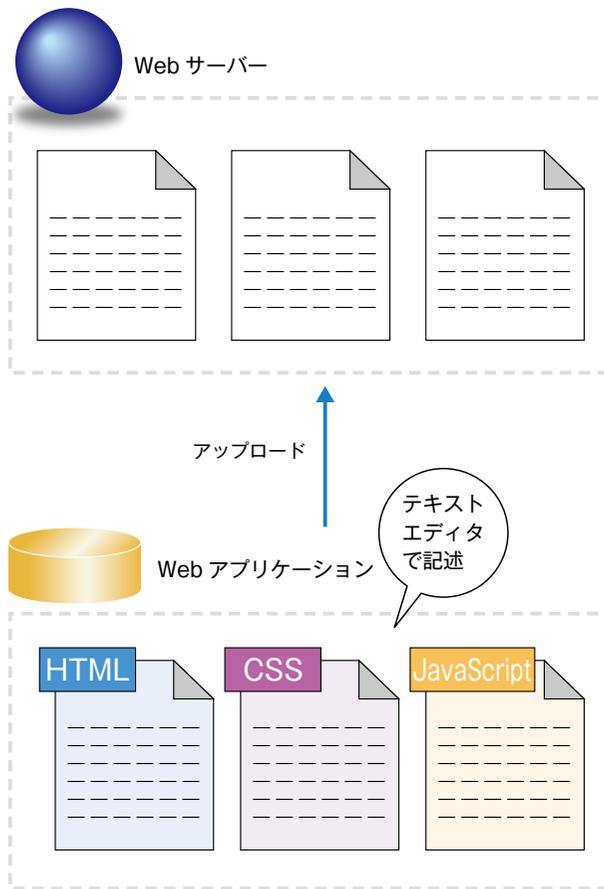
- ・Webアプリケーション
- ・ネイティブアプリケーション
- ・ハイブリッドアプリケーション

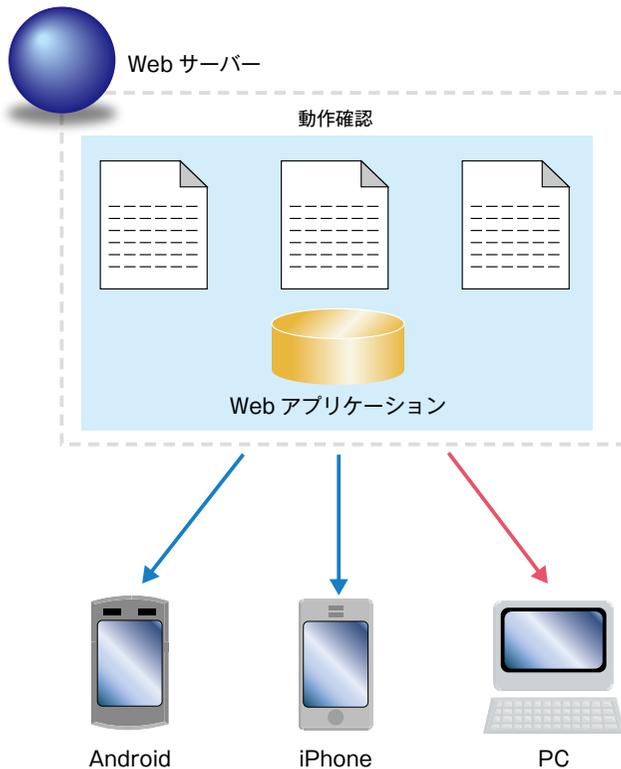
以下では、それぞれの開発手法について説明します。

## 1-3-2 「Webアプリケーション」 とは？

それでは、Webアプリケーションを開発するための方法に、一歩踏み込んでみましょう。ここではまず最初に、簡単に本書内での「アプリケーション」を定義しておきます。本書では、HTML、CSS、JavaScriptといったWebに関する仕様に基づいて作成され、Webブラウザで開いた際に動作するアプリケーションのことを、「Webアプリケーション」と呼びます。ただし、本書はHTML5によるアプリケーション開発をテーマとしているため、HTML5の仕様を用いることを前提とします。

HTML、CSS、JavaScriptは、「メモ帳」や「テキストエディット」等のテキストエディタさえあれば誰でも記述・作成できます。ちょっとしたアプリケーションの開発なら、特別な開発ツールを導入することなく、すぐに開発を始めることが可能です。





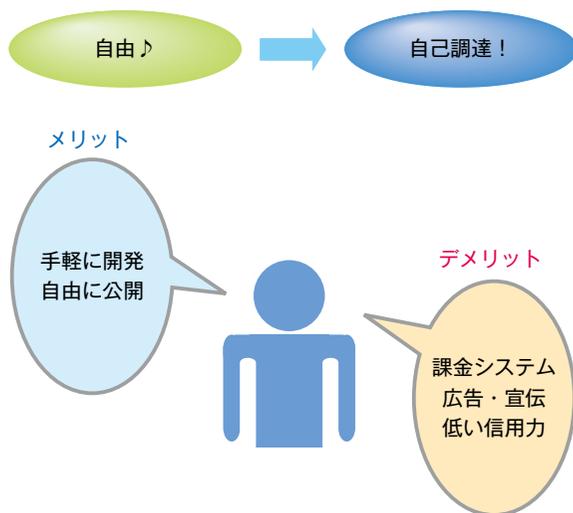
作成してWebサーバにアップロードしたWebアプリケーションは、パソコンからでも、iPhoneやAndroidのようなスマートフォンからでも、HTML5対応のWebブラウザさえ実装していれば、簡単に動作を確認できます。

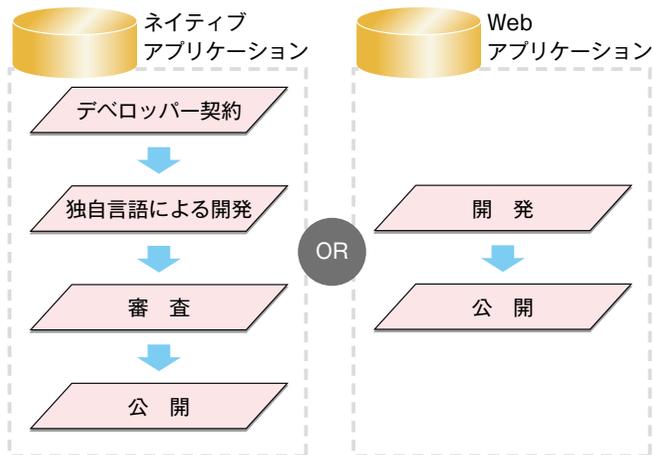
加えて、Webアプリケーションは、スマートフォンをはじめとしたモバイル端末上で動作するアプリケーションの可能性を大きく広がります。たとえばiPhoneで動作するiOSアプリケーションの場合、iOS SDKやXcodeという社独自の開発環境を用いて開発し、同社による審査の後にApp Storeで公開する、といった一連の手順を踏む必要があります。

「Webアプリケーション」の場合、特別な開発環境を整える必要も、特定のサイトで公開するための許可を得る必要もありません。

手軽にWebアプリケーションを作成して、公開できるというのは、初期投資を最小限に抑える上で極めて大きなメリットです。しかし、反面、数多くのユーザーがアプリを検索するApp Storeでの公開はできません。

アプリケーションをどのように宣伝するのか、またどのように課金するのか、といった問題を解決する必要があります。





### 1-3-3 「ネイティブ アプリケーション」とは？

Webアプリケーションが端末に左右されずにブラウザ上で動作するのに対し、特定のハードウェア上でのみ動作するアプリケーションのことを「ネイティブアプリケーション」と呼びます。たとえばiPhoneやAndroid端末で、ブラウザを介することなく直接実行するアプリケーションがネイティブアプリケーションにあたります。

ネイティブアプリケーションを開発するには、独自の開発環境を導入し、独自の開発言語を用いる必要があります。iPhoneの場合ならObjective-C、Androidの場合ならJavaという言語を用いて開発し、コンパイラでアプリケーションを作成します。

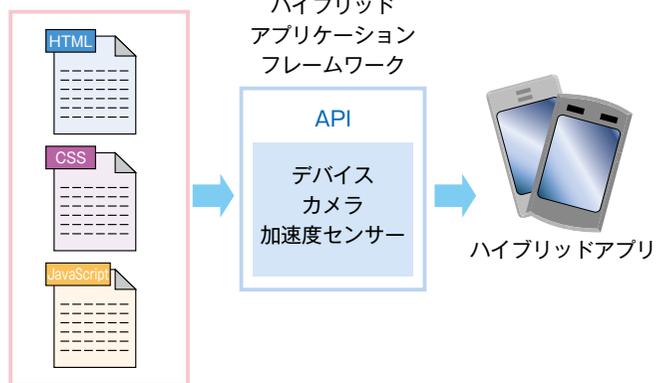
これらの開発環境を整え、独自の言語を習得してアプリケーションを開発するのは、プログラミングの初級者にとって、容易なことではありません。ただし、正式な「ネイティブアプリケーション」として公式のマーケットで公開したり、課金したりしたい場合には、この方法で開発する必要があります。

### 1-3-4 「ハイブリッド アプリケーション」とは？

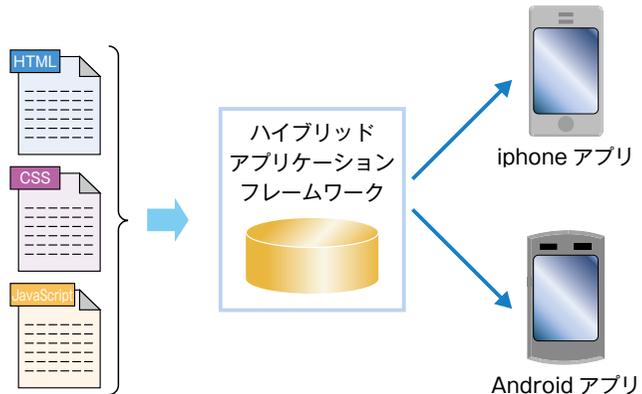
加えて考慮する必要があるのが、HTML5、CSS、JavaScript等のWeb技術を用いて、ネイティブアプリケーションを開発する「ハイブリッドアプリケーション」の存在です。ハイブリッドアプリケーションは、「ハイブリッド」が意味するとおり、Webアプリケーションとネイティブアプリケーションとの混合型のアプリケーションです。

ハイブリッドアプリケーションを開発するには、サードパーティのフレームワークを導入します。これによって、通常のWebアプリケーションだけでは利用できないカメラのコントロール、デバイス通知機能、加速度センサーなど、デバイス独自の機能を、JavaScriptから扱うことができるようになります。この方法には、他に「ネイティブアプリケーション」開発における最大の障害ともいえるObjective-CやJavaなどの言語を使用しないで済むという大きな利点があります。

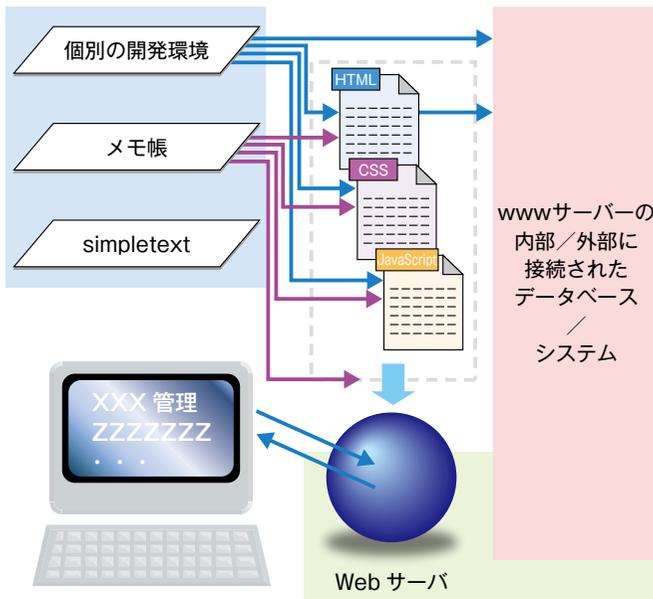
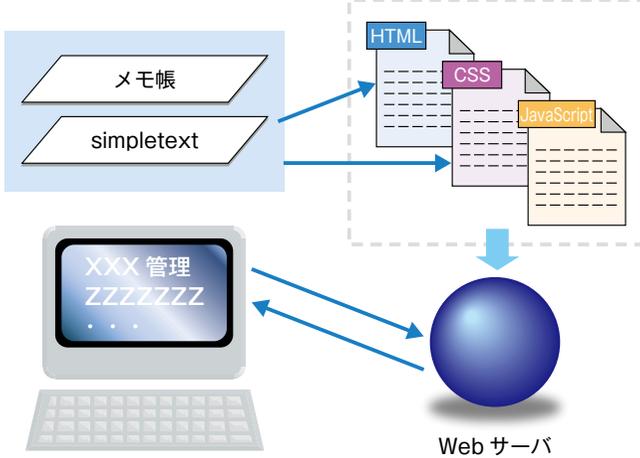
ソースコード



ハイブリッドアプリケーションの  
フレームワーククロスプラットフォーム



## もっともシンプルな Web アプリケーション開発の流れ



Webアプリケーションはブラウザ上で動作するため、Android MarketやApp Storeで公開することはできません。しかしハイブリッドアプリケーションの場合、AndroidやiPhoneで直接動作するネイティブアプリケーションとして作成することができるため、Android MarketやApp Storeで公開したり、課金したりすることが可能です。

また、ハイブリッドアプリケーションのフレームワークの多くがクロスプラットフォーム開発に対応しており、1つのソースコードで同時にAndroidとiPhone向けのアプリケーションを開発できるという利点もあります。

## 1-3-5 各アプリケーションごとの 開発環境

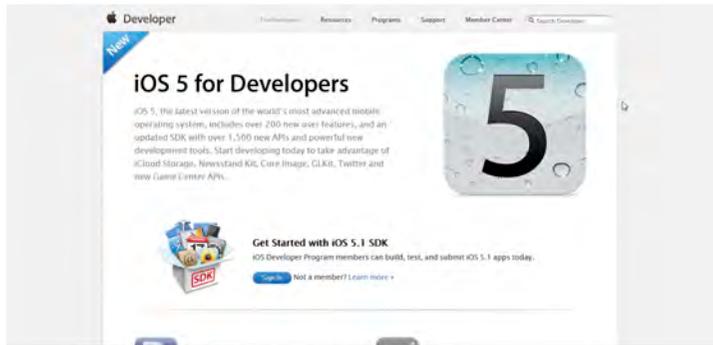
### ■Webアプリケーション の開発環境

Webアプリケーションの開発には、Webブラウザと、テキストエディタ等によってHTML、CSS、JavaScriptファイルを記述・保存できる環境さえあれば、いつでも、また誰でも開発に着手できます。

ページやパーツを作成したり操作性をアップするためのフレームワークやテンプレートを利用することで、よりネイティブアプリケーションに近いWebアプリケーションを作成することも可能です。代表的なフレームワークには、Sencha Touch、Wink toolkitなどがあります。

### ■ネイティブアプリケー ションの開発環境

ネイティブアプリケーションの開発環境について説明します。AndroidやiPhoneといったアプリケーションを動作させたい端末に固有の開発環境を利用します。iPhone向けにはiOS SDK、Android端末向けにはAndroid SDKが提供されています。



◆iOS SDKの技術サイト

<https://developer.apple.com/jp/technologies/ios5/>



◆Android SDKのダウンロードサイト

<http://developer.android.com/sdk/index.html>

第1章

第2章

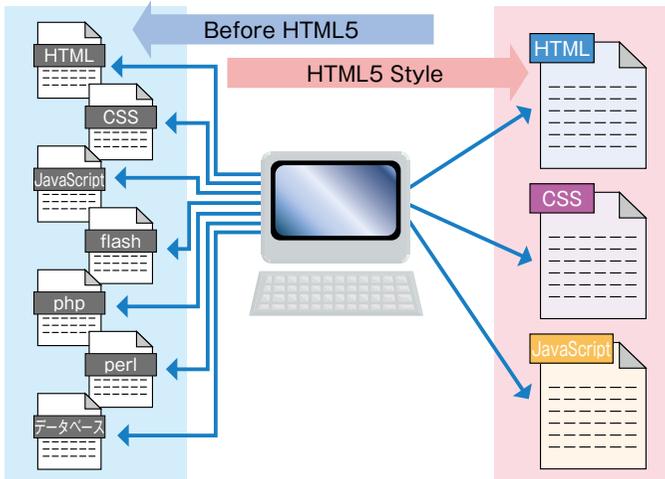
第3章

第4章

第5章



◆PhoneGap公式サイト  
<http://phonegap.com/>



## ■ハイブリッドアプリケーションの開発環境

最後にハイブリッドアプリケーションの開発環境ですが、2012年現在、さまざまなフレームワークが存在しており、インターネット関連の開発環境で、もっともホットな分野だといえます。

ハイブリッドアプリケーションの開発では、HTML、CSS、JavaScriptといったWeb技術を使いますが、それぞれのフレームワークごとに特徴があり、開発の流れやAPIを利用する方法などが異なります。

代表的な開発環境に、PhoneGapとTitanium Mobileがあります。両方とも端末固有の機能を扱うことができ、クロスプラットフォーム開発にも対応しています。

第1章

第2章

第3章

第4章

第5章

# 1-4 押さえておきたいHTML5の技術

第1章

第2章

第3章

第4章

第5章

HTML5によるWebアプリケーションの概要

## 1-4-1 HTML5仕様の概略

HTML5の仕様は、次世代のHTML標準として、現在W3C (World Wide Web Consortium) によって、2014年の勧告を目指して策定が進められています。ただし、マークアップの仕様であった旧来のHTMLとは異なり、HTML5の仕様の策定作業はW3C、Khronos、WHATWG等の団体によって機能ごとに進められています。機能によって仕様策定の状況はさまざまですが、すでにブラウザへの実装が始まっています。

以下で説明するHTML5の仕様概略は、HTML5でアプリケーション開発を行なう上での前提となる知識です。ごく基本的な内容を中心に、機能ごとにまとめて紹介していきます。より詳細な情報は、策定作業中の公式サイトで参照できます。

### 広義のHTML5の範囲とは…

各ベンダーが個別に開発を進めている未知のAPI

認知はされているもののW3Cでの策定作業が未着手のAPI

W3Cで策定作業が概ね完了したAPI

W3Cで策定作業が概ね完了したが、使い物にならないAPI

W3Cで策定作業が完了したが、再検討・排除されたAPI

### HTML5の仕様と参照先

#### W3C → HTML5

##### WHATWG

セマンティクス (HTML5 Vocabulary)

オンラインストレージ (Web Storage)

パフォーマンス (Web Workers)

##### Khronos

3Dグラフィクス (Web GL)

W3C Editor's Draft



## HTML5

A vocabulary and associated APIs for HTML and XHTML

Editor's Draft 8 May 2012

**Latest Published Version:**

<http://www.w3.org/TR/html5/>

**Latest Editor's Draft:**

<http://dev.w3.org/html5/spec/Overview.html>

**Previous Versions:**

<http://www.w3.org/TR/2012/WD-html5-20120329/>

<http://www.w3.org/TR/2011/WD-html5-20110525/>

<http://www.w3.org/TR/2011/WD-html5-20110405/>

<http://www.w3.org/TR/2011/WD-html5-20110113/>

<http://www.w3.org/TR/2010/WD-html5-20101019/>

<http://www.w3.org/TR/2010/WD-html5-20100824/>

<http://www.w3.org/TR/2009/WD-html5-20090825/>

<http://www.w3.org/TR/2009/WD-html5-20090423/>

<http://www.w3.org/TR/2009/WD-html5-20090212/>

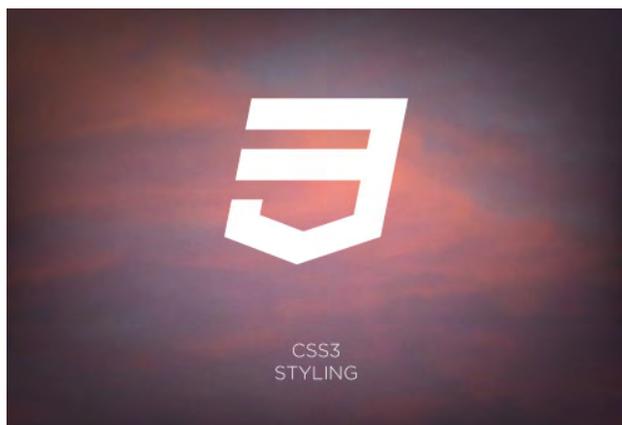
<http://www.w3.org/TR/2008/WD-html5-20080810/>

<http://www.w3.org/TR/2008/WD-html5-20080122/>

dev.w3.org からデータを転送しています...

◆W3Cの公式サイト(エディタースドラフト)

<http://dev.w3.org/html5/spec/single-page.html>



## 1-4-2 HTML5仕様の 機能カテゴリ

### ■HTML5 SEMANTICS

このカテゴリは、HTMLの「セマンティクス要素」についての仕様です。「1-2-3ソースの構造化」でソースの構造化について説明しましたが、HTML5には、文書内のテキストを意味づけするための要素が追加されています。

### ■CSS3 STYLING

Webページのレイアウト設定やUI（ユーザーインターフェース）のスタイリングを行うCSSファイルの拡張、および追加機能についての仕様です。

CSS3については、「1-4-3CSSについて」で後述します。

### ■DEVICE ACCESS

さまざまなデバイスを認識して、位置情報、カメラ、マイク、連絡先やスケジュールなどのデータを利用するための仕様です。

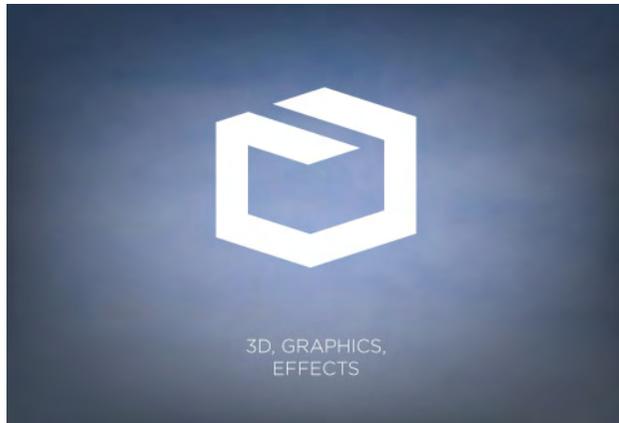
## MULTIMEDIA

オーディオとビデオをHTML5で扱うための仕様です。HTML5では、プラグインを使用しなくても、<video>および<audio>タグを利用して、オーディオとビデオを再生することが可能です。



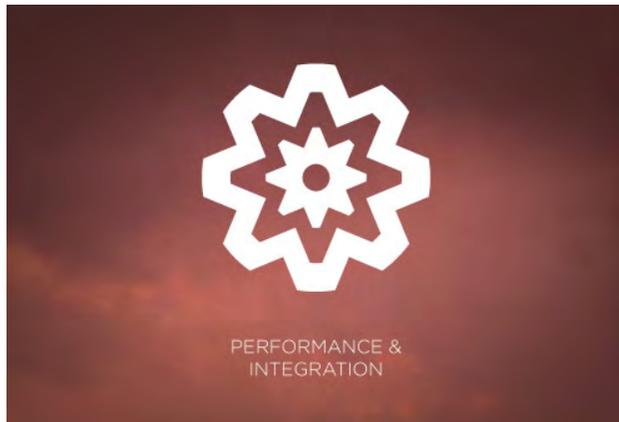
## 3D、GRAPHICS、EFFECTS

SVG (XMLで記述されるベクター画像)、Canvas (2D/3D描画機能)、WebGL (3Dアクセラレート)などのグラフィックやエフェクト機能についての仕様です。これによって、従来はAdobe Flashなどのプラグインを必要としていたコンテンツの描画をHTML5で置き換えることが可能となります。



## PERFORMANCE & INTEGRATION

Web Workers (バックグラウンドでJavaScriptを実行する機能)やXMLHttpRequest 2 (ページ遷移することなしにサーバーとデータ交換するための機能)といった、動的なコンテンツの作成に欠かせない機能に関する仕様です。





## ■OFFLINE & STORAGE

Webアプリケーションをローカルにキャッシュしておき、オフライン環境で実行したり、高速に起動したりするための仕様です。



## ■CONNECTIVITY

Web Sockets (HTTPを使用せずにWebサーバーと通信するための仕組み)、Server-Sent Events (HTTP通信を使用してサーバーからデータをプッシュするための仕組み)などの技術を使用して、効率よくデータ通信を行うための仕様です。

以上の8つのカテゴリに分類された機能によって、無秩序に拡散を続けてきたWeb関連の技術や仕様が1つに収束することが期待されています。

### One Point

HTML5仕様の8つのカテゴリには、それぞれサブアイコンが用意されています。Webサイトでどの機能を利用しているのかをわかるようにする目的で、これらのアイコンを使用できます。

## 1-4-3 CSSについて

CSS3では、さまざまな機能がモジュールとして提供されています。2D/3D変形、アニメーション、グラデーション、透過色、音声、テキストの縦書きやルビなど、リッチコンテンツを作成するためさまざまなモジュールが、別々の仕様として策定されています。

ここでは、CSS3で新しく追加された機能である段組みレイアウト、セレクターおよびエフェクト、ベンダープレフィックスについて説明します。

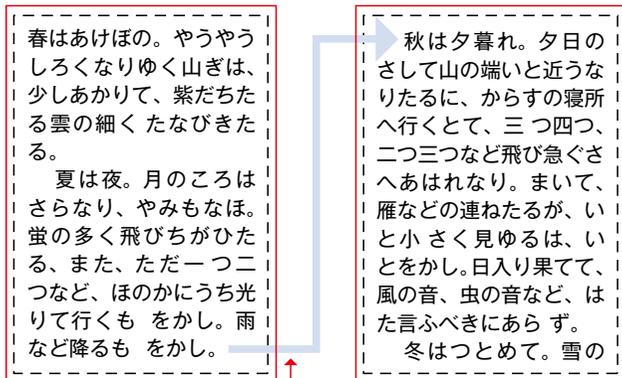
### ■段組みレイアウト

CSS3の画期的な新機能に、段組みレイアウトがあります。複数のカラムに記事が流し込まれる体裁は新聞・雑誌でおなじみですが、従来のHTMLで段組みレイアウトを行うには、tableタグやCSSのfloatプロパティなどを利用して調整する必要がありました。CSS3では、簡単に段組みレイアウトを行うことが可能となっています。

たとえば、CSS3で複数のカラムを作成して文章を流し込んだ場合、1カラム目と2カラム目で自動的に文章の高さをそろえて表示することができます。

CSS3で段組みレイアウトを行うには、column-countでカラム数を定義する方法、column-widthでそれぞれのカラムの幅を定義する方法があります。

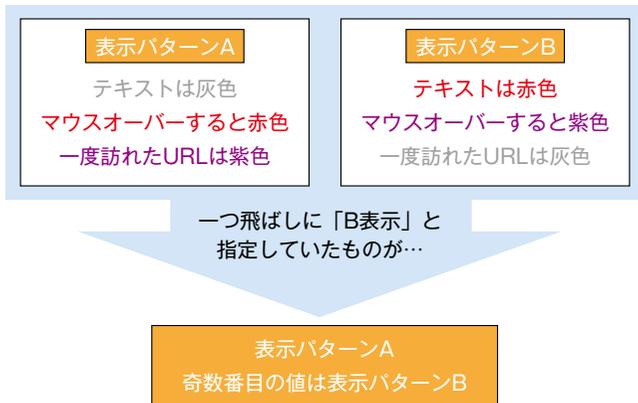
### column-countでカラムを2つに設定した場合の例



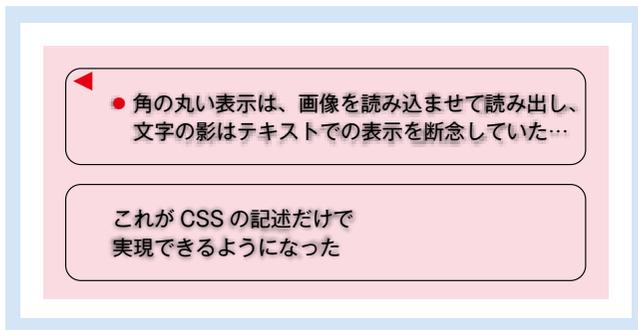
テキストの自動流し込みが可能

段の高さを揃えることが可能

## CSSセレクターの例：nth-of-type (2n+1)



## CSS3では多種多様なエフェクトが利用可能に…



## ■CSSセレクターとエフェクト

スタイルを適用する対象を選択するための規則のことをCSSセレクターといいます。CSS3では新しいセレクターが多く追加されています。CSS3セレクターをうまく利用することで、これまでHTMLとJavaScriptで記述していた部分を大幅に減らすことが可能です。

また、CSS3へのバージョンアップでは、多様なエフェクトが追加されています。CSS3では、CSSプロパティで、角丸の枠線や、文字のドロップシャドウ、透明度などを設定することが可能です。

## ■ベンダープレフィックス

最後にベンダープレフィックスについて説明します。

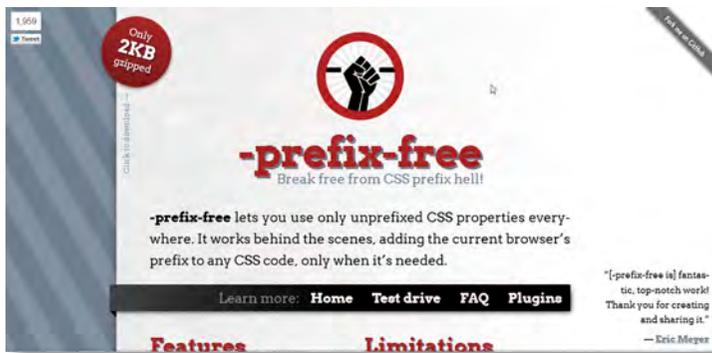
CSS3の様子はモジュール化されているため、それぞれの仕様の勧告状況が異なります。草案段階の拡張機能を先行実装しているブラウザへの対応として用意されているのが、「ベンダープレフィックス」と呼ばれるベンダー識別子です。

先行して実装されているCSS3の拡張機能を使うためには、「-moz-」(Firefox)や「-webkit-」(Google Chrome、Safari)といったベンダープレフィックスをプロパティ名や値の先頭に付ける必要があります。

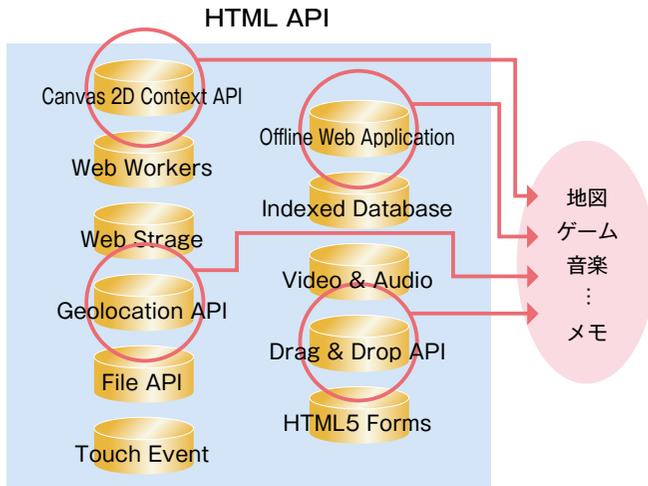
プレフィックスを手動で対応するのが面倒な人のために、自動で処理してくれるツールや、回避のためのパッチを無償で提供しているサイトがあります。



◆CSSPrefixer (CSSプレフィクスジェネレーター)  
<http://cssprefixer.appspot.com/>



◆-prefix-free (CSSプレフィクスパッチング)  
<http://leaverou.github.com/prefixfree/>



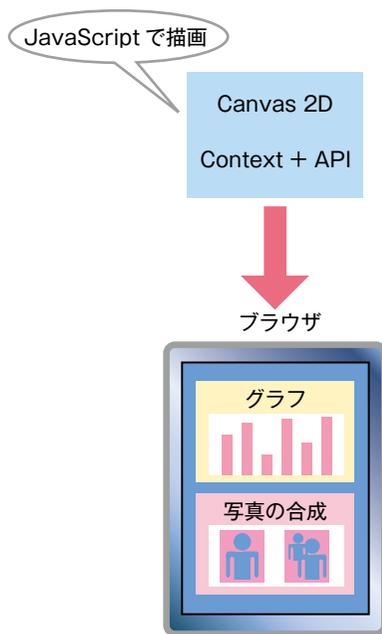
#### 1-4-4 HTML5のAPIについて

HTML5によるWebアプリケーションは、複数のAPIを組み合わせることで実現します。したがって、HTML5で「何ができるか」を考える上で、主要なAPIにはどのようなものがあるのかを理解しておくとは必ず役に立つはずで

## ■Canvas 2D Context API

JavaScriptを用いてブラウザ上に図形を描画するためのAPIです。座標でパスを定義することで、線や円などの図形を描画します。このほか、画像ファイルを読み込んだり、テキストを描画することもできます。また、簡単なアニメーションを作成することも可能です。

なお、Canvasは、2Dのみならず、WebGLを通じて3Dの描画もサポートしていますが、ブラウザによって実装状況が異なります(2012年5月現在)。



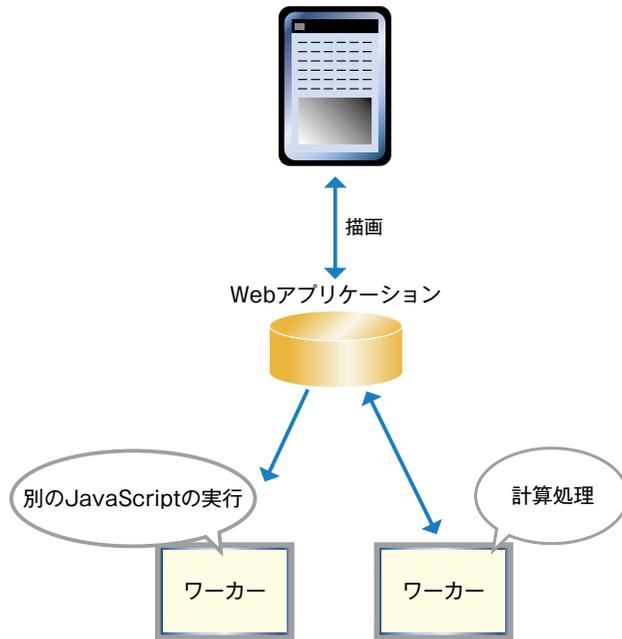
第1章

第2章

第3章

第4章

第5章



## ■ Web Workers

Web Workersは、JavaScriptによる複数の処理を並行して実行するためのAPIです。たとえば、ブラウザ上には見えないところでJavaScriptを実行しつつ、描画処理を行うといった処理が可能となります。

この機能は、インタラクティブ性の高いWebアプリケーションや計算処理の負荷が大きいWebアプリケーションで重要な役割を果たします。

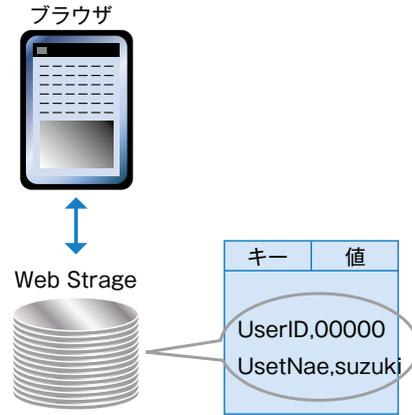
## ■Web Storage

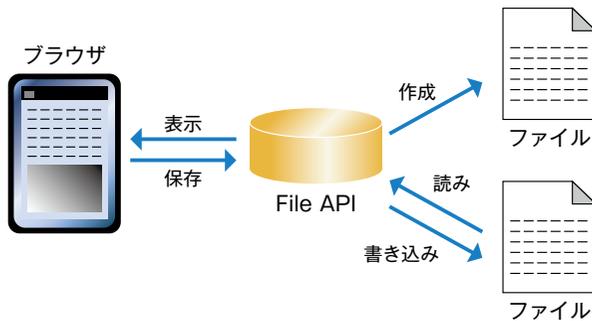
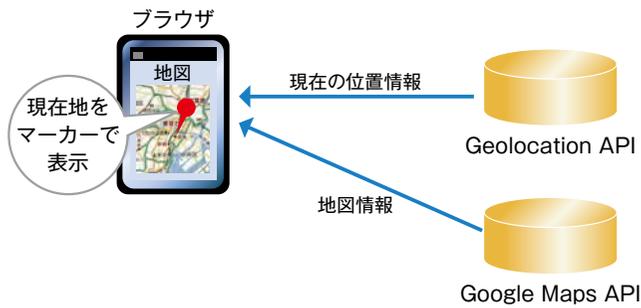
データをブラウザに保存しておくためのAPIです。

従来のクッキーによるデータ保存は4KBytesまででしたが、Web Storageでは5MBytesまで保存できますので、小規模なWebアプリケーションとしては十分なデータ量を扱うことができます。また、クッキーとは異なり、毎回すべてのデータが送信されるわけではないため、通信の負荷を軽減することが可能です。

Web Storageでは、キー(key)と値(value)のペアによるシンプルなデータとして保存されます。

なお、Web Storageは複数の処理を1つにまとめて実行するトランザクション処理には対応していません。データベースとしての処理を行うには、Indexed Database API(後述)などを利用する必要があります。





## ■Geolocation API

Geolocation APIは位置情報を扱うためのAPIです。位置情報は、緯度と経度の座標として取得されます。Geolocation APIはシンプルな仕組みですので、たとえば、現在位置を取得して、Google Mapにマーカーを付けるというWebアプリケーションを簡単に作成することができます。

Geolocation APIでは、無線LAN、携帯電話基地局、GPSなどから位置情報を取得するため、接続環境によって取得される位置情報の精度が異なる場合があります。

## ■File API

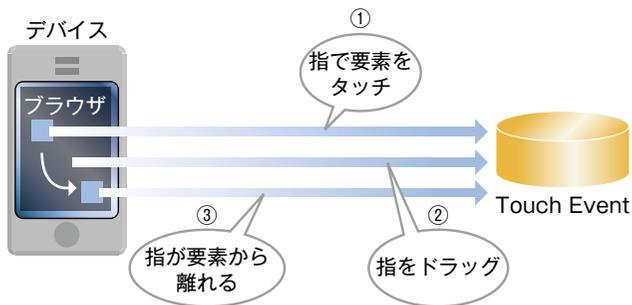
File APIは、ブラウザからローカルのファイル进行操作するためのAPIです。これによって、画像ファイルやテキストファイル、Office関連のファイルなどに読み込んでブラウザ上に表示したり、ブラウザ上で内容を修正して保存したりすることが可能となります。また、ローカルフォルダ上にフォルダやファイルを作成することも可能です。

ただし、現時点では、ブラウザによって実行できない操作もあります。この機能は単独でも有用ですが、後述のDrag & Drop APIと組み合わせて使用すると、さらに効果的なWebアプリケーションを作成することができます。

## ■Touch Event

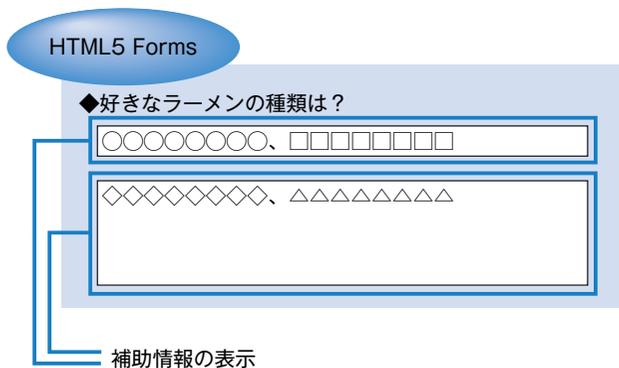
Touch Eventは、ブラウザで画面を直接指でタッチして操作する場合の入力処理(タッチイベント)を取得するためのAPIです。iPhoneやAndroidなどのモバイル端末での操作を想定した機能です。

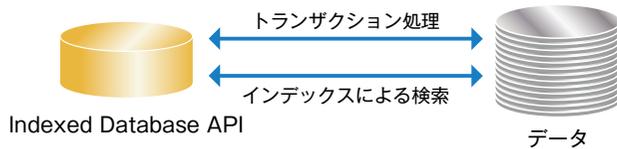
サポートされているタッチイベントや取得されるタッチイベント精度などは、ブラウザの実装によって大きく異なる可能性があります。



## ■HTML5 Forms

HTML5 Formsは、HTML5で拡張・強化されたフォーム機能の総称です。入力チェックやカラーピッカー、スライダー、カレンダー等、従来はJavaScriptで実装する必要のあったさまざまな機能を手軽に利用できるようになりました。





## ■ Indexed Database

Indexed DatabaseはSQLを用いずにローカルでデータベースを実現するためのAPIです。Web Storageと同じようにkey/value型でデータを保存しますが、インデックス機能やトランザクション機能といったデータベースとしての基本的な仕組みを備えています。

ただし、Indexed Database APIへの対応はブラウザによって異なります。

### One Point

現状、HTML5では本格的なRDB(リレーショナルデータベース)が実装されないままW3Cの勧告に到る可能性が濃厚です。実は、Web SQL Database APIというリレーショナルデータベースのAPIの仕様策定が進められていましたが、W3Cによる方向転換もあり、現時点では非推奨となっています。

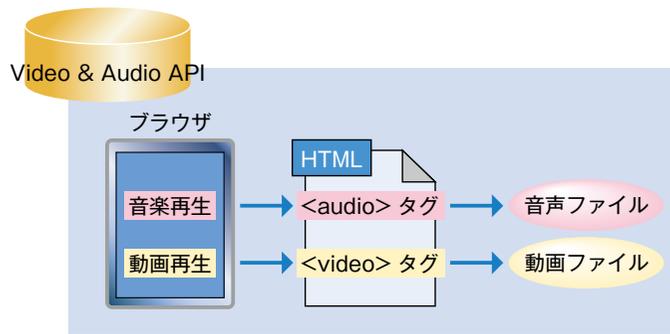
一方で、Web Storageも、大規模なWebアプリケーションとしては容量的に中途半端である点や、使える機能が限定されている点など、データベースとしての使用には課題があります。

そこで、各ブラウザへの実装が期待されているのが、Indexed Databaseです。

## ■Video & Audio

ブラウザのみで動画や音声を再生するためのAPIです。video要素やaudio要素を使うことで、プラグインなしで簡単にメディアを再生することが可能です。

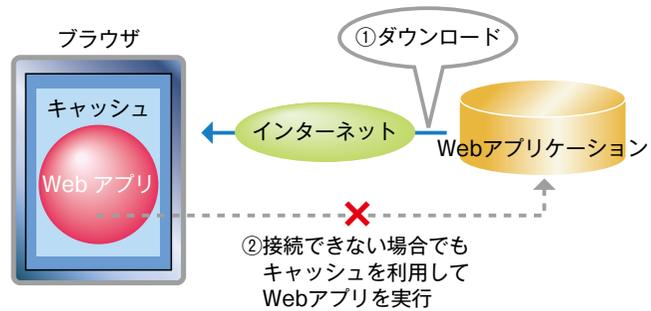
ただし、現状では、各ブラウザによってサポートしているメディア再生フォーマット(コーデック)が異なるため、ブラウザごとの対応が必要となります。



## ■Offline Web Application

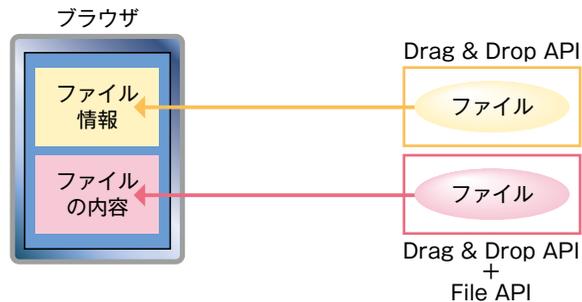
オフライン状態でWebアプリケーションを動作させるためのAPIです。HTML5ではモバイル環境での利用を考慮し、オフライン環境下でもWebアプリケーションの機能を継続して利用するための仕様を策定しました。

ブラウザは、.appcacheという拡張子を持つ「キャッシュマニフェストファイル」の記述に従って、オンライン時に必要なファイルをダウンロードしておき、オフライン時はローカルのキャッシュを利用してアプリケーションの実行を継続します。



## ■Drag & Drop API

ブラウザにファイルをドラッグ&ドロップして、ファイル情報を操作するためのAPIです。前述のFile APIを組み合わせることで、ブラウザ上にファイルの内容を表示するといった便利な使い方が可能です。



## 1-4-5 アプリ開発に役立つ フレームワークと ツールについて

ここではHTML5でWebアプリケーションを開発する際に役立ついくつかのツールとフレームワークを開発シーンごとに分類して紹介します。

まず、HTML5によるWebアプリケーション開発に役立つフレームワークおよびプラットフォームを紹介します。



◆PhoneGap(モバイル向けハイブリッドWebアプリ開発のフレームワーク)  
<http://phonegap.com/>



◆Monaca(スマートフォンアプリ開発のプラットフォーム)  
<http://monaca.mobi/>



◆Sencha Touch(モバイル向けHTML5対応JavaScriptフレームワーク)  
<http://www.extjs.co.jp/products/touch/>

次に各種ジェネレーターを紹介  
します

ジェネレーターとは、自動的に  
プログラムなどを作成してくれる  
プログラムのことです。

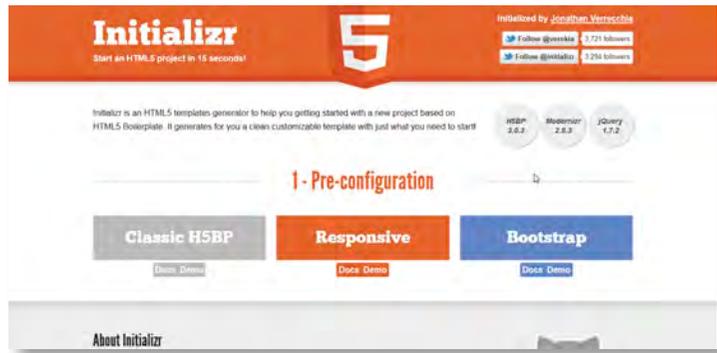
第1章

第2章

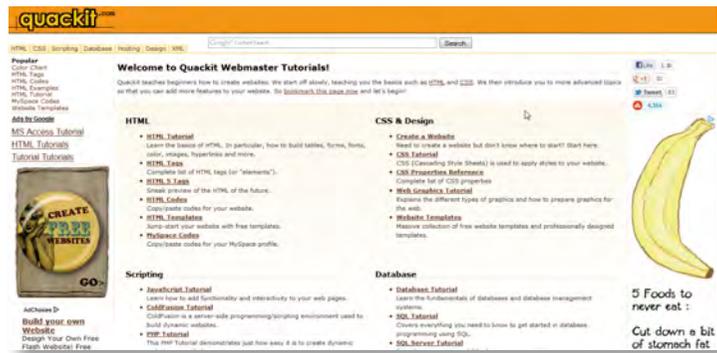
第3章

第4章

第5章



◆Initializr (HTML5テンプレート作成)  
<http://www.initializr.com/>



◆quackit.com (HTML5タグジェネレーター)  
<http://www.quackit.com/>



◆CSS3 Generator (CSS3ジェネレーター)  
<http://www.css3generator.com/>



◆HTML5 & CSS3 Readiness(ブラウザ対応状況)  
<http://html5readiness.com/>



◆When can I use(ブラウザ互換表)  
<http://caniuse.com/>

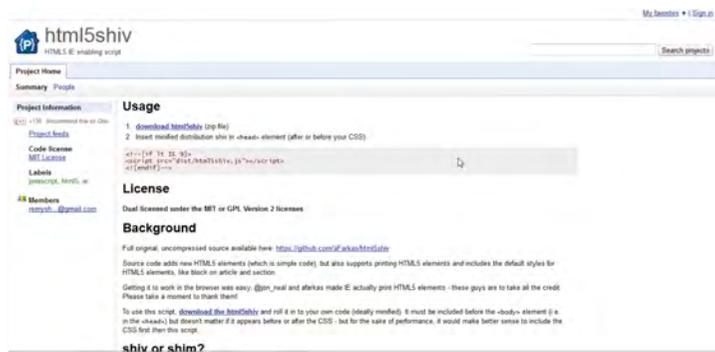


◆The HTML5 test(実装状況チェッカー)  
<http://www.html5test.com/>

最後に、ブラウザ互換・対応状況が確認できるサイトや、ファイルコンバートツールのサイトを紹介します。



◆HTML5 API チェッカー (HTML5 API チェッカー)  
<http://www.html5.jp/tutorial/apichecker/index.html>



◆html5shiv (HTML5 互換パッチ for IE)  
<http://code.google.com/p/html5shiv/>



◆Wallaby (fla ファイルコンバーター)  
<http://labs.adobe.com/technologies/wallaby/>