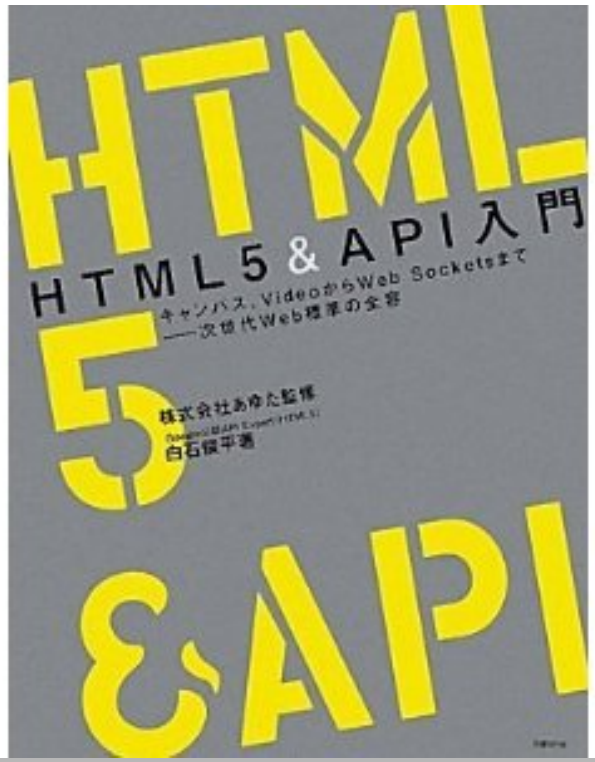
HTML5で変わるWebの世界

2010/12/2 白石俊平

自己紹介

- 白石俊平と申します。
- HTML5開発者コミュニティ、html5-developers-jp管理人
- Google API Expert (HTML5)
- Twitter: @Shumpei
- 著書：HTML5&API入門



本日の流れ

- HTML5の基礎知識
- HTML5の3つの意義
- HTML5の可能性を表すデモアプリたち

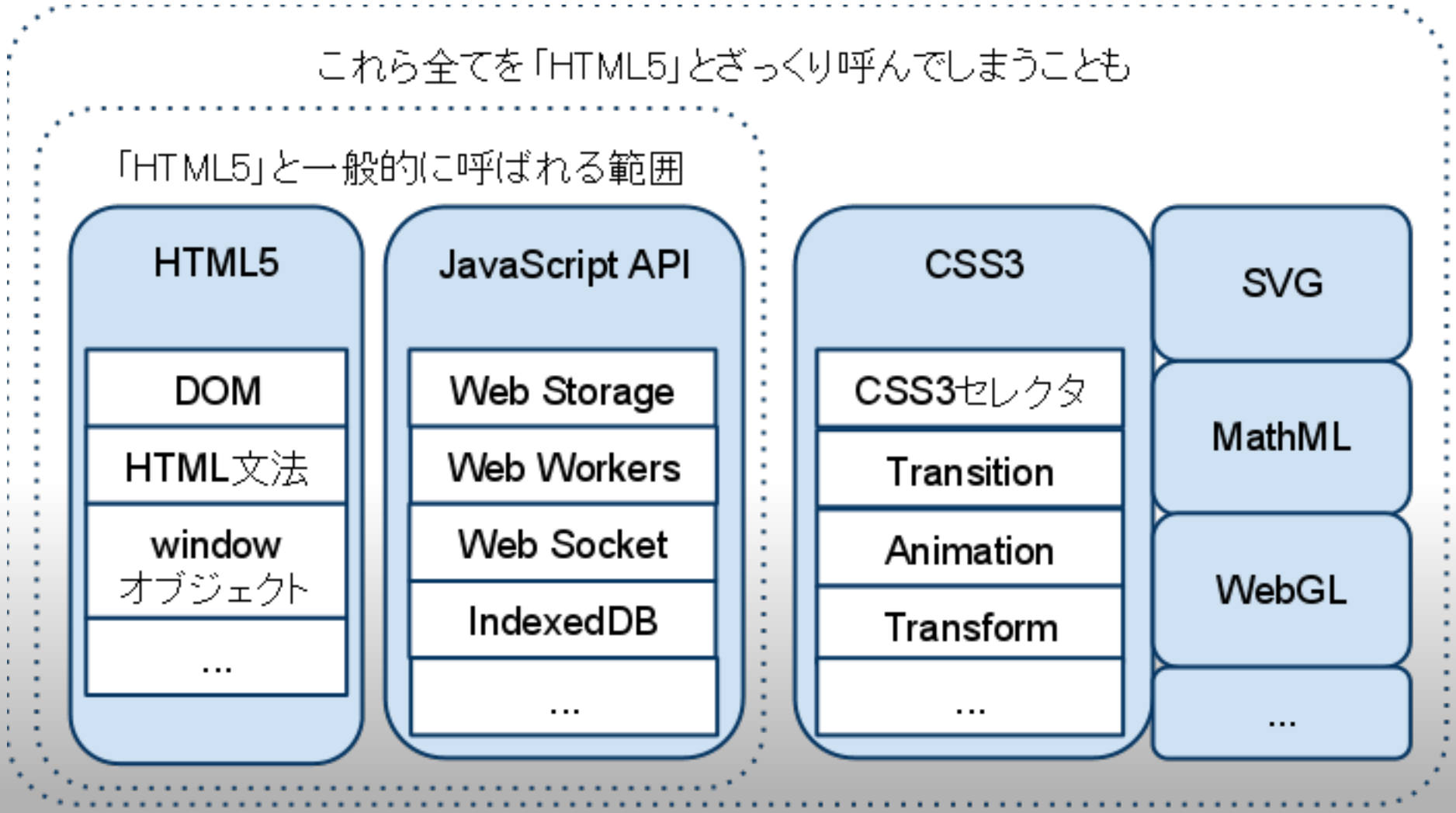
HTML5の基礎知識

HTML5って、なんだろう？

- HTML（Hyper Text Markup Language）の最新バージョン！
- W3C（World Wide Web Consortium）で標準化作業中
  - 2011/5/22に仕様が確定する（Last Call）予定
- Webブラウザによる実装も着々と進行中

どこまでがHTML5か？

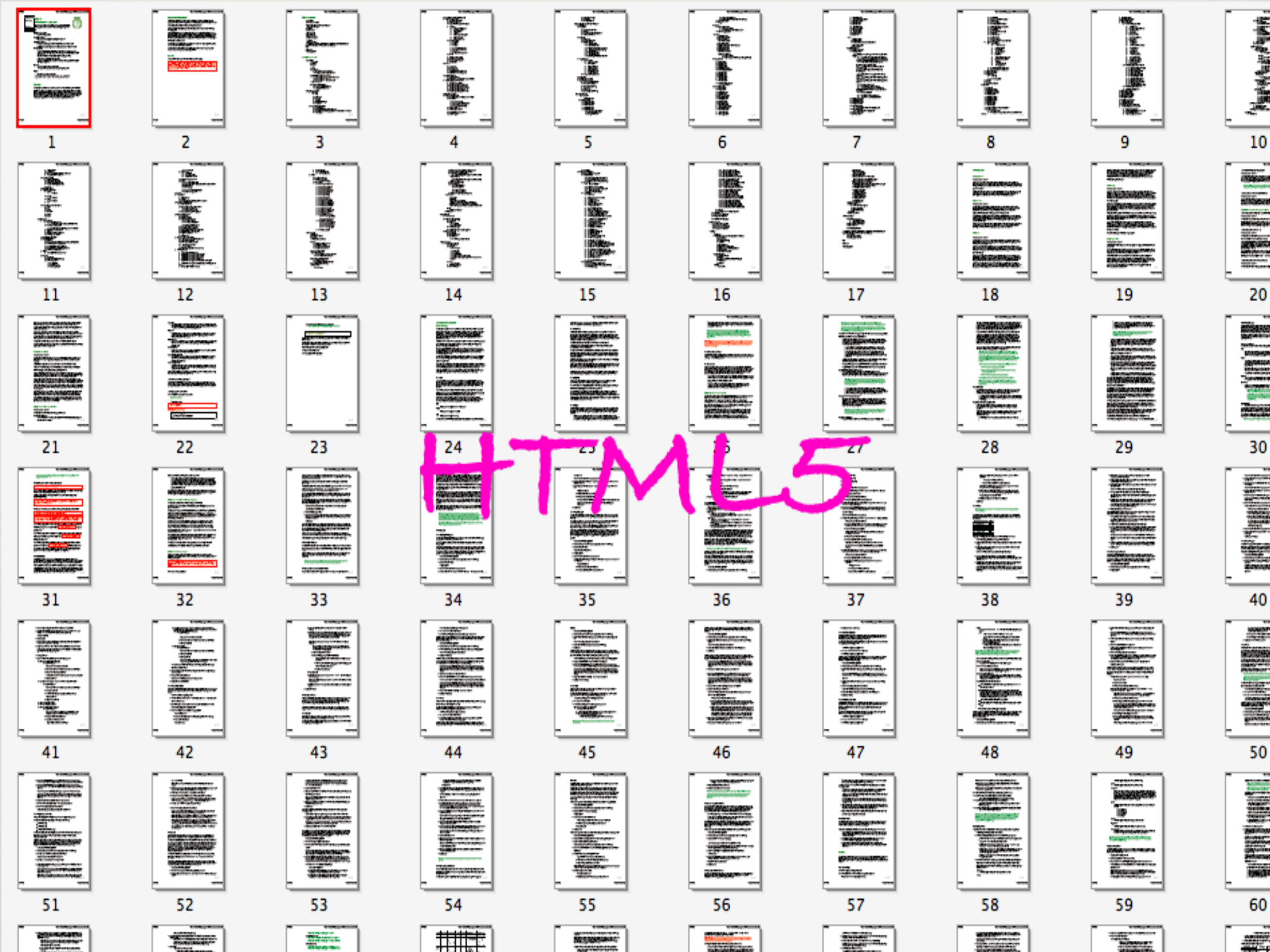
- 「HTML5」と呼ばれている技術は、実際には様々なプログラミング環境を総称したもの。

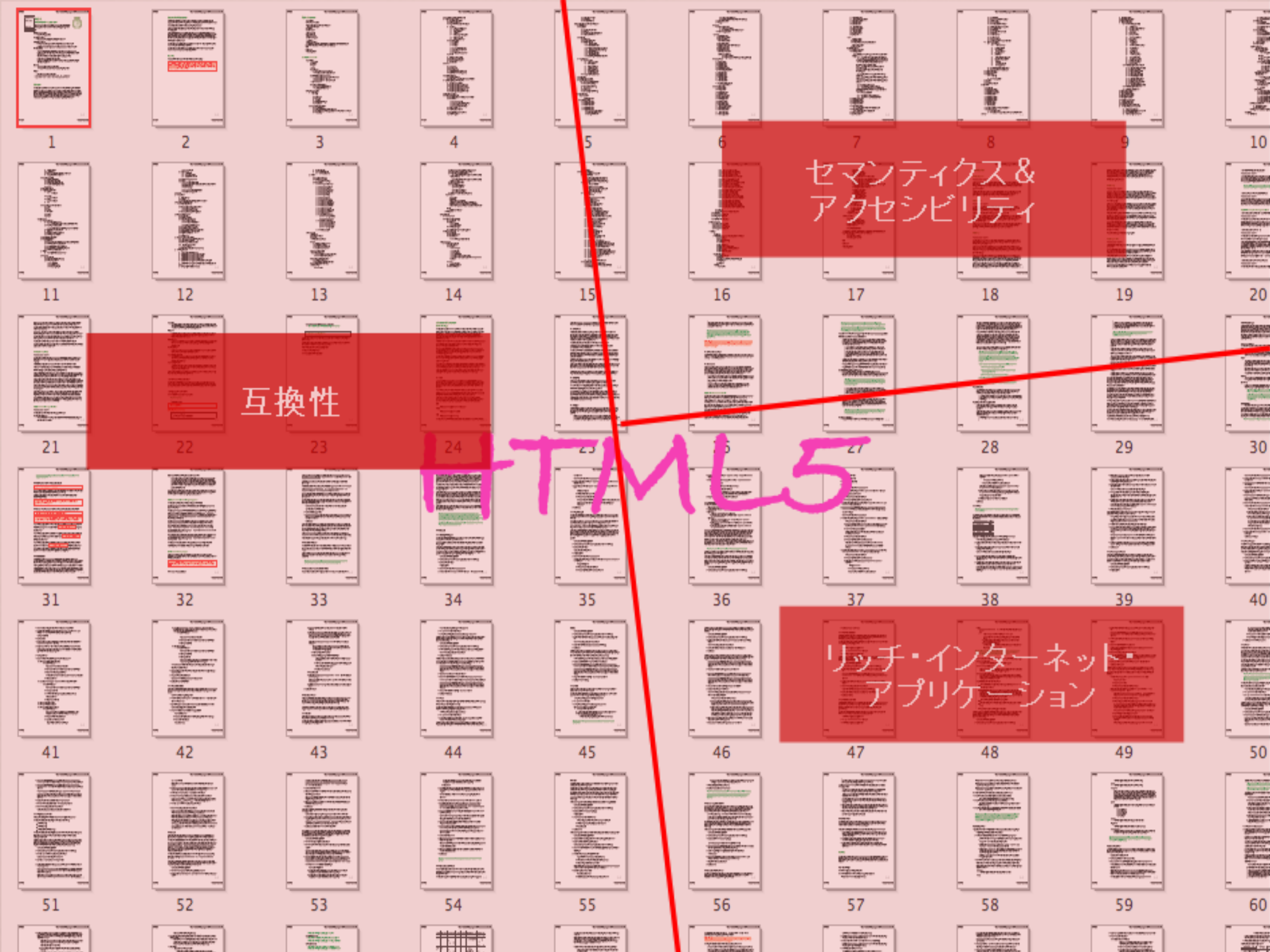


HTML5の3つの意義

HTML5は膨大で、テーマを絞り込むのは難しい。

それでもあえて分けるならば・・・


HTML5


セマンティクス＆
アクセシビリティ
互換性
HTML5
リッチ・インターネット・
アプリケーション

HTML5の3つの意義

- 白石なりに分けると以下の3つが大きな意義と感じられる。
  - セマンティクスとアクセシビリティ
  - 互換性の追求
  - リッチ・インターネット・アプリケーション

セマンティクスとアクセシビリティ

More Readable for

Everyone

セマンティクスとアクセシビリティ

More Readable for

Everyone

健常者、障害者、
プログラムも含めて！

セマンティクスとアクセシビリティ

- セクション要素（section/article/aside/nav...）
- その他の新要素（header/footer/time/command...）
- 既存要素の意味も変化（strong/small/b/i/address/menu...）
- マイクロデータとの統合
- WAI-ARIAとの統合

セマンティックな要素を利用する

```
<div class=”header”>
  <h1>Site Title</h1>
  <div class=”site-nav”>
    <ul><li>Home</li><li>Profile</li><li>Settings</li></ul>
  </div>
</div>
<div class=”content”>
  <div class=”article”>Main Content
    <div class=”date”>2009/11/23に投稿</div>
    <div class=”address”>
      連絡は<a href=”mailto:a@a.jp”>こちら</a></div>
  </div>
</div>
<div class=”footer”>
  <div class=”copyright”>Copyright ...</div>
</div>
```

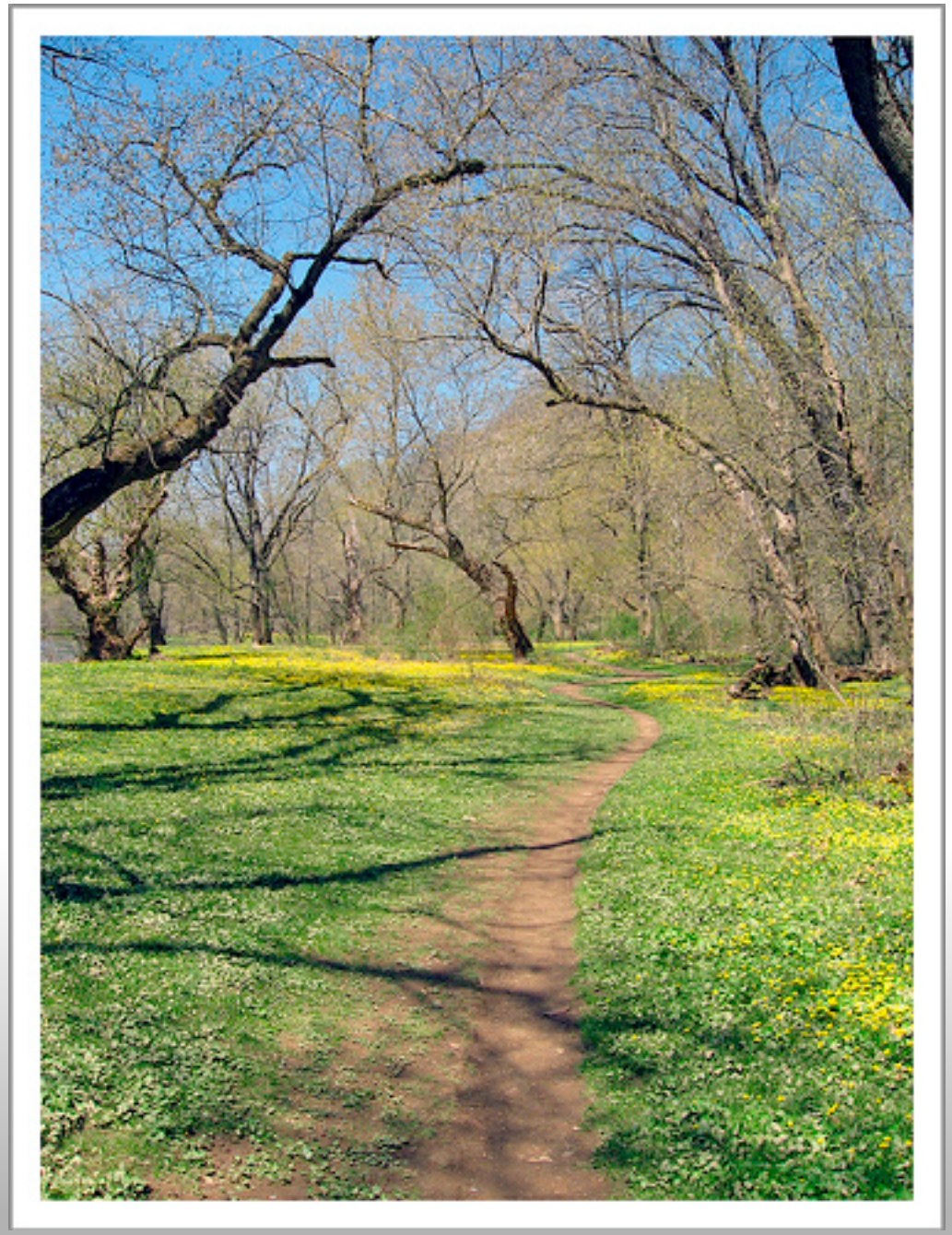
セマンティックな要素を利用する

```
<header>
  <h1>Site Title</h1>
  <nav>
    <ul><li>Home</li><li>Profile</li><li>Settings</li></ul>
  </nav>
</header>
<div class=”content”>
  <article>Main Content
    <time datetime=”2009-11-23”>2009/11/23に投稿</time>
     <div>連絡は<address>
       <a href=” mailto:a@a.jp”>こちら</a></address></div>
  </article>
</div>
<footer>
  <small class=”copyright”>Copyright ...</small>
</footer>
```

互換性の追求

- 現在のWebの大きな問題点が、ブラウザによって挙動が異なること
  - 仕様があいまい
  - 仕様が存在しない
- HTML5はこの問題に正面から取り組んだ。

"Pave the Cowpaths"



Noel Sarah Dietrich@Flickr

"一度書けば、どのブラウザでも動く”

・・・そんな理想的なWebを目指して

リッチ・インターネット・アプリケーション

- HTML5は、「アプリケーションプラットフォーム」を目指す。
- 「Webアプリにできないこと」をどんどん減らしていく

HTML5 & APIが可能にすること

- リッチな入力フォーム
- 2D & 3Dグラフィック
- 動画・音声の再生・生成
- オフラインWebアプリケーション
- クライアントサイドストレージ
- バックグラウンド処理
- サーバプッシュ・双方向通信
- クロスドメイン通信
- ドラッグ &ドロップ
- ファイルの読み書き
- デバイス固有の機能へのアクセス

強化された入力フォーム

- HTML5では、入力フォームが大きく強化された。
  - 入力タイプが増えた
  - 新たな要素が追加された
  - 入力値のチェックを手軽に行える
  - 送信するフォームをボタンごとに切り替えられる
  - その他

追加された入力タイプ

以下のようなタイプが追加された。

date	datetime	datetime-local
month	week	time
number	range	email
url	search	tel
color		

新たに追加されたフォーム関連要素

- progress･･･タスクの進捗状況を表す要素
- meter･･･メーターを表す要素
- keygen･･･公開鍵を生成するための新たな
  フォーム要素
- output･･･画面に出力され、サーバにも値が
  送信されるフォーム要素。
- datalist･･･データのリストを定義するための不可視の
  要素。optionタグを用いて個々のデータを指定する

入力値のチェック

- 以下のような属性を用いて、簡単に入力値をチェック出来る。
  - require・・・入力必須
  - pattern・・・正規表現で入力パターンをチェック
  - step・・・値の刻み
  - max/min・・・最大値/最小値
- 入力値が不正な場合、「:invalid」擬似クラスが適用される

2D & 3Dグラフィック

HTML5 <canvas> element and API,SVG

HTML5 <canvas> element and API

- □描画範囲を表す<canvas>要素と、そこに描画するコンテキストの組み合わせ

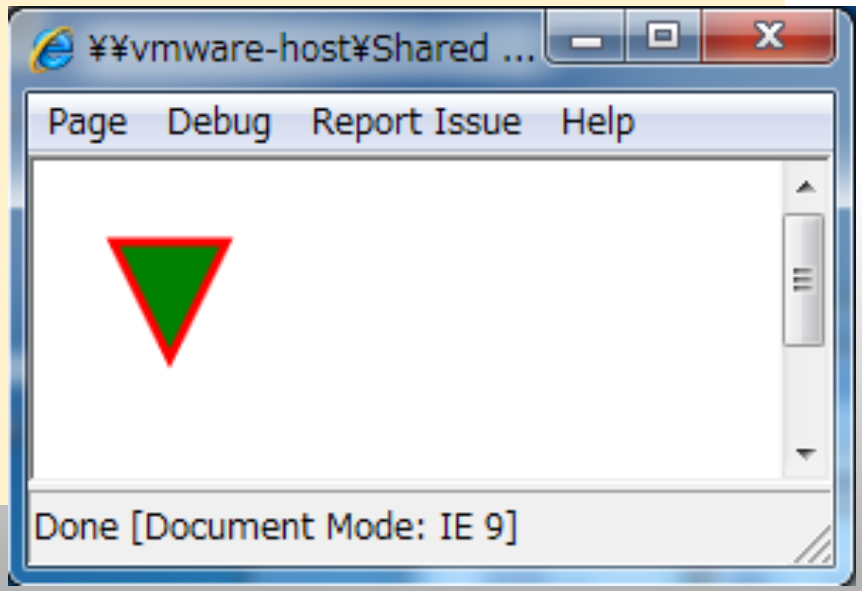
```
var canvas = document.getElementsById("c1");
var context = canvas.getContext("2d");
context.drawImage(...)
```

- コンテキストは現在2種類
  - Canvas 2D Context ・・・2次元グラフィック用。W3Cで標準化作業中（コンテキストIDは″2d″）
  - WebGL ・・・3次元グラフィック用。Khronosで標準化作業中（コンテキストIDは″webgl″）

SVG（Scalable Vector Graphics）

- XMLによるベクタグラフィック形式
- DOMとしての操作も可能

```
<!DOCTYPE html>
<html>
  <svg>
    <path
      d="M 20 20 L 60 20 L 40 60 z"
      fill="green"
      stroke="red"
      stroke-width="3" />
  </svg>
</html>
```



動画・音声の再生

HTML5 <video>/<audio> element and API

HTML5 <video>/<audio> element and API

- □<video>要素で動画の再生、<audio>要素で音声の再生
- src属性、もしくは<source>要素を使用して、リソースのURLを指定する。

```
<video src="sample.ogv"></video>
<video>
  <source src="sample.ogv">
</video>
```

- □JavaScriptによるAPI呼び出し、DOMとしての操作、CSSによる視覚効果との組み合わせも全部OK

モダンブラウザによるビデオフォーマットのサポート状況

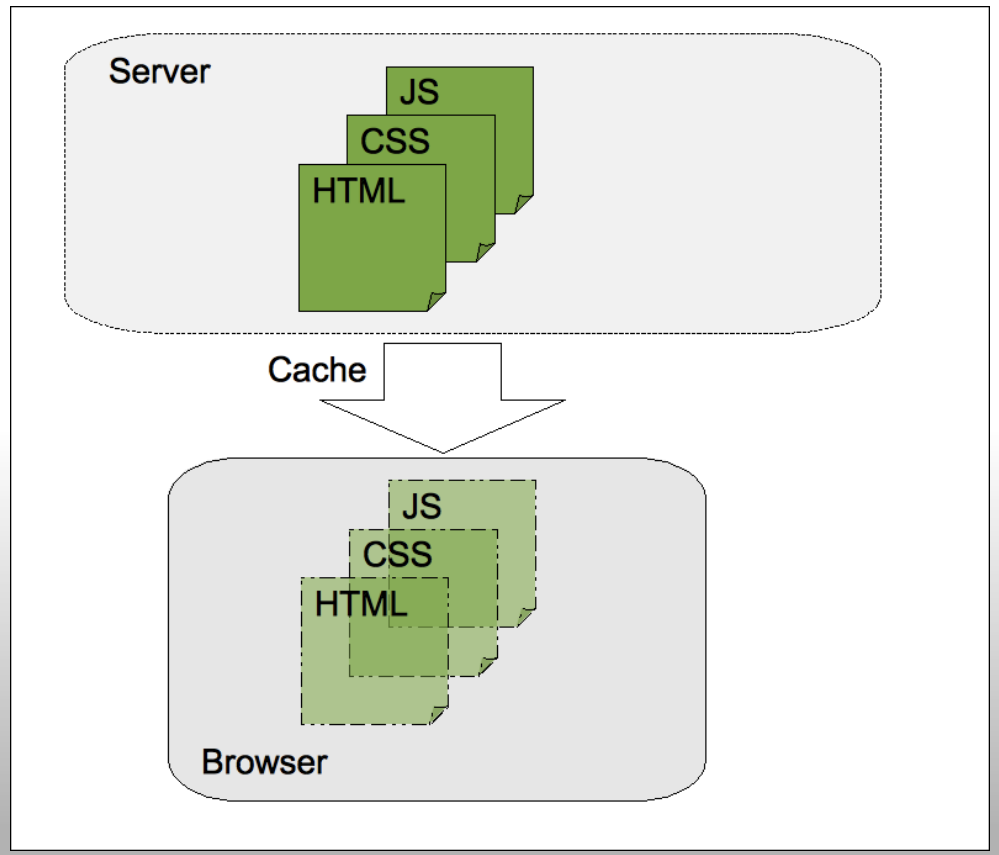
	WebM	Ogg/Theora	mp4/h.264
IE9	●（有効にすれば）		●
Firefox4	●	●	
Chrome7	●	●	●
Safari5			●
Opera10	●	●	

オフラインWebアプリケーション

HTML5 <html manifest=attribute> and ApplicationCache

オフラインWebアプリケーション

- オフラインでもWebアプリが動作する！！
- HTML/CSS/JavaScript/画像などの、Webアプリが必要とするリソースを全てローカルにキャッシュすることで実現



オフラインWebアプリケーション

- キャッシュするリソースを書き連ねた「キャッシュマニフェスト」と言うファイルを用意し、html要素のmanifest属性に指定するだけで実現可能。

hello.manifest

```
CACHE MANIFEST

hello.html
hello.js
```

hello.html

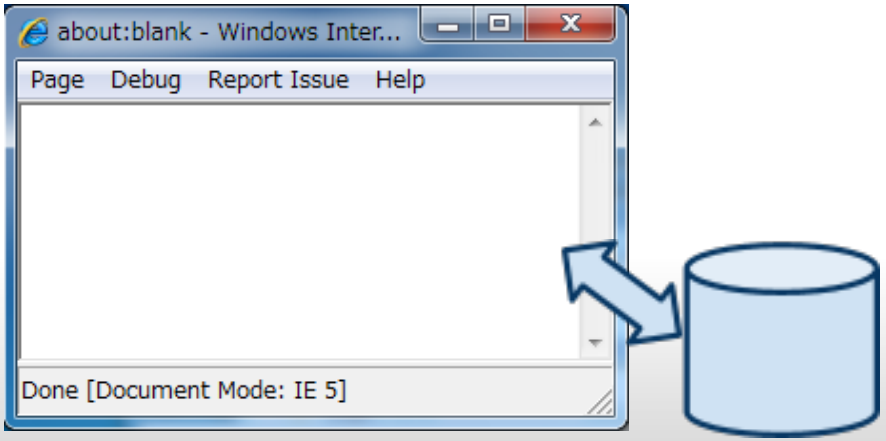
```
<!DOCTYPE html>
<html manifest="hello.manifest">
...
</html>
```

クライアントサイドストレージ

Web Storage, Web SQL Database, Indexed Database API

Web Storage（ローカルストレージ）

- 以下のような特徴を持つ、キー・値型のストレージ
  - サイズ制限なし（仕様上）
  - 永続期間無制限（仕様上）
- JavaScriptの仕様ともマッチしていて、異常に簡単に扱える。



- ウィンドウと同じ生存期間・スコープを持つ「セッションストレージ」というストレージもある

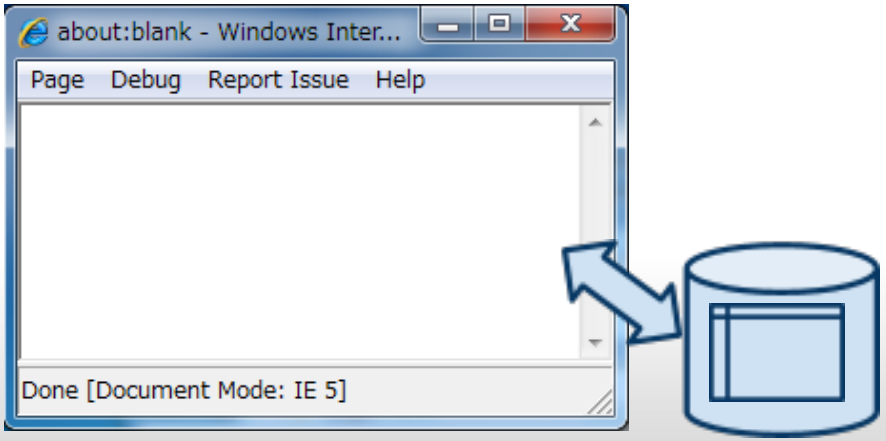
WebStorageのサンプルコード

```
// ストレージへの保存
localStorage.counter = 1;
// ストレージからの読み出し
alert(localStorage.counter);
// ストレージからの削除
delete localStorage.counter;

// JSON文字列にして保存
localStorage.user = JSON.stringify({
  name: "白石", age: 31
});
// JSONから値を復元
var obj = JSON.parse(localStorage.user);
```

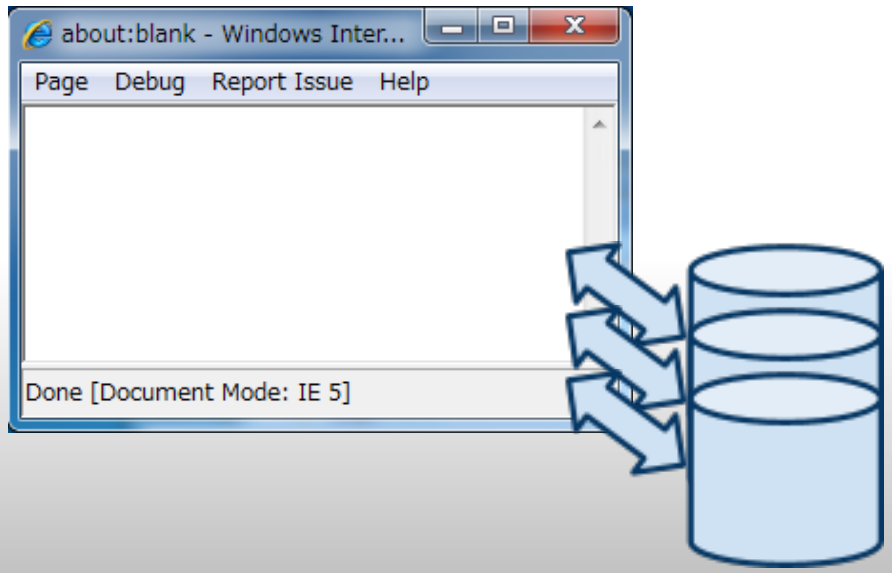
Web SQL Database

- ブラウザに組み込まれたリレーショナル・データベース
- 現在は仕様策定が停止している
- が、SafariとChromeでは実装済み



Indexed Database API

- 組み込みのキー・バリュー・ストア。
- Web Storageよりも複雑だが高機能
  - オブジェクトのカテゴライズやカーソル操作が可能
- Firefox4で部分的にじっ

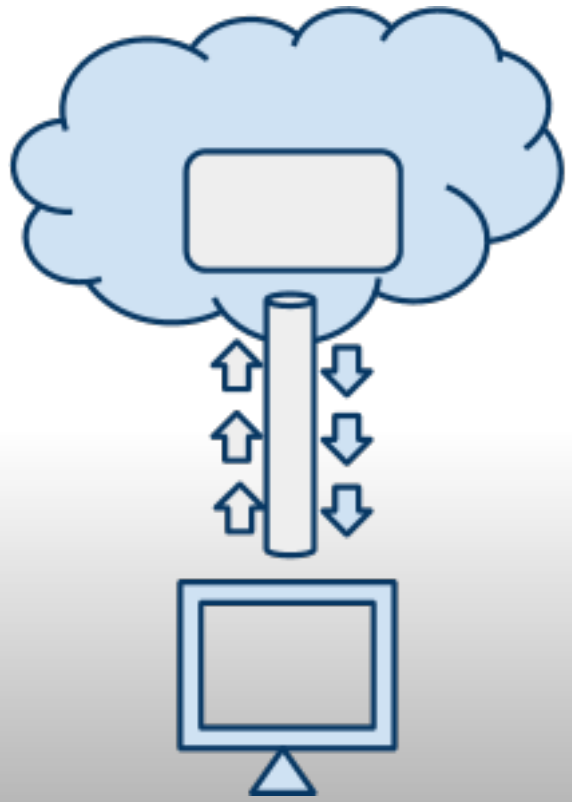


サーバプッシュ・双方向通信

WebSocket

Web Sockets

- 「ファイアウォールを超えられるソケット通信」
- HTTP（Web）との相性がよい
- 双方向
- 少ない通信コスト



WebSocketのサンプルコード

```
// ソケットの作成
var ws = new WebSocket("ws://localhost/echo");
// ソケットからデータを読みだす
ws.onmessage = function(event) {
  alert(event.data);
};
// ソケットにデータを書き込む
ws.send("Hello");
```

HTML5の可能性を表すデモアプリたち

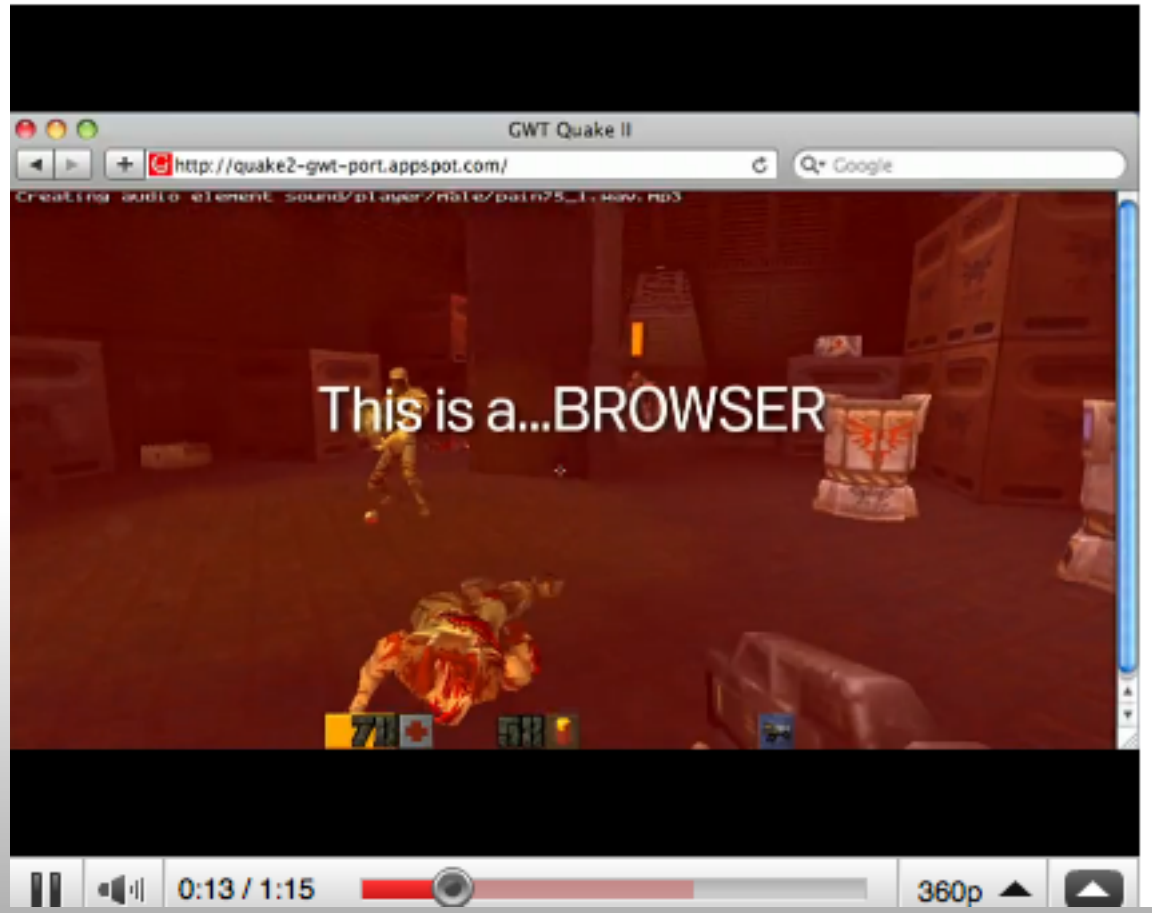
Blowing apart fragments of Video

- 使用しているHTML5要素
  - Video
  - Canvas

1. video要素とcanvas要素2つ（作業用と表示用）を準備し、タイルに分割
2. 一定時間ごとに、不可視のvideoからcanvasに画像を表示
3. マウスクリックされたら、タイルの座標を再計算してcanvasを再表示

Quake Ⅱ GWT Port

- 使用しているHTML5要素
  - WebGL
  - Video/Audio
  - WebSockets
  - LocalStorage



ご清聴ありがとうございました

Contact:

- twitter: @Shumpei
- blog: http://d.hatena.co.jp/Syunpei
- mailto: shumpei.shiraishi [at] gmail.com
- mailto: shumpei.shiraishi [at] openweb.co.jp