けいはんな情報通信オープンラボ研究推進協議会

HTML5セミナー

2010年3月3日

有限会社futomi
代表取締役 羽田野 太巳
http://www.futomi.com/
http://www.html5.jp/
http://twitter.com/futomi

HTML4 & XHTML

旧来のHTML

- マークアップの規定が中心。
- 論文をベースとした語彙が中心。
- ブラウザー向けの要件は規定されず。

ブラウザーの互換性

- 規定違反のマークアップに遭遇したときの扱いが違う。
- JavaScriptコードの実行結果が異なる。

BOM/DOM仕様のドキュメント

- 多くの仕様が策定されていない。
- 仕様があったとしても古く、現状にあっていない。
- ブラウザーごとに解釈が違う。

ウェブ・アプリケーションの壁

- ブラウザーの互換性対策
- ブラウザーによって機能がない

- リッチなアプリケーション制作に限界。
- リッチなアプリケーションになるほど、ブラウザー互換性の対策に、大きな労力を要する。

ウェブを再定義

- ブラウザーごとの非互換性を解決
- リッチなウェブ・アプリケーションに求められるAPIを新たに開発
- 現代のウェブの用途に合わせてマークアップの語彙を追加

HTML5の誕生

HTML5とは

HTML5をひとことでいうと

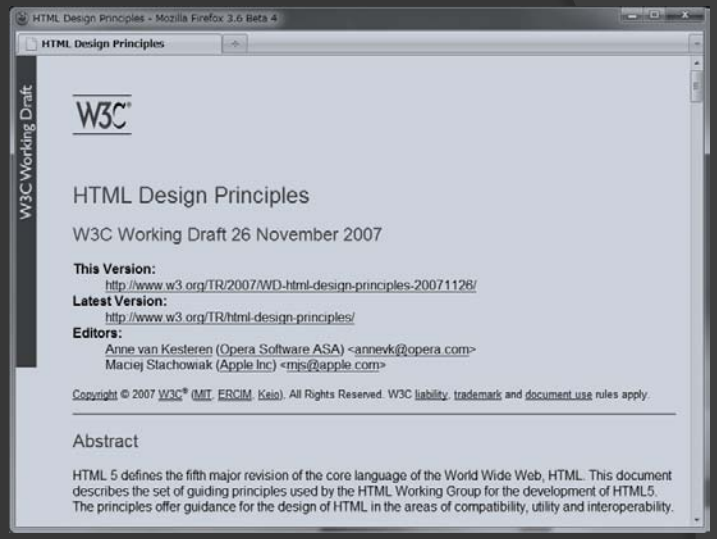
- Webプラットフォーム
- Open Web

HTML5の特徴

- マークアップ
  - 下位互換性
  - 人が見ても分かりやすい。
  - 誰でも簡単に作れる。
- ウェブ・アプリケーション
  - 互換性のあるAPI
  - 新たなAPI

HTML5の設計原則

- 互換性
- 実用性
- 相互運用性
- ユニバーサル・アクセス



- http://www.w3.org/TR/html-design-principles/

互換性

- コンテンツの互換性
  - 過去のコンテンツが利用できなくなるわけではない。
- ブラウザーの互換性
  - どのブラウザーでも同じように見える。
- 再利用
  - ブラウザー独自の機能を標準として採用
  - 車輪の再発明をしない
- 革新より発展
  - 理想を求めてまったく新しいものを作るのではなく、今あるものを改良することを優先する
  - XHTML 2.0の反省？

実用性

- 理想より実用性を優先
- 優先順位
  - ユーザー、ウェブ制作者、ブラウザー・ベンダー、仕様書、理想
- セキュリティー重視

相互運用性

- ブラウザーの挙動の互換性を保証
- 複雑さは、ブラウザー間の互換性を損なう可能性がある
- 複雑な仕様は、ウェブ制作者が利用をためらう原因にも
- シンプルであることを重要視

ユニバーサル・アクセス

- すべてのメディアに
  - PC、モバイル端末、テレビ、ゲーム機など
- すべての言語に
  - 英語だけでなく、日本語やアラビア語なども考慮
- すべての人に
  - ハンディキャップを持った人でも利用できる

- アクセシビリティが重要

W3Cの勧告の条件

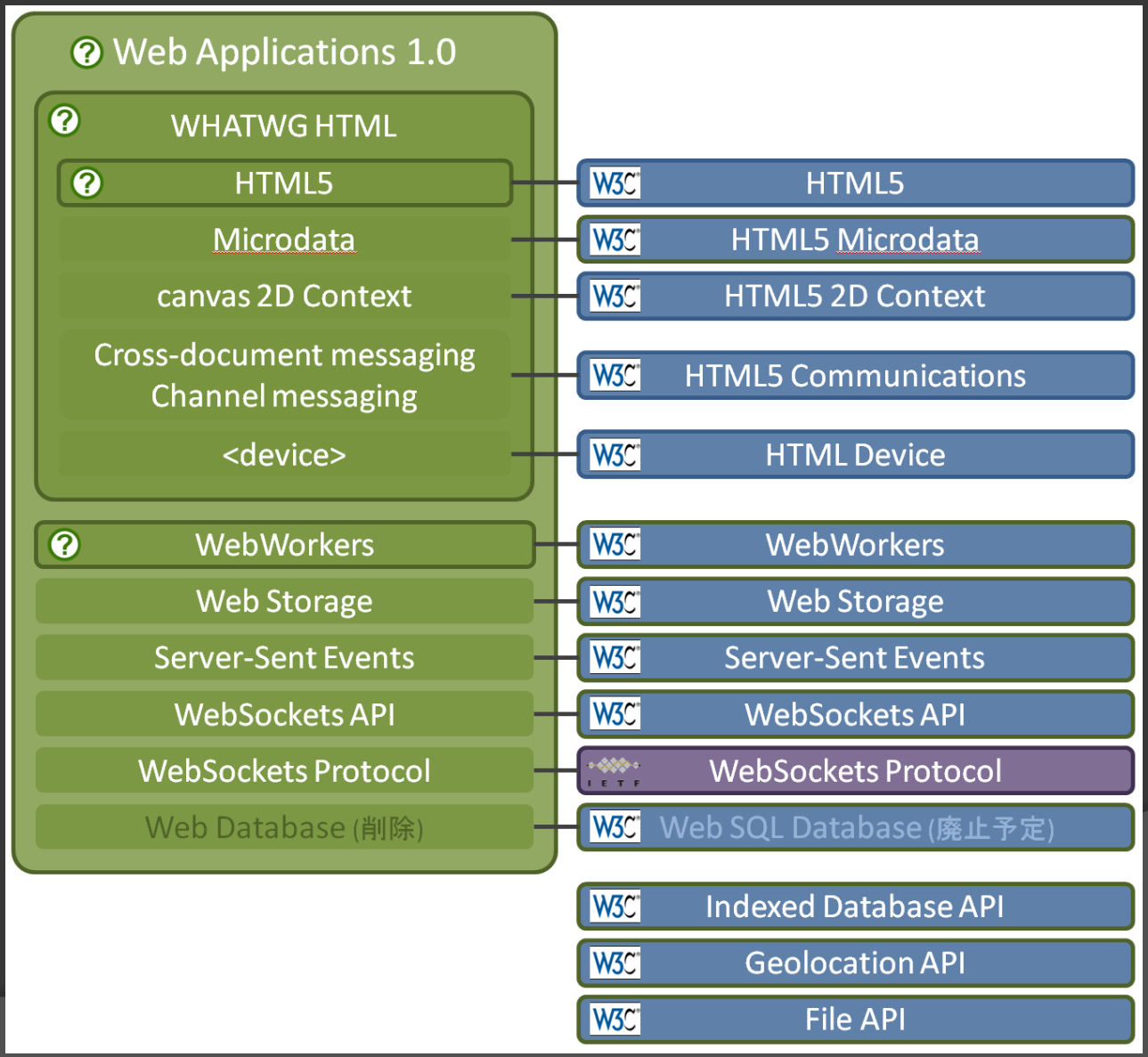
- 2つ以上のブラウザーがHTML5を実装しなければ勧告にできない。
- 2009年から、各ブラウザー・ベンダーはHTML5の実装を強く推し進めている。
- MicrosoftもInternet Explorer 9でHTML5の実装を推し進めることを表明（Microsoft PDC09）
- http://microsoftpdc.com/

W3C勧告までのマイルストーン

- 2010年01月：最終草案※
- 2010年12月：勧告候補
- 2012年01月：勧告案
- 2012年03月：勧告

W3C HTML Working Group
http://www.w3.org/html/wg/
※2010年3月時点でまだ最終草案に至っていない

HTML5仕様の関連


Web Applications 1.0
WHATWG HTML
HTML5
Microdata
canvas 2D Context
Cross-document messaging
Channel messaging
<device>
WebWorkers
Web Storage
Server-Sent Events
WebSockets API
WebSockets Protocol
Web Database (削除)
HTML5
HTML5 Microdata
HTML5 2D Context
HTML5 Communications
HTML Device
WebWorkers
Web Storage
Server-Sent Events
WebSockets API
WebSockets Protocol
Web SQL Database (廃止予定)
Indexed Database API
Geolocation API
File API

HTML5の範囲

- HTML5の範囲は曖昧
- もともとはWHATWGが開発した仕様の範囲を表す。
- W3Cが開発したAPIも含めてHTML5と呼ぶことが多い。
- 文脈によっては、CSS3も含めて、次世代ウェブ・テクノロジーの代名詞として使われることもある。

Forms

Webアプリの入り口

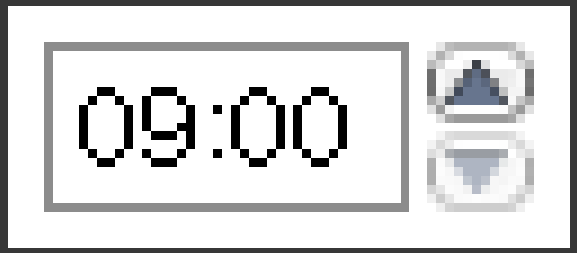
- 新たに追加されたフォーム・コントロール
- Webアプリケーションが、デスクトップ・アプリケーションに近いUIを実現

日付

- <input type="date" name="dt" />

時間

- <input type="time" name="tm" />

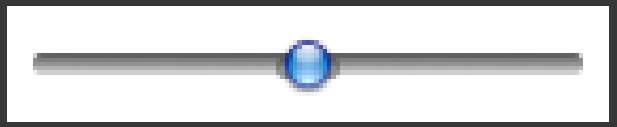


数値

- <input type="number" name="mb" />

スライダー（数値）

- <input type="range" name="mb" />

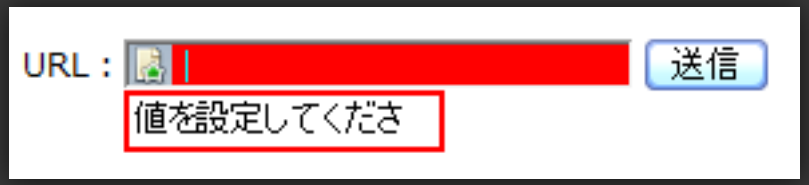


色

- <input type="color" name="cl" />

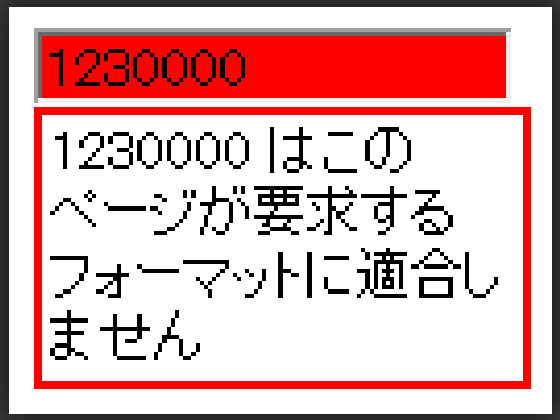
必須

- <input type="url" name="url" required>



正規表現

- <input type="text" pattern="¥d{3}¥-¥d{4}">



Canvas

<canvas>

- ウェブページに図を描くことができる。
- canvas要素にAPIが規定されている。
- JavaScriptからAPIを通して図を描く。

```
<canvas width="300" height="150">
  <!-- ここにフォールバック・コンテンツを入れます。 -->
</canvas>
```

Canvas 2d context

- CanvasのAPIを規定
- HTML5仕様から分離
- 現在は2dのみ
- 将来的には3d、SVGとの連携、アクセシビリティ（WAI）を考慮
- http://dev.w3.org/html5/2dcontext/

CanvasとSVGの違い

- Canvas
  - JavaScriptを使って描画
  - 描いてしまった図を個別に認識できない。
  - 描画そのものは高速
  - ピクセル操作が可能
- SVG
  - XML形式のマークアップで図を表現
  - JavaScriptから、各要素にアクセス可能
  - 要素が多すぎると重い
  - ピクセル操作はできない

```
/* 半透明度を指定 */
ctx.globalAlpha = 0.7;
/* 円 #1 */
ctx.beginPath();
ctx.fillStyle = 'rgb(192, 80, 77)'; // 赤
ctx.arc(70, 45, 35, 0, Math.PI*2, false);
ctx.fill();
/* 円 #2 */
ctx.beginPath();
ctx.fillStyle = 'rgb(155, 187, 89)'; // 緑
ctx.arc(45, 95, 35, 0, Math.PI*2, false);
ctx.fill();
/* 円 #3 */
ctx.beginPath();
ctx.fillStyle = 'rgb(128, 100, 162)'; // 紫
ctx.arc(95, 95, 35, 0, Math.PI*2, false);
ctx.fill();
```

Canvasによるアニメーション

- アニメーションに特化した機能はない
- Canvasは描画速度が非常に高速
- パラパラマンガを作る（1コマずつ全体描画）
- setTimeout() やsetInterval()を組み合わせる

VideoとAudio

<video>/<audio>

- プラグインなしにビデオやオーディオの再生を可能とする。

```
<video id="video" src="sample.mp4">
 <p>ご利用のブラウザーでは再生できません。
 <a href="test.mp4">こちら</a>
 からダウンロードしてください。</p>
</video>
```

Chrome 4

Firefox 3.6

Safari 4

Video/AudioのAPI

- ビデオやオーディオの再生は、JavaScriptから制御可能。
- イベント、メソッド、プロパティが仕様で規定される。
- 独自のビデオ・コントロール・インタフェースを構築することが可能。

Web Workers

Web Workers

- JavaScriptの処理をバックグラウンドで実行
- ブラウザーのUIをブロッキングしない
- すでにFirefox 3.5+, Chrome 3+ で利用可能。

- http://www.w3.org/TR/workers/

workerの種類

- 専用ワーカー（Dedicated workers）
  - 1つのブラウジング・コンテキスト専用で動作するワーカー
  - Chrome 3+, Firefox 3.5+がサポート
- 共有ワーカー（Shared workers）
  - 複数のブラウジング・コンテキストで共有して動作するワーカー
  - Chrome 4+がサポート予定

Web Storage

Web Storage

- ブラウザーにデータを蓄積
- Cookieと同様にキーと値のペアを保存
- 数MBのデータも保存可能
- 保存されたデータは、サーバーに送信されない
- オフライン・アプリケーション、サーバーの負荷低減などに有効
- すでにInternet Explorer 8、Firefox 3.5+、Chrome 3+、Opera 10.5 pre-alphaで利用可能。
- http://www.w3.org/TR/webstorage/

セッション・ストレージ

- ブラウジング・コンテキスト単位にデータを保存・管理。
- 保存されたデータは、ブラウジング・コンテキストが終了閉じるまで保存。その後は破棄。
- Cookieに例えれば、有効期限を指定しなかった場合と同じ。
- var o = window.sessionStorage;

ローカル・ストレージ

- オリジン単位でデータを保存・管理。
- オリジンが少しでも違うサイトでは、別々のストレージとして管理。
- お互いのデータを参照することはできない。
- ブラウザーを閉じても、そのデータは消えない。
- var o = window.localStorage;

その他のAPI

Indexed Database API

- ブラウザーにデータを蓄積
- データのリスト
- データの検索
- 大量のレコード数を考慮
- 2009年9月より策定が始まる。
- 策定が始まって間もないため、対応したブラウザはない
- http://www.w3.org/TR/IndexedDB/

Web SQL Database

- ブラウザーにデータを蓄積
- データ読み取り・保存にSQLを利用
- 複雑なデータ構造の保存に有効
- すでにSafari 4+、Chrome 4+、Opera 10.5 pre-alphaで利用可能
- データベース・エンジンに実装されているSQLの方言を懸念される。
- ブラウザー・ベンダーの意見がまとまらず、仕様策定が実質的に停止。
- http://www.w3.org/TR/webdatabase/

Server-Sent Events

- リアルタイムにサーバーからのメッセージを受信するAPI
- HTTPを利用する
- コネクションを明示的に閉じない限り、再接続される。
- サーバー側では、text/event-streamで出力する。
- サーバーから送信するデータは「data:メッセージ」
- Perl/CGIやPHPなどで、簡単にサーバー側の仕組みを用意できる
- すでにWebKit Nightly Buildsで利用可能
- http://www.w3.org/TR/eventsource/

WebSockets API

- サーバーと双方向に通信するためのAPI
- 全二重である点が重要
- オーバーヘッドが少ないため、効率的にメッセージの送受信が可能となる
- チャットなどのリアルタイム双方向通信を必要とするアプリケーションに有効
- サーバー側では、WebSocketsプロトコルに対応する必要がある。
- すでにChrome 4で利用可能。
- http://www.w3.org/TR/websockets/

Drag & Drop / File API

- File APIは、fileタイプのinput要素に指定されたファイルを扱うためのAPI
- ドラッグ＆ドロップと組み合わせることで、デスクトップ上のファイルを扱うことも可能。
- ファイルのメタ情報だけでなく、ファイルの中身（データ）をJavaScriptで扱うことも可能。
- ブラウザーとデスクトップをシームレスにつなげるキー・テクノロジーとなる。
- すでにFirefox 3.6+ で利用可能。
- http://www.w3.org/TR/FileAPI/

Geolocation API

- 位置情報を取得するAPI
- 緯度、経度、高度、進行方向、移動速度
- 日本国内でも各キャリアが位置情報サービスを提供しているが、サーバーとの連携が必要。さらにキャリアごとに互換性が一切ない。
- Geolocation APIはJavaScriptだけで位置情報を取得でき、標準化されたAPIであるため、将来的にはデバイス・フリーの位置情報アプリケーションが期待できる。
- すでにFirefox 3.5+、iPhone 3.0で利用可能
- http://www.w3.org/TR/geolocation-API/

HTML5 Showcase

HTML5 Canvas and Audio Experiment

by 9elements

http://9elements.com/io/projects/html5/canvas/

http://9elements.com/

in love with HTML5

pavan_sry

採用テクノロジー

<canvas>
- 図の描画

<audio>
- 音楽の再生

Modernizr
- http://www.modernizr.com/

Processing.js
- http://processingjs.org/

Movement tracker by Paul Rouget

http://people.mozilla.com/~prouget/demos/tracker/tracker.xhtml

http://blog.mozbox.org/

Movement tracker

Args:

Threshold (0 -> 255)	230
Work factor	20
Zoom factor	1
Sensibility	0
Display video	☐
Display diff	☑

Matrix:

- (•) Sobel
- ( ) Kirsch
- ( ) Harris

fps

fps: 24

Go !

採用テクノロジー

<video>
    ビデオの再生
<canvas>
    ビデオの表示
Web Workers
    動きを検知する処理

Sketchpad by COLOR JACK

http://mugtug.com/sketchpad/
http://www.colorjack.com/


PAINT
LIGHT
COLOR
GRADIENT
PATTERN
SWATCH
HISTORY
OPTIONS
TOOLS
DRAWING.PNG
009966
H 160
S 100
V 60
R 0
G 153
B 102
A 100
PATTERN
GRADIENT
Brush
DIAMETER 25
HARDNESS 60
FLOW 92
OPACITY 70
HISTORY
original
brush #1
SWATCH
SOLIDS
GRADIENTS
PATTERNS
Squidfingers
BY: TRAVIS

採用テクノロジー

<canvas>

- パレットの表示
- 履歴の記録

Wars Episode IV: A NEW HOPE by Guillermo Esteves

http://www.gesteves.com/experiments/starwars.html
http://www.gesteves.com/


STAR
WARS


A NEW HOPE
It is a period of civil war.
Rebel spaceships, striking
from a hidden base, have won
their first victory against
the evil galactic Empire.
During the battle, Rebel
spies managed to steal secret
plans to the Empire's
ultimate weapon, the DEATH
STAR, an armored space
station with enough power
to destroy an entire planet.
Pursued by the Empire's
inister agents, Princes
ia races home aboard h

採用テクノロジー

- CSS3 Webフォント
  - フォントをサーバーからダウンロード
- CSS3 Transitions
  - 文字の変形
- CSS3 アニメーション
  - 文字の動き

まとめ

open

- ライセンス・フリー
- 仕様の詳細が完全にオープン
- 互換性が保たれたブラウザー環境へ

Web ≠ ホームページ

- ホームページ・プラットフォームからアプリケーション・プラットフォームへシフト
  - Webがホームページからデスクトップ・アプリケーションまでをカバー
  - ブラウザー（デスクトップ）とクラウド（サーバー群）の組み合わせで、デスクトップ・アプリケーションの多くを代替できる。
- Webディベロッパー ≠ ホームページ制作者
  - コンピューター・アプリケーション・ディベロッパー

マルチ･デバイス

- Webはパソコンだけにあるのではない。
- あらゆるコンピューター・デバイスへ。
- モバイル端末、テレビ、カーナビ、ゲームコンソール、セットトップボックス、その他組み込み機器 etc.

FORD In-Dash Computer / Opera

http://www.fordworksolutions.com/Products/In-Dash
http://www.youtube.com/watch?v=D3nHCxokyT0

知識より想像力が問われる時代へ

- HTML5のAPIの仕様はシンプル
  - 誰でも簡単にアプリケーションが作れる。
  - 仕様の詳細は誰でもアクセス可能。
  - 知っていること、覚えていることに、価値はない。
- Webテクノロジーを、どこで、どのように使えるのかを考えることが重要。

JavaScriptがフロントエンドの中核に

- HTML5のAPIは、JavaScriptで。
- プロプライエタリーなプラットフォームの必要性が低下。
- ブラウザー + JavaScriptだけで、多くのコンピューター・アプリケーションをカバー。
- つぶしが効くスキルへ。