

**HTML**



平成22年度OISA技術研究会 HTML5部会

# HTML5の可能性

～進化するWebアプリケーション～



## メンバー

株式会社オーイーシー

渡抜 剛

株式会社オーイーシー

西村 良太

大銀コンピュータサービス(株) 上福 梨彩

九州東芝エンジニアリング(株)

安部 幸城

大分大学修士課程2年

小原 辰徳

大分大学修士課程1年

角野 正和

大分大学修士課程1年

張 勇

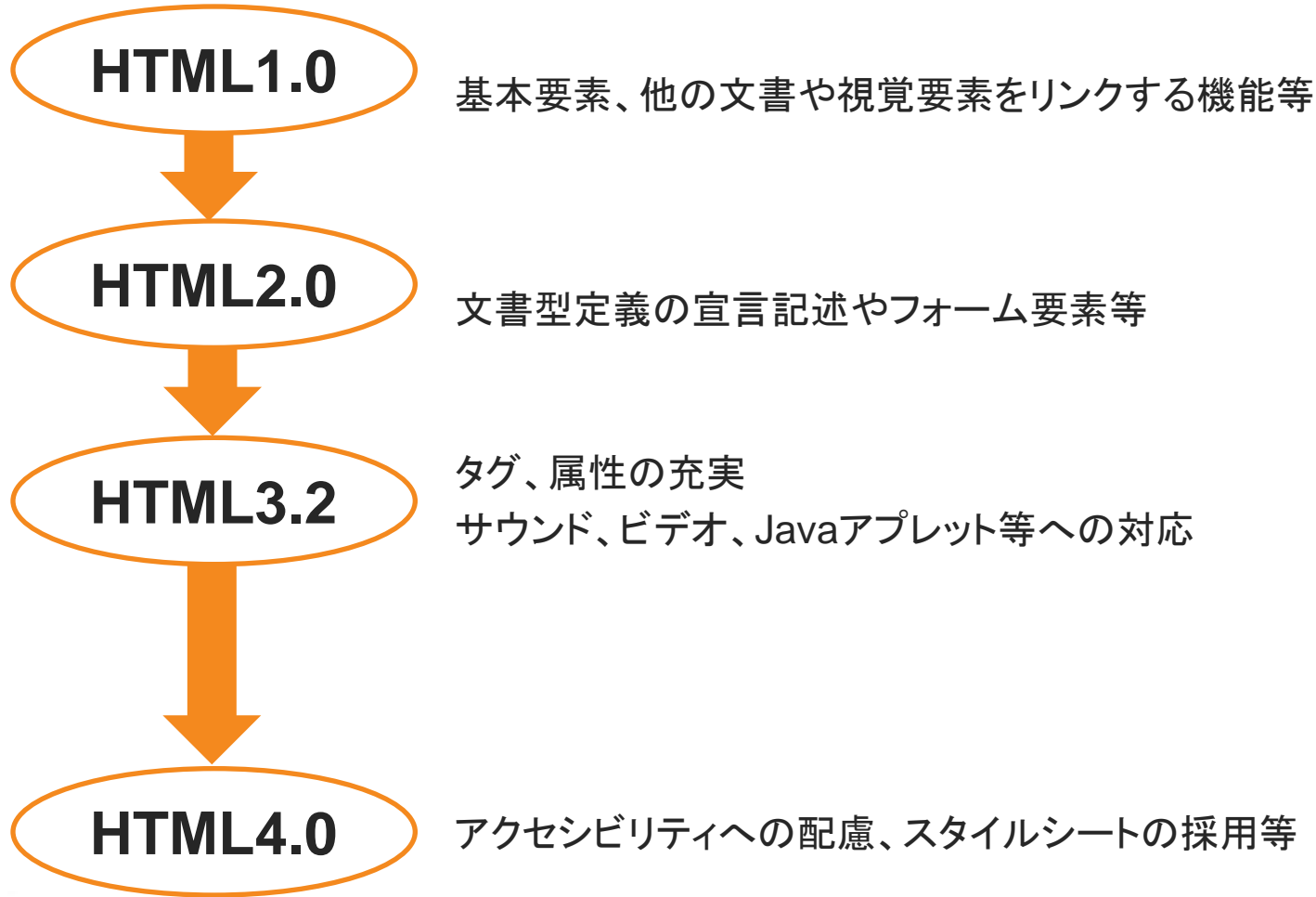


はじめに

# HTML5とは



# HTML5の略歴





# HTML5の背景

HTML4

W3Cにより、XMLへの移行が計画

XHTML

2002年頃

ブログの普及等で使用されたものの、text/htmlとして扱われるシーンが多く、XHTML文書としての意味が薄れた

AJAX

2009年 XHTML2.0標準化を中止

HTML4、XHTMLに対する不満

音楽、動画などの大型コンテンツの一般化  
Web上で表現可能なリッチなコンテンツを求める声

2004年頃

WHATWG

HTML



2008年頃から  
W3Cも協力



# HTML4との相違

Webアプリ用要素



HTML  
4

マルチメディア用要素



従来機能の底上げ及び、新たなAPIの追加





# 新レイアウト要素

HTML5で新たに追加拡充されたレイアウト要素



# 新レイアウト要素

## レイアウト要素とは

ホームページやWebアプリケーションにおいて、ページの構造を明確にするための要素

## レイアウト要素の経緯

初期・・・TABLE(表)タグを利用



HTML4まで・・・DIVタグ + CSSを利用



※ TABLE・DIVは、本来レイアウトのためのタグではない。

HTML5・・・レイアウト要素の追加により、役割に応じたタグが利用可能







# 新レイアウト要素

## HTML5で追加されたレイアウト要素

- header ページの上部に配置し、セクションのヘッダを表す。
- footer ページの最下部に配置し、セクションのフッタを表す。
- section 記事を細分し、コンテンツを識別しやすくする。
- article ページ上のメイン記事コンテンツを扱うためのコンテナの役割
- nav ナビゲーション要素であることを表す。各ページへのリンクを設定する。
- aside サイドバーや関連コンテンツを設定する。
- hgroup 下位の見出し要素をまとめる。
- menu コンテキストメニュー、ツールバー、リストを構成する。

など 全15種類

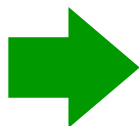
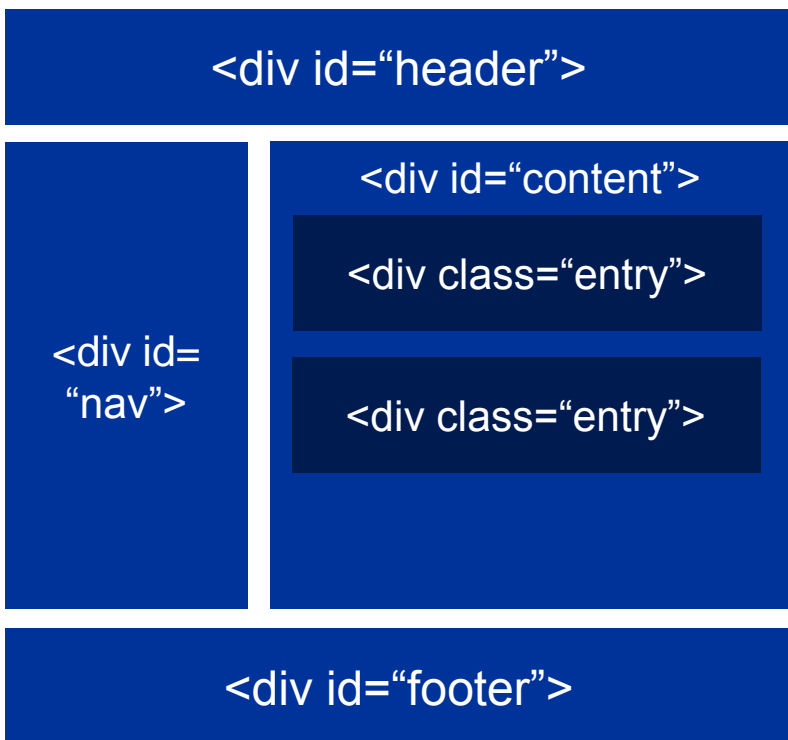


# 新レイアウト要素

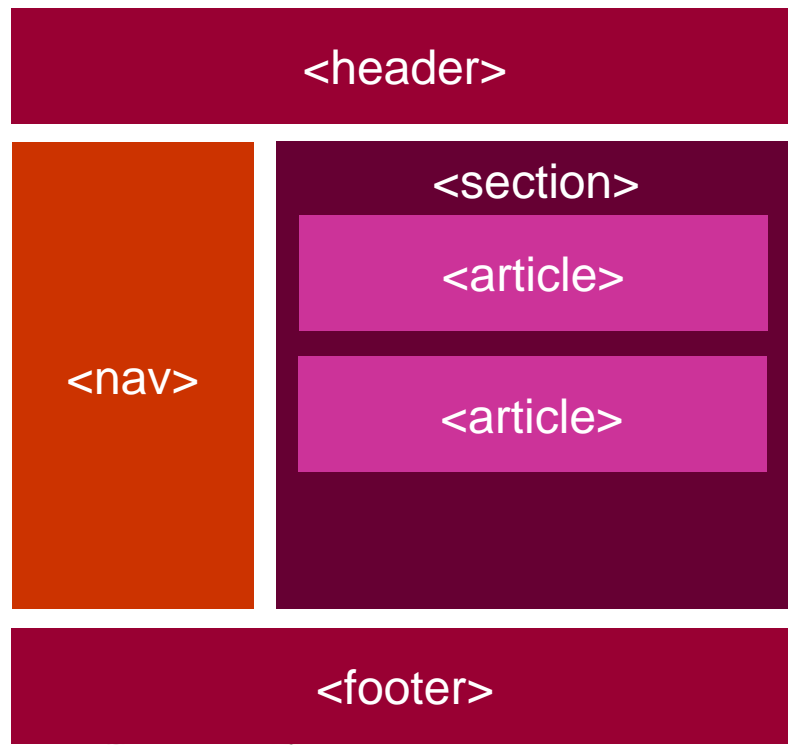
## HTML4とHTML5の比較

### ①ソースコード

#### HTML4



#### HTML





# 新レイアウト要素

## ②画面例





# 新レイアウト要素

HTML4

The screenshot shows a web browser window titled 'html5-layout' with the address bar displaying 'file:///D:/技術研究会HTML5/html5.htm'. The browser's bookmark bar includes Google, Google グループ, Zoho Viewer - オンラ..., Yahoo! JAPAN, and Google Docs - All it... The page content is a blog layout with a yellow header, an orange sidebar, a green main content area, and a pink footer. The header contains the title '〇〇のブログ' and subtitle 'ブログサブタイトル'. The sidebar lists five categories: カテゴリ 1, カテゴリ 2, カテゴリ 3, カテゴリ 4, and カテゴリ 5. The main content area features three sections: '最近の投稿' (Recent Posts), '買い物' (Shopping), and '読書' (Reading), each with a date and description. The footer contains the copyright notice 'Copyright c 2010-2011 XXXXX All Rights Reserved.'.

**HTML4**

```
<div id="header">
<H1>〇〇のブログ</H1>
<H3>ブログサブタイトル</H3>
</div>
```

**HTML5**

```
<header>
<H1>〇〇のブログ</H1>
<H3>ブログサブタイトル</H3>
</header>
```

**HTML4**

```
<div id="footer">
<P>Copyright c 2010-2011
XXXXX All Rights
Reserved.</P>
</div>
```

**HTML5**

```
<footer>
<P>Copyright c 2010-2011
XXXXX All Rights
Reserved.</P>
</footer>
```



# 新レイアウト要素

HTML4

HTML5

```
<div id="menu">
```

```
<H2>カテゴリ</H2>
```

```
<UL>
```

```
<LI>カテゴリ 1</LI>
```

```
</UL>
```

```
</div>
```

```
<nav>
```

```
<H2>カテゴリ</H2>
```

```
<UL>
```

```
<LI>カテゴリ 1</LI>
```

```
</UL>
```

```
</nav>
```

〇〇のブログ

ブログサブタイトル

カテゴリ

- ・ [カテゴリ 1](#)
- ・ [カテゴリ 2](#)
- ・ [カテゴリ 3](#)
- ・ [カテゴリ 4](#)
- ・ [カテゴリ 5](#)

最近の投稿

買い物

今日は買い物に出かけた...

```
<div id="content">
```

```
<H1>最近の投稿</H1>
```

```
<div class="entry">
```

```
<H1>買い物</H1>
```

```
<P>今日は買い物に出かけた...</P>
```

```
</div>
```

```
</div>
```

```
<section>
```

```
<H1>最近の投稿</H1>
```

```
<article>
```

```
<H1>買い物</H1>
```

```
<P>今日は買い物に出かけた...</P>
```

```
</article>
```

```
</section>
```

© 2010-2011 XXXXXX All Right



# 新レイアウト要素

## 新レイアウト要素のメリット

### レイアウト方法の共通化

設定する区画に適切な要素を使用できるため、共通化・標準化が可能

### 文書の構造化（セマンティックWebの実現）

ソースコードから画面上での位置や役割が把握しやすい。

ブラウザや検索エンジンに対して明確に文書構造を伝えることができ、  
検索精度の大幅向上、ホームページのデータベース化が可能。

### アクセシビリティの向上

例えば読み上げソフトを使ってWebページを閲覧する際に、ヘッダやメニュー等を飛ばして、本文の部分だけを読み上げる、といったことも可能。





## 新フォーム要素

HTML5で新たに追加拡充されたフォーム要素



# 新フォーム要素

## 従来のフォーム要素(HTML4)

`<input type="text" value="文字列" />`

文字列

`<input type="password" value="PASSWORD" />`

.....

`<input type="radio" value="radio" />`ラジオボタン

☐ ラジオボタン

`<input type="checkbox" value="check" />`チェックボックス

☐ チェックボックス

文字列の入力⇒「type="text"」1種類

14

年前から変わらず



# 新フォーム要素

## HTML5で追加されたフォーム要素

`<input type="tel" value="090-0000-0000" />`

電話番号入力

`<input type="search" value="HTML5 検索" />`

検索文字入力

`<input type="url" value="http://hoge.com/" />`

URL入力

`<input type="email" value="hoge@hoge.co.jp" />`

Eメール入力

`<input type="datetime" value="2011/12/22 15:00" />`

日時(グローバル)入力

`<input type="date" value="2011/12/22" />`

日付入力

`<input type="month" value="2011/12/1" />`

月入力

`<input type="week" value="2011/12/22" />`

週入力

`<input type="datetime-local" value="2011/12/22" />`

日時(ローカル)入力

`<input type="number" value="99" />`

数値入力

`<input type="range" value="5" />`

スライダーでの数値入力

`<input type="color" value="rgb(255,255,255)" />`

カラーパレット



# 新フォーム要素

コンポーネント化により、入力する情報の種類に合わせて使い分けることが可能となった。

## メリット

- ・入力された情報の信頼性向上
- ・HTML5の文書データ処理の効率化
- ・Webアプリケーションにおいて、入力チェックをブラウザで処理することによる、実装・テスト工程の作業ボリューム軽減



# 新フォーム要素

## 属性の追加

- placeholder - 入力例の表示
- autofocus - 初期フォーカス
- required - 必須入力



# 新フォーム要素のデモ







# 新フォーム要素

## 新フォーム要素を使ってできること

- ・項目入力、入力チェック機能等をHTMLで吸収。  
→ソースコードがコンパクトになり、コーディング、テスト時間を短縮可能。
- ・今後の各ブラウザのサポートが進む。  
→標準化され、クロスブラウザの面からも有効。



## Canvas要素

新たに追加されたCanvasタグ

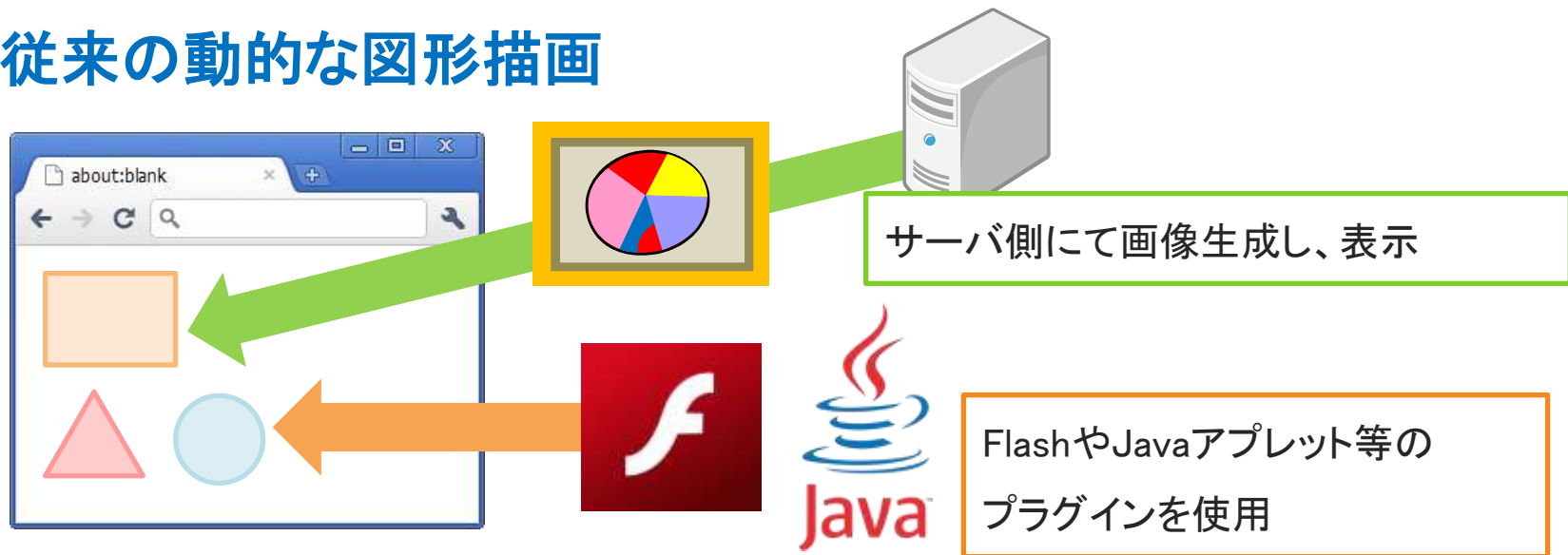


# Canvas要素

## Canvas要素とは

HTML5から新しく導入された、ブラウザ上に図を描くために策定された仕様。

## 従来の動的な図形描画



Canvasは、FlashやJavaのようにプラグインを使わずに、  
JavaScriptベースで図を描くことが可能。



# Canvas要素

## Canvasを使用した図の表現

真っ白なキャンバスに筆を使って絵を描く

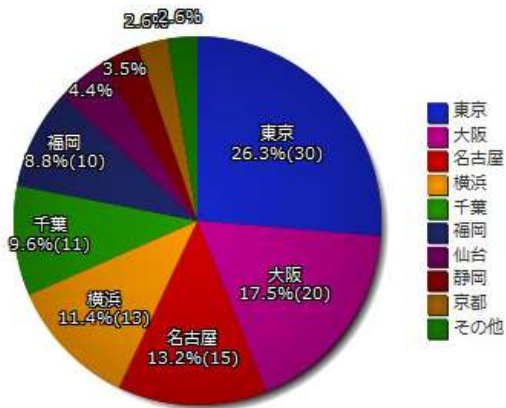
Canvas要素

JavaScript



状況に応じて動的な図を表現する際に有効

例えば、敷居の高いチャート処理が、比較的簡易的に実現可能



直線、矩形、円弧、二次曲線、ベジェ曲線等の描画が可能



## Canvas要素ができないこと

- ・アニメーションのように、描いた図を動かすことは不可能。
- ・埋め込みフォントが使えない。

## まとめ

- ・特別なプラグインを使わずに、動的に図の描画が可能。
- ・複雑な図形描写やアニメーション等、全ての用途を満たせるわけではない。
- ・Flashの代替技術ではないため、状況に応じた選択が必要。
- ・今後のブラウザサポート、フレームワークの拡充が期待される。





# Canvas要素

## 図形描画のサンプル

<body>

```
<canvas id="canvas" width="600"  
height="200"></canvas>
```

```
<script type="text/javascript">  
    var canvas=document.getElementById("canvas")
```

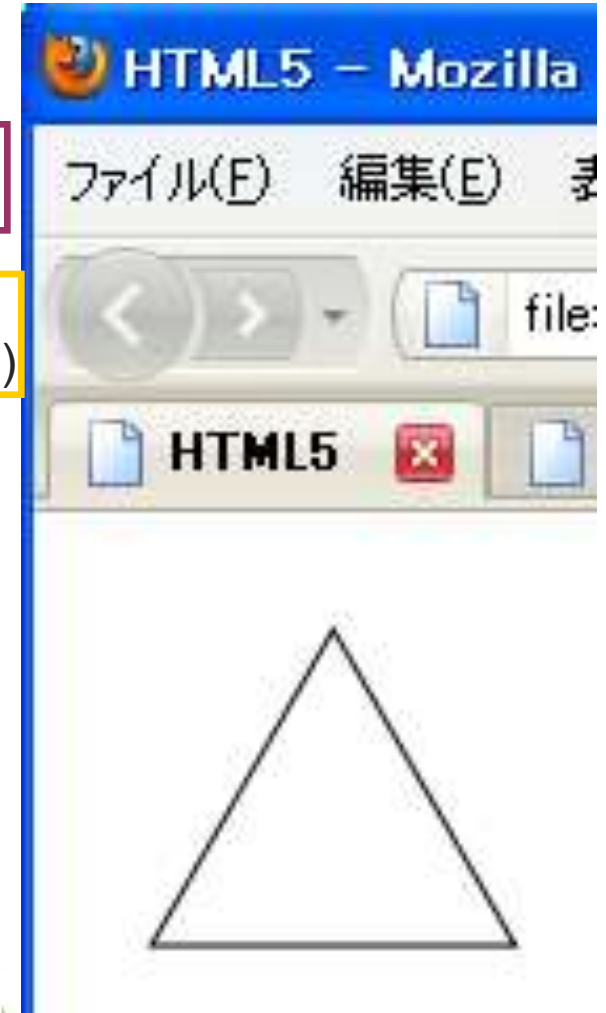
```
    var context=canvas.getContext("2d");
```

```
    context.beginPath();  
    context.moveTo(20,95);  
    context.lineTo(106.602,95);  
    context.lineTo(63.301,20);  
    context.closePath();  
    context.stroke();
```

</script>

</body>

描画要素の準備







# Canvas要素のデモ





## Video/Audio要素

HTML5で新たに追加されたVideo/Audio要素



# Video/Audio要素

## Video/Audio要素とは

HTML5から新しく導入された、ブラウザ上で動画/音声を再生させるために策定された仕様。

## 従来の動画・音声再生方法

FlashやQuickTimeなどのプラグインを利用し、HTMLへの埋込みを行っていた。

```
<object width="500" height="300">  
<param name="allowfullscreen" value="true" />  
<param name="allowscriptaccess" value="always" />  
<param name="movie" value="http://hoge.com/hoge.swf" />  
<embed src="http://hoge.com/hoge.swf" type="application/x-shockwave-flash" allowfullscreen="true"  
allowscriptaccess="always" width="500" height="300"></embed>  
</object>
```

複雑で多岐に渡るパラメータ



# Video/Audio要素

## 使用方法

<video src="動画ファイルのURL" controls="controls"></video>

プラグインなしに、ブラウザのみで動画を再生することが可能。



属性	説明
src	動画/音声データのURL
poster	動画/音声データを利用できない場合に、代わりに使用する画像のURLを指定する(videoのみ)
autobuffer	動画/音声データのバッファリングを自動で開始するかをbooleanで指定する
autoplay	再生を自動で行うかどうかをbooleanで指定する
loop	ループ再生を行うかどうかをbooleanで指定する
controls	ブラウザ固有のコントロールUIを表示するかどうか
width	動画の幅(ピクセル)(videoのみ)
height	動画の高さ(ピクセル)(videoのみ)



# ビデオ要素のデモ

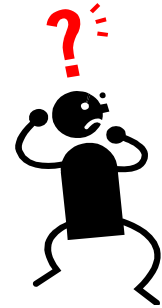




# Video/Audio要素

## 問題点

標準となるコーデックやフォーマットが定まっておらず、  
ブラウザの対応状況もバラバラである。



## ブラウザの対応状況

ブラウザ	H264	OGG	WebM
IE9	○		△
IE6～IE8			
Firefox		○	○
Safari	○		
Chrome		○	○
Opera		○	○
Android	○		
iOS	○		
WP7	○		
Flash	○		△
Silverlight	○		

### H.264陣営

(Microsoft、Apple)

## VS

### WebM陣営

(Google、Mozilla、Opera)





# Video/Audio要素

## Web業界の背景

ブラウザのコーデック対応に伴う問題

Apple iPhoneのFlash非対応による、コンテンツ業界の動き

動画運営サービスである、Vimeoやニコニコ動画が、HTML5動画のサポートを開始。  
YouTubeも試験運用段階であるが、HTML5動画の表示をサポート開始。





# Video/Audio要素

## Video要素とFlashの比較



Flashが優れている点

- ①ビデオストリーミング
- ②カメラ／マイクへのアクセス



HTML5が優れている点

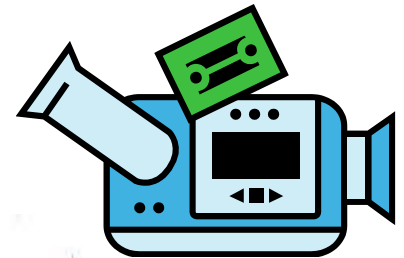
- ①ブラウザのみで再生可能
- ②オープンソース・標準化
- ③業界の流れ



# Video/Audio要素

## Video/Audioの今後

- ・シンプルな実現方法による簡易化、JavaScriptによる表現の幅の拡大  
→身近なコンテンツとして継続してWebへの定着化が促進される。
- ・ユーザーサイドとしては、Flash/HTML5の違いによる影響は少ない。  
(iPhoneユーザーがFlashを再生できない局所的な影響あり)
- ・標準コーデックの行方がそのままVideo/Audioの今後に影響  
(結局Flashがデファクトスタンダードとなる可能性もあり)





## その他の要素

ドラッグ&ドロップ、WebDB、オフライン、Geolocation



その他の機能

# ドラッグ&ドロップ機能



# ドラッグ & ドロップ API

## ドラッグ & ドラッグ API

HTML5から新しく導入された、ブラウザ上でドラッグ&ドロップを実現させるために策定された仕様。

## ドラッグ & ドロップ専用のAPIが利用できる

- ・ドラッグしたいオブジェクトには  
「draggable="true"」を追加するだけでドラッグ可能

- ・シンプルなコードで実現
- ・ブラウザ以外のアプリケーションとテキストやファイルのやり取りが可能



# ドラッグ & ドロップ API

## ドラッグ & ドラッグ API のイベント

イベント名 (event.type)	イベント発生対象 (event.target)	イベント発生のタイミング
dragstart	ドラッグ元	ドラッグ開始時
drag	ドラッグ元	ドラッグ中
dragend	ドラッグ元	ドラッグ終了時
dragenter	ドロップ先	ドラッグオーバー開始時
dragover	ドロップ先	ドラッグオーバー中
dragleave	ドロップ先	ドラッグオーバー終了時
drop	ドロップ先	ドロップ時



その他の機能

# オフライン機能



# オフライン機能

## 従来のWebアプリケーション


使えるのはインターネットに接続している間だけ。  
オフライン時には全く使えない。

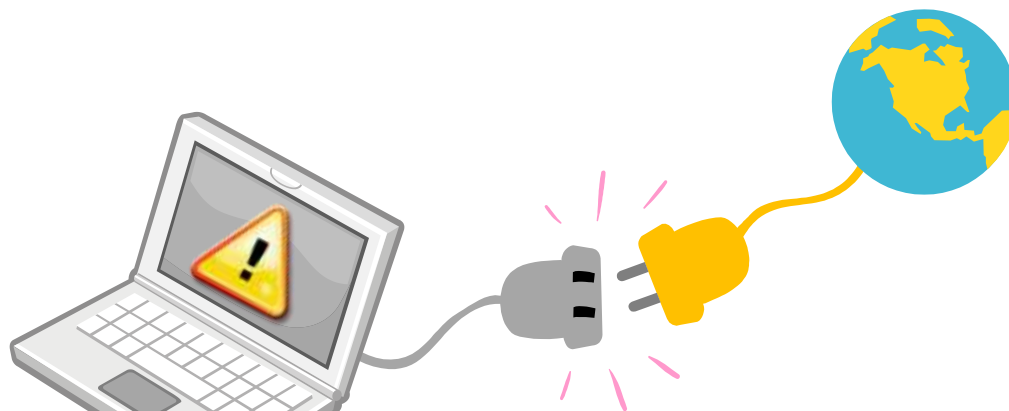


ページが見つかりません

検索中のページは、削除された、名前が変更された、または現在利用できない可能性があります。

次のことを試してください：

- アドレス バーにページ アドレスを入力した場合は、ページ アドレスを正しく入力したかどうかを確認してください。
- ホーム ページを開いてから、表示する情報へのリンクを探してください。
- 別のリンク先を表示するには、[戻る](#) ボタンをクリックしてください。
-  [検索](#) ボタンをクリックして、インターネット上の情報を検索してください。

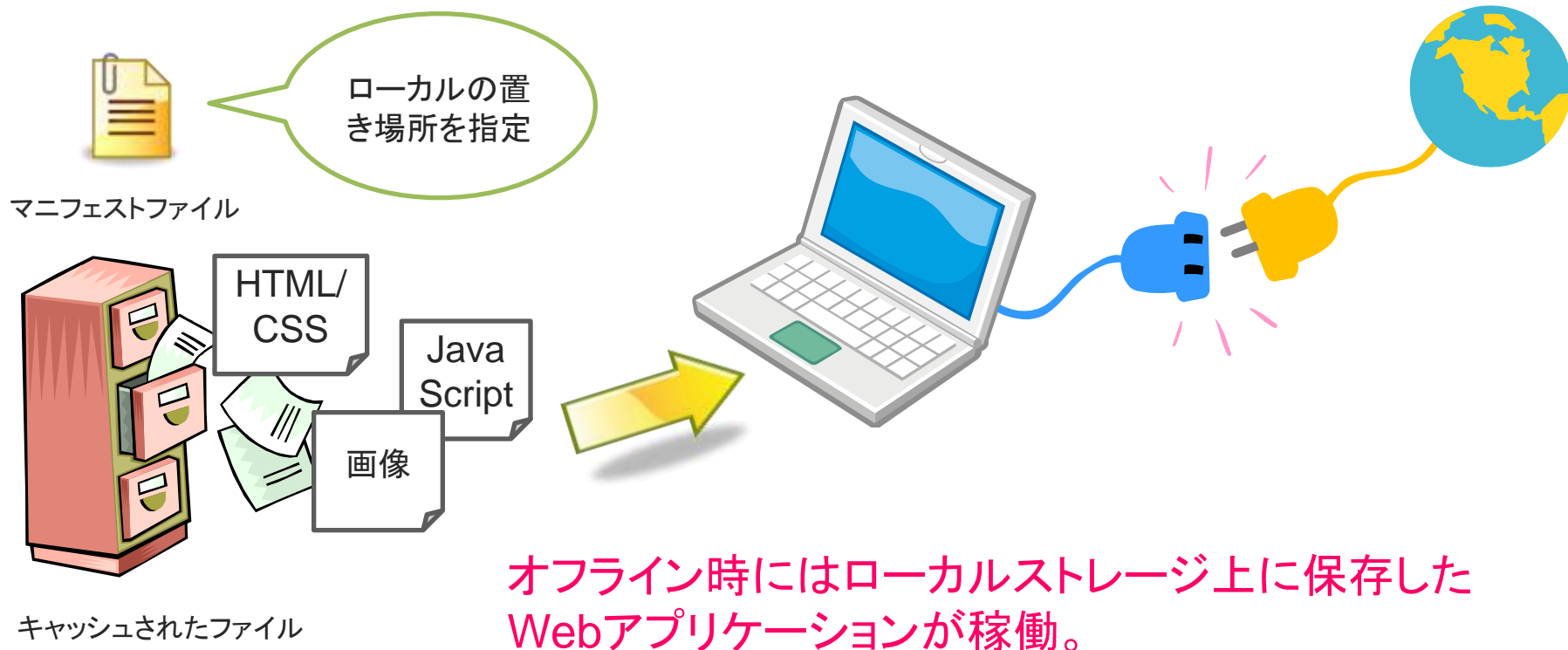




# オフライン機能

## HTML5でのWebアプリケーション

アプリケーションキャッシュにより、オフライン時にも使用可能。





その他の機能

# ローカルストレージ

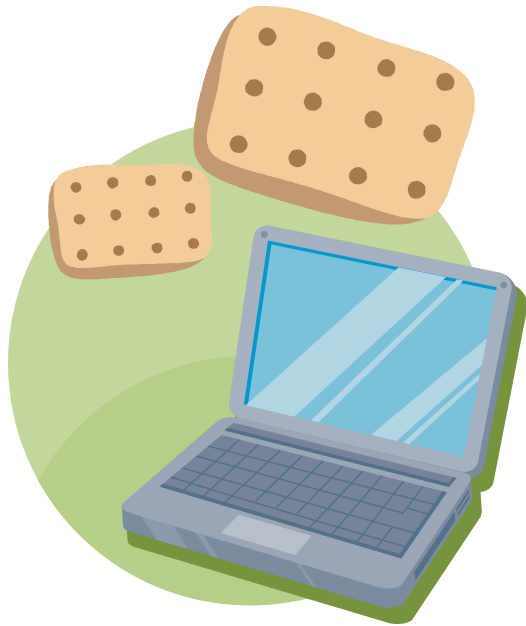


# ローカルストレージ

## これまでのクライアント側のデータ保存

# Cookie

ログイン時のユーザー名など、かんたんな情報をWebブラウザに保存するための仕組み。



### Cookieの制限

- ・Cookie1つ当たりの容量は4,096バイト
- ・Cookieの個数は全部で300個
- ・1つのWebサーバーにつきCookieは20個

→ 本格的なデータ保存に使うには不十分。





# ローカルストレージ

シンプル



高機能

- Web Storage

キーバリュ型でシンプルな仕組みを提供。

従来のCookieに近い。

主流のWebブラウザの最新版でサポートされつつある。

- Indexed Database

キーバリュ型だが、検索等の高度な機能も備えている。

複雑な処理が必要だが、Web SQL DBを使うほどでもない場合に好都合。

- Web SQL Database

リレーショナルなデータベース構築により、複雑なデータ処理が可能。



その他の機能

# Geolocation API



# Geolocation API

## これまでの位置情報取得

携帯電話等からの使用は一般的になりつつあったが、GPSを内蔵していないPC等のデバイスから、現在地を正確に取得することはできなかった。

## Geolocation API

機器の現在地情報を、GPSに加えWi-Fi、IPアドレスといったネットワーク情報から取得することにより、具体的な取得技術を意識することなく取得可能。





# Geolocation API

「モバイルでもPCでも同じAPIが利用できる」



## 利用形態の標準化

全てのWebアプリケーションが、利用者の位置情報を簡単なAPIで取得可能。

位置情報の活用については、現在さまざまなアイデアが実験されつつある。



Foursquare



Gowalla



Google Latitude



RunKeeper

中小規模ビジネスのアウトソーシングを主に手掛けるFreelancer.comに登録されたデータで比較すると、2010年の1Qから2Qにかけて、位置情報関連の仕事は909%増えたとの統計がある。



# デモアプリケーション

## 勤怠管理システム



- ・日々の作業及び工数の入力・管理
- ・作業の内訳をグラフ表示
- ・オフライン利用可  
(社外作業の社員向け)

### 使用したHTML5の機能

新レイアウト要素、新フォーム要素、Canvas要素、  
オフライン機能、ローカルストレージ



HTML5とは

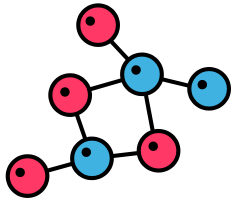
# まとめ





# まとめ

## HTML5の3つの特徴



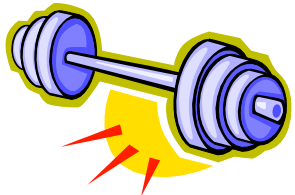
### 構造化

情報の意味の明確化を促進し、セマンティックな構造化言語へと進化



### 互換性

これまでのHTMLの課題であった、Webブラウザ間の互換性問題  
解消の工夫



### 機能強化

これまでの機能不足を解消するフォーム機能の大幅な強化や、  
新しいAPIの追加



# まとめ

## HTML5を取り巻く背景

### 1. スマートフォン・タブレットの台頭



### 2. Apple社の方針



RIAを実現する手段の筆頭として注目



# まとめ

HTML

CSS

Java  
Script

「どんなUIでも、HTMLで実現するのが当たり前」

ウェブ標準のテクノロジーが、  
あらゆるアプリケーションのベースとなる可能性



PC



スマートフォン



TV

# HTML



HTML5は一過性の流行ではありません。  
標準の技術は朽ちることがなく、  
HTML5も例外ではないからです。

**HTML5を使って、進化するWebアプリケーションの可能性に触れてみませんか？**



# HTML



ご静聴ありがとうございました。







# 参考文献

@IT - Webの3つの問題を解決する「HTML5」とは何なのか

<http://www.atmarkit.co.jp/fwcr/design/benkyo/html5appli01/>

HTML5サンプル集

<http://ayuta.co.jp/html5-samples/>

マイコミジャーナル - 詳解!HTML 5と関連APIの最新動向

<http://journal.mycom.co.jp/special/2009/html5-2/001.html>

Nikkei BP Net - Web利用はオフラインに拡大へ, 企業アプリも変わる?

<http://itpro.nikkeibp.co.jp/article/COLUMN/20080626/309579/>

アークウェブシステム開発SandBox - HTML5 3Days Tech Talk:HTML5に搭載されるAPIの勉強会

<http://www.ark-web.jp/sandbox/wiki/4859.html>

Tech Crunch - 位置情報、クラウド、HTML5などの人気が急上昇

<http://jp.techcrunch.com/archives/20100706freelancer-geolocation-html5-jobs/>

