

HTML5について



佐野 尚之





本ドキュメントのライセンスについて

この作品は、クリエイティブ・コモンズのAttribution 3.0 Unportedライセンスの下でライセンスされています。

この使用許諾条件を見るには、<http://creativecommons.org/licenses/by/3.0/>をチェックするか、クリエイティブ・コモンズに郵便にてお問い合わせください。

住所は：171 Second Street, Suite 300, San Francisco, California 94105, USA です。



原作者のクレジット（氏名、作品タイトルとURL）を表示することを守れば、改変はもちろん、営利目的での二次利用も許可される最も自由度の高いCCライセンス。

<http://creativecommons.jp/>





変更履歴

■ 第2版

2013/5/14

最新ブラウザ(IE10,Firefox 20.0.1,Opera 12.15,Chrome 26.0.1410.64 m)で動作確認。

Windows版の提供が終了したSafariを削除。

■ 第1版

2012/7/14

オープンソースの「LibreOffice 3.5.4」を使用して作成



目次

- ・ 動作確認環境 5
- ・ HTML5について 6
- ・ HTML5で変更になった書き方 7
- ・ 2D / 3Dグラフィックス – canvas要素 8
- ・ フォーム要素 11
- ・ ドラッグ&ドロップ 16
- ・ Geolocation API 17
- ・ オフラインWebアプリケーション 18
- ・ クライアントにデータ保存 – Web Storage API 19
- ・ 書籍(一部) 20
- ・ 参考情報 21





動作確認環境

■動作確認環境

Acer Aspire 1410

Intel Celeron processor SU2300(1.2GHz, 800MHz FSB)。8GBメモリに変更。HDD250GB。

Windows 7 Home Premium(64bit版) Service Pack1

■ブラウザ

- Firefox 20.0.1
- Internet Explorer 10
- Opera 12.15
- Google Chrome 26.0.1410.64 m



HTML5について

- Web技術の標準化団体であるW3C(World Wide Web Consortium)が管理している標準化された仕様。
- 2014年までの正式勧告を目指して策定されている。
- 従来のHTML(4.01は1998年)の問題点を解決。
 - Webブラウザ間の互換性が向上した。
 - 文書内に埋め込まれた「意味」が明確になった。
 - フォームの入力補助やチェック機能を属性で指定できる。
 - 動画や音声データをHTMLからシンプルに扱える。
 - 様々なAPIが追加された。
- 2008年以降に発表されたウェブブラウザの多くは HTML5 に段階的に対応している。



HTML5で変更になった書き方

- 先頭に「<!DOCTYPE html>」を書く必要がある。
- ファイルの文字エンコーディングは「<meta charset="UTF-8">」のように指定する。
- 特定の要素(p, li, dt, ddなど)は終了タグを省略できる。html, head, bodyも状況によってはOK。
- 属性値を引用符で囲む必要なし。
- script要素(デフォルト値は text/javascript)とstyle要素(デフォルト値は text/css)はtype属性は省略できる。

その他のことについては、以下のサイトを見てください。

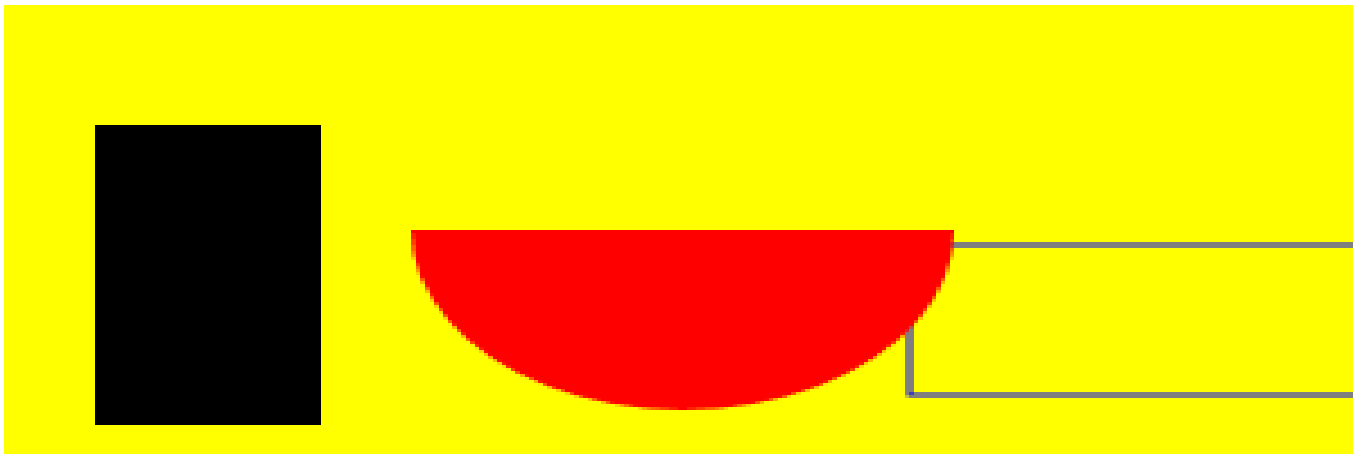
W3C - HTML 5 differences from HTML 4 日本語訳

http://www.html5.jp/trans/w3c_differences.html



2D / 3Dグラフィックス – canvas要素(1/3)

canvasで図形を描く





2D / 3Dグラフィックス – canvas要素(2/3)

```
1 <!DOCTYPE html>↵
2 <html>↵
3 <head>↵
4 <meta charset="UTF-8">↵
5 <title>canvasで図形を描く</title>↵
6
7 <script type="text/javascript">↵
8 <!--↵
9 function sample() {↵
10 //描画コンテキストの取得↵
11 var canvas = document.getElementById('sample1');↵
12 if (canvas.getContext) {↵
13 ↵
14 var context = canvas.getContext('2d');↵
15 ↵
16 //左から20上から40の位置に、幅50高さ100の四角形を描く↵
17 context.fillRect(20,40,50,100);↵
18 ↵
19 //色を指定する↵
20 context.strokeStyle = 'rgb(00,00,255)'; //枠線の色は青↵
21 context.fillStyle = 'rgb(255,00,00)'; //塗りつぶしの色は赤↵
```



2D / 3Dグラフィックス – canvas要素(3/3)

```
22  ←  
23 //左から200上から80の位置に、幅100高さ50の四角の枠線を描く ←  
24 context.strokeRect(200,80,100,50); ←  
25  ←  
26 //左から150上から75の位置に、半径60の半円を反時計回り（左回り）で描く ←  
27 context.arc(150,75,60,Math.PI*1,Math.PI*2,true); ←  
28 context.fill(); ←  
29  ←  
30 } ←  
31 } ←  
32 //--> ←  
33 </script> ←  
34  ←  
35 </head> ←  
36 <body onLoad="sample()"> ←  
37 <h2>canvasで図形を描く</h2> ←  
38 <canvas id="sample1" style="background-color:yellow;"> ←  
39 図形を表示するには、canvasタグをサポートしたブラウザが必要です。 ←  
40 </canvas> ←  
41 </body> ←  
42 </html> [EOF]
```



フォーム要素(1/5)

```
1 <!DOCTYPE html>↵
2 <html>↵
3 <meta charset="UTF-8">↵
4 <title>フォームのサンプル</title>↵
5 <body>↵
6 <h1>フォームのサンプル</h1>↵
7 <form action="" method="post">↵
8 pattern属性(正規表現による入力制限。半角英数のみ)<input type="text" name="userid" pattern="^[0-9A-Za-z]+$"><br>↵
9 複数の値(multiple属性)<input type="text" name="hobby" list="hobbys" multiple><br>↵
10 <datalist id="hobbys">↵
11 <option value="映画">↵
12 <option value="音楽">↵
13 <option value="絵画">↵
14 <option value="読書">↵
15 </datalist>↵
16 search(検索テキスト)<input type="search" name="search" placeholder="検索するキーワードを入力してください" autofocus><br>↵
17 tel(電話番号。required属性:入力必須)<input type="tel" name="tel" required><br>↵
18 url(URL)<input type="url" name="url"><br>↵
19 email(メールアドレス)<input type="email"><br>↵
20 file(ファイル)<input type="file"><br>↵
21 number(数値。最小値:1。最大値:4)<input type="number" min="1" max="4"><br>↵
22 range(一定範囲の数値)<input type="range"><br>↵
23 color(色)<input type="color"><br>↵
24 datetime(UTC(協定世界時)による日時)<input type="datetime"><br>↵
25 date(日付)<input type="date"><br>↵
26 month(月)<input type="month"><br>↵
27 week(週)<input type="week"><br>↵
28 time(時間)<input type="time"><br>↵
29 datetime-local(UTC(協定世界時)によらないローカル日時)<input type="datetime-local"><br>↵
30 <input type="submit" value="送信"><br>↵
31 </form>↵
32 </body>↵
33 </html>↵
```



フォーム要素(2/5)

・Operaの場合

OperaはHTML5対応をちゃんとやっていますね。でもGoogle Chromeのカレンダーのほうが良いと思う。

フォームのサンプル

pattern属性(正規表現による入力制限。半角英数のみ)

複数の値(multiple属性)

search(検索テキスト)

tel(電話番号。required属性:入力必須)

url(URL)

email(メールアドレス)

file(ファイル)

number(数値。最小値:1。最大値:4)

range(一定範囲の数値)

color(色)

datetime(UTC(協定世界時)による日時)

date(日付)

month(月)

week(週)

time(時間)

datetime-local(UTC(協定世界時)による日時)





フォーム要素(3/5)

- Firefoxの場合・・・Pattern, requiredは使えるみたいですが、ほとんどダメですね。

フォームのサンプル

pattern属性(正規表現による入力制限。半角英数のみ)

複数の値(multiple属性)

search(検索テキスト) 検索するキーワードを

tel(電話番号。required属性:入力必須)

url(URL)

email(メールアドレス)

file(ファイル)

number(数値。最小値:1。最大値:4)

range(一定範囲の数値)

color(色)

datetime(UTC (協定世界時) による日時)

date(日付)

month(月)

week(週)

time(時間)

datetime-local(UTC (協定世界時) によらないローカル日時)

入力された値がフィールドに指定された書式と異なります。





フォーム要素(4/5)

- ・IEの場合・・・IE9よりもHTML5への対応している物が増えてます。カレンダーは、次に出るIEに期待しましょう。

フォームのサンプル

pattern属性(正規表現による入力制限。半角英数のみ)

複数の値(multiple属性)

search(検索テキスト)

tel(電話番号。required属性:入力必須)

url(URL)

email(メールアドレス)

file(ファイル)

number(数値。最小値:1。最大値:4)

range(一定範囲の数値)

color(色)

datetime(UTC(協定世界時)による日時)

date(日付)

month(月)

week(週)

time(時間)

datetime-local(UTC(協定世界時)によらないローカル日時)





フォーム要素(5/5)

- ・ Google Chromeの場合・・・ Pattern, required, multiple, numberのmin/max, range, dateは使えるみたいです。

フォームのサンプル

pattern属性(正規表現による入力制限。半角英数のみ)

複数の値(multiple属性)

search(検索テキスト)

tel(電話番号。required属性: 入力必須)

url(URL)

email(メールアドレス)

file(ファイル)

number(数値。最小値: 1。最大値: 4)

range(一定範囲の数値)

color(色)

datetime(UTC(協定世界時)による日時)

date(日付)

month(月)

week(週)

time(時間)

datetime-local(UTC(協定世界時)によらないローカル日時)

2013年(平成25年) 5月

日	月	火	水	木	金	土
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

今日 クリア



ドラッグ&ドロップ

- ・「<http://www.htmq.com/dnd/sample/001.html>」に掲載されていたサンプルを動かしてみました。Safari以外は正常に動作しました。

好きな果物は何ですか？

下のボックスに入れてください。



Geolocation API

- 「<http://www.htmq.com/geolocation/sample/001.html>」に掲載されていたサンプルを動かしてみました。Safari以外は正常に動作しました。



Geolocation APIは、ユーザーの位置情報を扱うためのAPIです。Geolocation Aにも携帯端末などで位置情報を扱うことは出来ましたが、各携帯キャリア独自の詰めキャリアごとの仕様に合わせて開発する必要がありました。

Geolocation APIは、ウェブの標準化団体であるW3Cが仕様策定を進める規格でJavaScriptで位置情報を取得できるように標準化されています。すでに、Firefox・Chrome・Safariなどの一般的なブラウザでサポートされており、スマートフォンのよの携帯端末向けのウェブサイトだけでなく、一般的なブラウザで閲覧するいわゆるユーザーの位置情報を利用したコンテンツを提供することが可能になっています

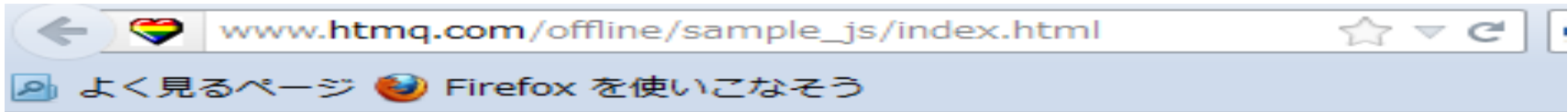
[とにかくサンプルを試してみる](#)





オフラインWebアプリケーション

- 「http://www.htmq.com/offline/sample_js/index.html」に掲載されていたサンプルを動かしてみました。IE9以外は正常に動作しました。インターネットに接続していないオフラインの状態でも、ユーザーがウェブアプリケーション（ウェブサイト）を利用できるようにするための、キャッシュマニフェストと呼ばれる仕様らしいです。



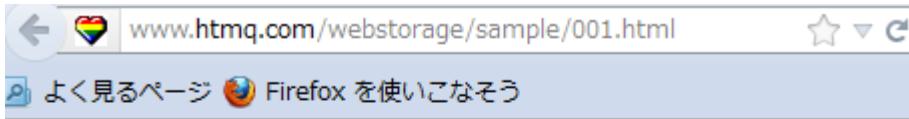
更新をチェック中...
更新はありません
更新をチェック中...
更新はありません

キャッシュを更新する



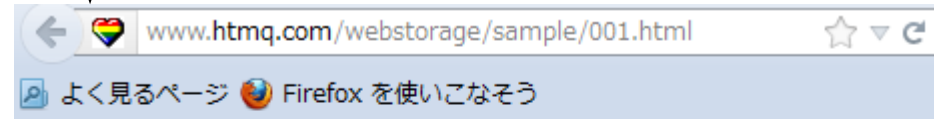
クライアントにデータ保存 – Web Storage API

- 「<http://www.htmq.com/webstorage/sample/001.html>」に掲載されていたサンプルを動かしてみました。全てのブラウザで正常に動作しました。Web Storageは、ユーザーのローカル環境（ブラウザ）にデータを保存するための仕組みです。データの保存・上書き・削除・全クリアなどの操作は、JavaScriptで行うことができます。



キー: 値:

ムリ: ダメ



キー: 値:



書籍(一部)

- ・ Web制作のためのHTML5入門 PC/スマートフォン/タブレット対応(2012/8/1)
- ・ HTML5でモバイルアプリ開発入門—モバイルアプリをプロデュース(2012/7)
- ・ HTML5×スマートフォン サンプルスイート (2012/6/27)
- ・ Dreamweaver+HTML5&CSS3レッスンブック (2012/6/26)
- ・ HTML5クックブック (2012/6/22)
- ・ ゲームプログラミングのためのHTML5 Canvas入門—JavaScriptの基本から本格的ゲームの作成まで(2012/6/21)
- ・ HTML5+JavaScriptで作る 魅せるFacebookページ【改訂2版】 (2012/6/16)
- ・ アプリを作ろう! HTML5入門 — HTML5+CSS3+JavaScriptで学ぶAndroid/iPhoneアプリ作成 (2012/6/7)
- ・ HTML5 Webアプリケーションプログラミング入門 (2012/6)
- ・ つくって覚えるJavaScript+HTML5入門 (2012/5/16)
- ・ 基本から学ぶHTML5+JavaScript iPhone/Android対応 スマートフォンアプリの作り方 (2012/5/2)
- ・ HTML5入門—Flash代替として注目!最新マークアップ言語 (2012/5)
- ・ WebGL+HTML5 3DCGプログラミング入門(2012/5)
- ・ HTML5 + JavaScriptで動く電子書籍アプリを自作する本 —jQuery Mobile、PhoneGap、Titaniumに対応(2012/4/26)
- ・ HTML5の絵本 (2012/4/24)
- ・ HTML5 開発ポケットリファレンス (2012/4/17)





参考情報

HTML5.JP

<http://www.html5.jp/>

「HTML5」とか「アプリ開発入門」

http://www.atmarkit.co.jp/fwcr/design/index/index_html5appli.html

HTML5リファレンス

<http://www.htmq.com/html5/index.shtml>

IE9でなぜかCanvasがうまく動かない

<http://sute.nekok.com/index.php/2011/04/55/>