

Google Developer Day 2009



HTML5により拓かれる 次世代Web

田村健人, Google
2009-06-09

Google
Developer
Day2009

1. HTML5の概要
2. 規格の概要
3. 新機能の解説
 - Web Workers
 - Web Sockets
 - Web Storage
 - アプリケーションキャッシュ
 - フォームの強化
 - メディア
 - アプリケーション向け要素
 - 構造化文書向け要素



1. HTML5の概要



HTML5はアプリケーションプラットフォーム

- HTML4までは、基本的には「構造化文書を定義する」という姿勢だった
- HTML5は構造化文書として強化されると共に、
 - アプリケーションを作るための要素が追加
 - 挙動の定義
 - 様々なAPIを定義
 - HTML DOMのプロパティの他、文書とはまったく関係のないAPIも定義
- アプリケーションを動かすための標準プラットフォームを定義する規格へ
 - 長期的な目標は、OSネイティブのアプリケーションに匹敵する機能をHTMLとスクリプトで作成できるようにすること

HTML5がウェブの使い易さを向上させる

- エンドユーザーにとってのHTML5のメリット
 - オフライン関連機能
 - SVG, MathML, canvas, audio, video などによる表現力向上
 - メニューやフォーム強化などによる利便性向上
 - スクリプトのコード減少によるアプリケーションのロード高速化と動作高速化

HTML5がウェブ開発を容易にする

- 開発者にとってのHTML5のメリット
 - 前ページのメリットをユーザに提供できること
 - プラグインを使わないと実現できなかったことの一部がHTMLとスクリプトで可能になる
 - メニューやフォーム検証などに大量のスクリプトコードが必要だったのが、HTMLの記述だけで可能になる
 - 規格で曖昧だった点や、まったく標準規格が存在しなかった部分が規格化されることで、ブラウザ間の非互換が減少

HTML5の主な新機能

- Web Workers: スクリプトでマルチスレッド
- Web Sockets: サーバとのメッセージ交換
- Web Storage: クライアント側へのデータ保管
- アプリケーションキャッシュ
- フォームの強化
- メディア要素: canvas, video, audio
- アプリケーション向け要素: menu, datagrid, progress, meter
- 構造化文書向け要素

HTML5の主な新機能 (続き)

- ドラッグ&ドロップ機構
- contenteditable属性の標準化
- ドキュメント間メッセージング
- ルビ
- data-* 属性: アプリケーション依存のデータを埋め込む
- SVGとMathMLの埋め込み

今からでもHTML5を始めましょう

- iPhoneやAndroidなど特定プラットフォーム向けのウェブアプリケーションなら、すぐに恩恵を受けられる
 - アプリケーションキャッシュ、canvas など
- 一般向けでも、「無くても構わないがあると便利」な機能はHTML5で実装しても現実的
- HTML5のドラフトに目を通してフィードバックする
- オープンソースブラウザのHTML5対応に貢献する

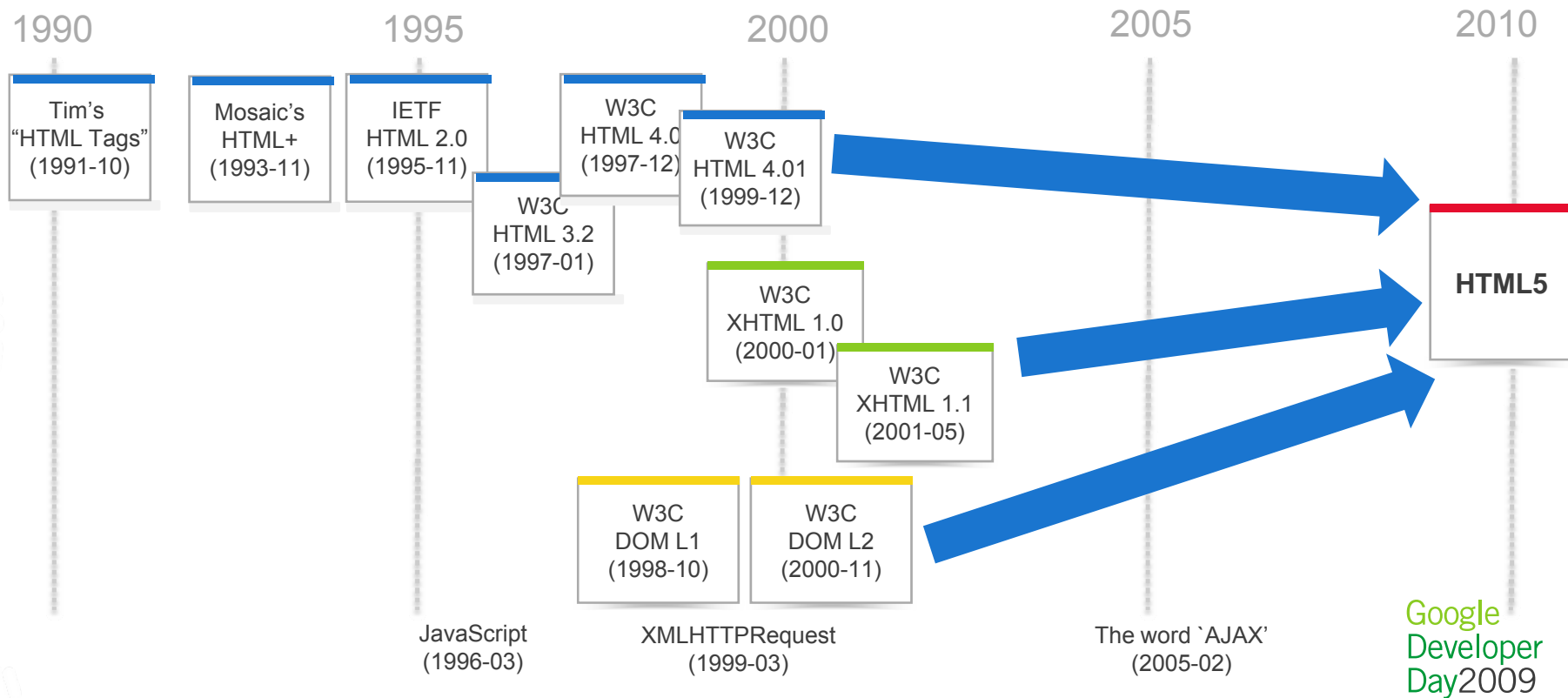


2. 規格の概要



HTML5は10年ぶりのHTML新バージョン

- HTML4、XHTML 1.1、DOM Level 2 HTML を包含し、機能を加える規格
- XHTML 2 および XForms は含まない
- W3Cの規格となるが、実作業はWHATWG



HTML5は4つの規格で構成

- 元は1つだったが、あまりに巨大なため分割された
 - `HTML5 - A vocabulary and associated APIs for HTML and XHTML`
 - `Web Storage`
 - `Web Workers`
 - `The Web Sockets API`
- 現在、すべてWorking Draftの状態
- Editorによると、2009年10月に最後のWorking Draft
 - 参考: W3C の標準化プロセス
Working Draft (複数回) → Last Call Working Draft → Candidate Recommendation → 実装試験 → Proposed Recommendation → 投票 → ディレクター承認 → Recommendation
Recommendationになるまでは内容が変わる可能性がある

HTML表記とXHTML表記

- HTML表記とXHTML表記をサポートする
- SVGとMathMLはXHTML表記のときのみサポート

```
<!DOCTYPE html>
<html>
  <meta charset="UTF-8">
  <title>...
```

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <title>...
```

- HTML表記ではDOCTYPE宣言が必須
- しかし、HTML表記はSGML準拠ではなくなった
- XML表記の名前空間はXHTML1.xと共通



3. 新機能の解説



Web Workers

- バックグラウンドでスクリプトを実行する仕組み
- 毎回新しいスレッドを生成するWorkerと、複数のウィンドウで共有できるSharedWorker
- ウィンドウを閉じてでも動き続けるpersistent workerを議論中

```
// Workerの生成
var worker = new Worker('foo.js');
worker.postMessage(data); // workerへメッセージを送る
// Workerからのメッセージを受け取るには onmessage ハンドラで
worker.onmessage = function(event) {
  ... = event.data;
};
```

```
// foo.js Workerの実装
// 他からのメッセージ受け取り
onmessage = function(event) { ... };
...
...
// 結果をどこかに返す
postMessage(data);
```

Web Sockets

- 専用のプロトコルでサーバと文字列の送受信をする仕組み
 - HTTPではないので、サーバ側での実装も必要
- XMLHttpRequestと比べると、オーバーヘッドが小さくリアルタイム性が高い

```
// サーバへ接続
var socket = new WebSocket('ws://example.com/channel1');

// サーバへ文字列のメッセージを送る
socket.postMessage('hello.');
```



```
// サーバからのメッセージを受けとるには、onmessage ハンドラで
socket.onmessage = function(event) {
  var str = event.data;
};
```

Web Storage

- ページ遷移しても消えない、クライアント側のデータ格納機能
- データ形式の違いで2種類のタイプ
 - 文字列の key-value ペアを扱う Storage 型
 - SQLを扱う Database 型
- Storage型はスコープの違いで2種類
 - window.localStorage
 - window.sessionStorage
- Database型はインターフェイスの違いで2種類
 - 同期型と非同期型

Web Storage – Storage型

- window.localStorage
 - サイト別に保持される key-value ペア。スクリプトやユーザが消さない限り、ローカルマシンに保持され続ける。
- window.sessionStorage
 - サイト別、かつウィンドウ別に保持される key-value ペア。複数のウィンドウで同じサイトを開いていても、別のインスタンスとなる。ウィンドウが閉じられると消滅する。
- 使い方は同じ
 - storage.getItem(key)
 - storage.setItem(key, value)
 - storage.removeItem(key)
 - storage.length
 - storage.key(index)
 - storage.clear()

Web Storage – Database型

- localStorageと同様、サイト別に保持される記憶領域
- SQLが使用可能
- 同期実行型と非同期実行型の2つのインターフェイス

```
var db = window.openDatabaseSync();
var resultSet = db.transaction().executeSql(
    'SELECT * FROM Table ...', []);
```

```
var db = window.openDatabase();
var sqlCallback = {};
sqlCallback.handleEvent = function(transaction, resultSet) {
    ...
};
var transactionCallback = {};
transactionCallback.handleEvent = function(transaction) {
    transaction.executeSql('SELECT * FROM Table ...', [],
        sqlCallback);
};
db.transaction(transactionCallback);
```

アプリケーションキャッシュ

- キャッシュしても良いURL、しないで欲しいURLを宣言
- ブラウザがオンライン時にキャッシュされていないURLを先読みし、オフライン時にはキャッシュを使う

```
CACHE MANIFEST
settings.html
images/logo.gif
js/main.js
js/settings.js
```

```
NETWORK:
cgi/submit.cgi
```

```
<!DOCTYPE html>
<html manifest="app.manifest">
...
  
```

フォームの強化

- input要素の新タイプ

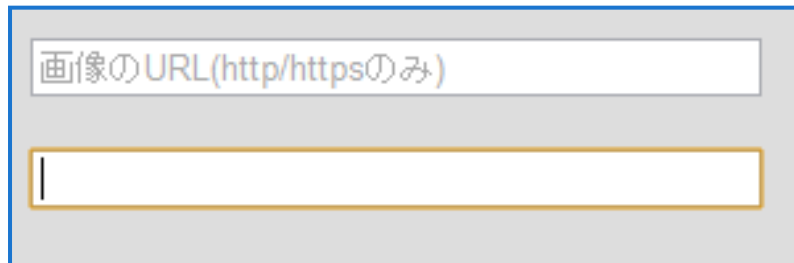
- `search` (検索語の入力用), `tel` (電話番号), `url` (URL),
`email` (電子メールアドレス), `month` (年と月), `week` (年と週),
`time` (時分秒), `datetime` (タイムゾーン非依存の年月日時分秒),
`datetime-local` (タイムゾーン依存の年月日時分秒),
`number` (数値), `range` (数値だが、`number`とは見た目が異なる),
`color` (RGB各8bitの色)

- フォーム検証機能

- フォームがsubmitされる前に、各フィールドの最小値・最大値・文字列パターン・入力必須かどうかをチェック
- 検証をパスしないとsubmitできない
- スクリプトで任意のタイミングで検証することも可能

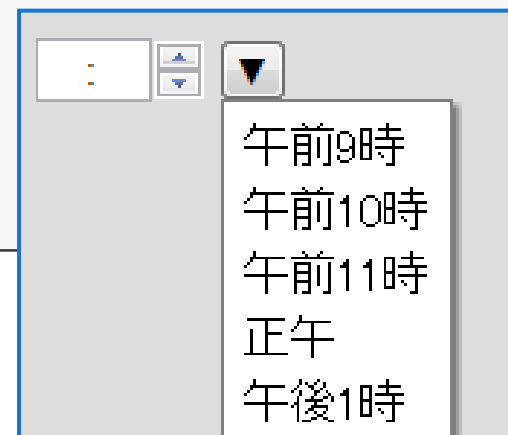
フォームの例

```
<input type="url" placeholder="画像のURL(http/httpsのみ)"  
  pattern="https?://.+¥.(gif|png|jpg|jpeg)">
```



画像のURL(http/httpsのみ)

```
<input type="time" min="9:00" max="21:00" step="1800"  
  required list="timelist">  
<datalist id="timelist">  
  <option value="9:00">午前9時</option>  
  <option value="10:00">午前10時</option>  
  ...  
  <option value="21:00">午後9時</option>  
</datalist>
```



午前9時
午前10時
午前11時
正午
午後1時

メディア要素: canvas

- スクリプトによる2次元グラフィックス描画

```
<canvas id="c1" width="320" height="240"></canvas>
<script>
var context = document.getElementById('c1').getContext('2d');
context.fillStyle = '#88ccff';
context.fillRect(10, 10, 300, 220);
</script>
```

- 現在定義されている機能

- 座標系の拡大縮小、回転、行列変換、原点の移動
- 色、透明度、ブラシパターン、グラデーション
- 線の太さ、先端の形状
- 矩形、直線、ベジェ曲線、円弧の描画
- 文字描画
- 矩形コピー
- 他のcanvas要素、img要素、video要素からの矩形コピー
- canvas内のピクセルの色の取得

メディア要素: videoとaudio

- img要素と同じように、src属性に動画URLや音声URLを指定する
- または、source要素を複数指定してブラウザに選ばせる

```
<video src="o3d.mp4" width="320" height="240" autobuffer autoplay
  loop>
  <p>HTML5対応ブラウザが必要です。</p>
</video>
```

```
<video width="320" height="240" autobuffer autoplay controls>
  <source src="o3d.mp4">
  <source src="o3d.wmv">
  <source src="o3d.mov">
</video>
```

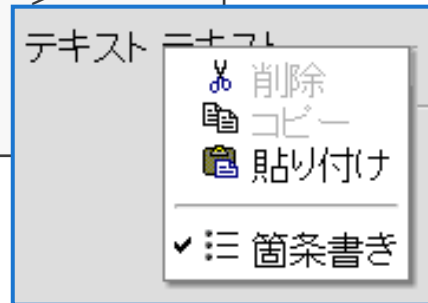
メディア要素: videoとaudioのAPI

- `.duration`: メディアの再生終了時間
- `.currentTime`: 現在の再生位置。代入可能。
- `.volume` / `.muted`: 音量
- `.playbackRate`: 再生速度
- `.load()`: バッファリング開始
- `.play()`: 再生開始
- `.pause()`: 一時停止と再開
- etc.

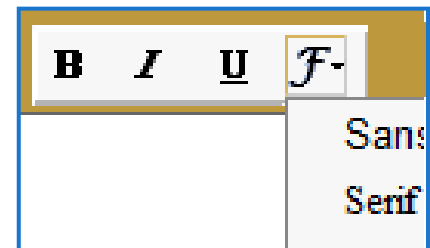
アプリケーション向け要素: menu要素

- あらゆる要素にコンテキストメニューを定義
- ツールバーを簡単に定義

```
<menu id="menu1" type="context">  
  <command label="削除" icon="cut.png" onclick="cut()" disabled>  
  <command label="コピー" icon="copy.png" onclick="copy()" disabled>  
  <command label="貼り付け" icon="paste.png" onclick="paste()">  
  <hr>  
  <command label="箇条書き" checked onclick="toggleListing()">  
</menu>  
  
<div contextmenu="menu1">テキスト テキスト</div>
```



```
<menu type="toolbar">  
  <command icon="bold.png" onclick="bold()">  
  <command icon="italic.png" onclick="italic()">  
  <command icon="underline.png" onclick="underline()">  
  <command icon="font.png" contextmenu="fontmenu">  
</menu>
```



アプリケーション向け要素: datagrid要素

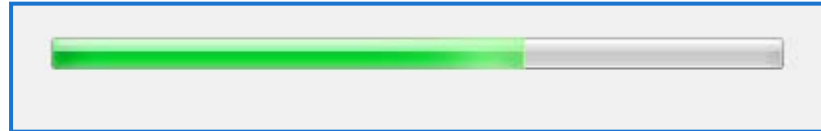
- まとまったデータを表現する
- リスト構造、木構造、表を扱う
- ソートや編集が可能
- フォームのようにサーバに送信するのではなく、大量のデータの表示やユーザが編集したデータをスクリプトで処理するのが目的

ID	Stars	Pri	Area	Type	Status	Summary + Labels	Modified	Owner	Msto
18	838	2	Extensions	Feature	Available	Wishlist: Chrome does not have an add-on system	Oct 2008	all-bugs-test@chromium.org	X
84	336	2	BrowserUI	Feature	Started	RSS or Atom support needed DesignDocNeeded	May 19	@chromium.org	3
19	246	2	BrowserBackend, Enterprise	Feature	Assigned	Automatic integrated windows authentication (aka automatic NTLM / Negotiate Auth support) NewHTTP	Apr 23	@chromium.org	X
60	220	2	BrowserUI	Feature	Available	Add "Remove" to the menu on items in the download bar HelpWanted	Mar 24	---	X
181	172	3	BrowserUI	Feature	Available	Feature request: mouse gesture support	Nov 2008	---	X
489	167	2	BrowserUI	Feature	Unconfirmed	Google Bookmark addon is not supported	Jan 07	all-bugs-test@chromium.org	---
172	160	2	BrowserUI	Feature	Available	Missing a Print Dialog	Dec 10	@chromium.org	X

アプリケーション向け要素: progressとmeter

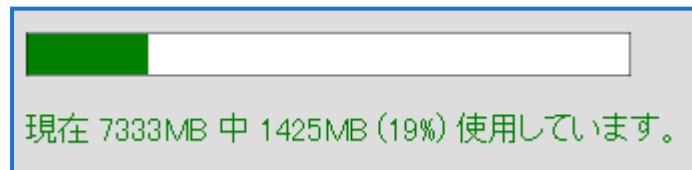
- progress要素

- あるタスクが進行中であることを表示するための要素
- 最終値と現在値を指定するタイプと、最終値が不明なタイプ



- meter要素

- 増減する量を表示するための要素
例: データホスティングサービスでの使用容量



構造化文書向け要素

- section, article, aside, header, footer, nav, time など
 - 文書の見た目にはほとんど影響がない
 - HTMLソースの可読性向上や、機械処理に有用
 - 例: 検索エンジンがarticle要素内のテキストを優先できる



まとめ



まとめ

- HTML5は、アプリケーションプラットフォーム
- ブラウザを限定できる状況なら一部の機能は今から利用可能
- まだWorking Draftで、規格の内容は追加・修正が可能
- 新機能の一部を紹介

Google Developer Relations Japan

- Google の デベロッパーコミュニティサイト
 - Google Developer Day セッションのスライドやビデオ
 - イベント情報

The screenshot shows the Google Developer Relations Japan website. The header includes the Google logo and the text 'Google Developer Relations Japan'. A search bar is located in the top right corner. The left sidebar contains a 'ホーム' link and a 'サイトマップ' link. Below this, there is a large graphic for 'あと 19 日で Google Developer Day 2009 Japan'. The main content area is divided into several sections: 'ホーム' (Home) with a welcome message and a 'トピック' (Topics) section listing Google Developer Day and Hackathons; 'API Expert' section; 'コミュニティ' (Community) section listing various developer groups; 'セミナー/イベント' (Seminar/Event) section with a calendar view for June 2009; and 'リファレンス/レポート' (Reference/Report) section. A photo gallery at the bottom shows a 'Hackathon in a Box' event.

<http://sites.google.com/site/devreljp/>

アンケートにご協力ください

- Google Developer Day をより充実したものとするためにアンケートにご協力ください。
- アンケートは2種類あります。
 - 全体アンケート
 - セッションアンケート
 - セッションアンケートはお手元の用紙でも回答できます。
 - アンケートは Google Developer Relations Japan で回答できます。

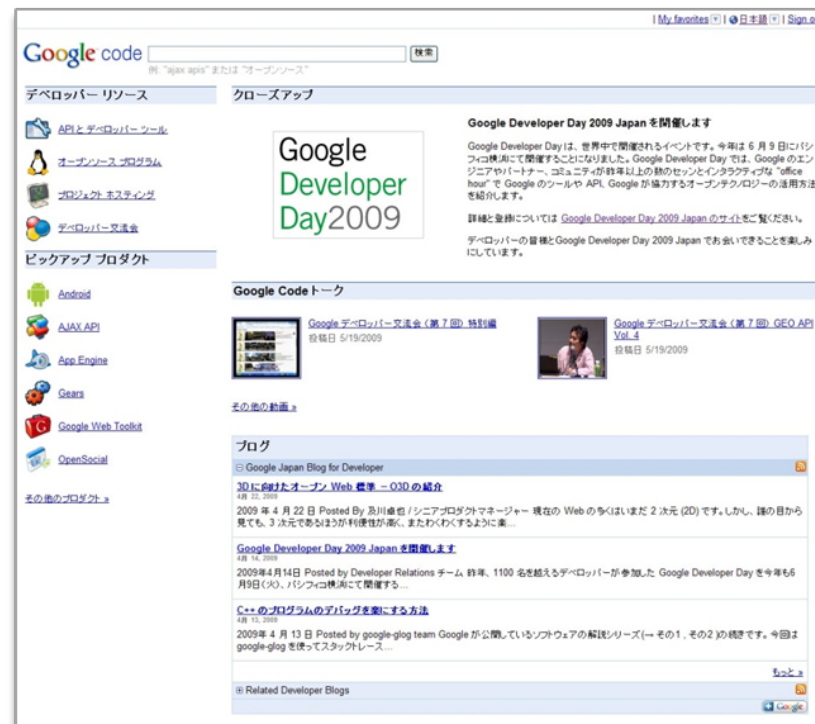
<http://sites.google.com/site/devreljp/>

Questions?

- オフィスアワー
 - 14:10～14:40は、この部屋の外で質問コメントなどをお受けします。

code.google.com

- Google の デベロッパー向けサイトをご利用ください。
 - 日本語のドキュメント
 - 日本語のデベロッパーコミュニティ



<http://code.google.com/intl/ja/>

Google
Developer
Day 2009