

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊

コンテンツ作成ガイドブック（案）

2015 年 3 月

目次

はじめに.....	1
1. ガイドブックの目的及び概要.....	2
1.1 ガイドブックの目的.....	2
1.2 学習・教育クラウド・プラットフォームの概要.....	2
1.3 ガイドブックの適用範囲および前提条件.....	2
2. コンテンツとして最低限必要な要素.....	4
2.1 認証.....	4
2.1.1 ユーザ認証結果情報の利用.....	4
2.2 共通インターフェイス.....	5
2.2.1 共通ユーザーインターフェースの利用について.....	5
2.2.2 iFrame を利用した共通ユーザーインターフェースの搭載方法について.....	9
2.2.3 教材コンテンツ画面上の終了ボタン等と共通ユーザーインターフェースの連動について.....	10
2.3 異なる画面サイズへの対応.....	12
2.3.1 概要.....	12
2.3.2 Media Queries.....	12
2.3.3 リキッドレイアウト.....	13
2.4 個人情報.....	13
2.4.1 概要.....	13
3. HTML 5 の要素に関して.....	14
3.1 セキュリティ.....	14
3.1.1 概要.....	14
3.2 Video 要素.....	14
3.2.1 概要.....	14
3.2.2 要素固有の属性.....	15
3.2.3 上記属性を除く IDL 属性とメソッド.....	16
3.2.4 記述例.....	19
3.2.5 外部リンク.....	19
3.3 Audio 要素.....	19
3.3.1 概要.....	19
3.3.2 コンテンツとして audio 要素を利用する際に必要となる要件.....	19
3.3.3 要素固有の属性.....	20
3.3.4 記述例.....	21
3.3.5 外部リンク.....	21
3.4 Track 要素.....	21
3.4.1 概要.....	21

3.5 Canvas 要素	22
3.5.1 概要	22
3.5.2 コンテンツとして canvas 要素を利用する際に必要となる要件	22
3.5.3 要素固有の属性	23
3.5.4 TIPS	23
3.5.5 外部リンク	24
3.6 Progress 要素	24
3.6.1 概要	24
3.6.2 要素固有の属性	24
3.6.3 記述例	24
3.6.4 外部リンク	24
3.7 Meter 要素	25
3.7.1 概要	25
3.7.2 要素固有の属性	25
3.7.3 記述例	26
3.7.4 外部リンク	26
3.8 MathML	26
3.8.1 概要	26
3.8.2 コンテンツ内で数式表現をする場合	26
3.8.3 数式表現の代替案	26
3.8.4 外部リンク	27
3.9 ストレージ	27
3.9.1 概要	27
3.9.2 コンテンツとしてストレージ系 API 利用する際に必要となる要件	27
3.9.3 ストレージ系 API の概要と特徴	27
3.9.4 ブラウザ別の API サポート状況と対処方法	28
3.9.5 オリジン単位で保存されるデータ	29
3.9.6 データの消去	29
3.9.7 プライバシーの対処	30
3.10 フォーム	30
3.10.1 概要	30
3.10.2 コンテンツとして video 要素を利用する際に必要となる要件	30
3.10.3 新たなフォームコントロール	30
3.10.4 新たな input 要素の属性	34
3.10.5 コントロールをサポートしていないブラウザの対処	35
3.10.6 入力制約のエラーメッセージ	36
3.11 グラデーション	37
3.11.1 概要	37
3.11.2 主要ユーザーエージェントに向けた実装方法	37
3.11.3 ツール	37
3.12 Writing-mode	37
3.12.1 概要	37

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

3.12.2	プロパティに指定できる値.....	37
3.12.3	記述例	38
3.12.4	ウェブブラウザ別表示例	38
3.12.5	TIPS.....	38
3.12.6	外部リンク	39

はじめに

本書は、平成 26 年度「クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証」において、技術仕様検討の一環として作成した、コンテンツ作成者向けの「コンテンツ作成ガイドブック」である。

1. ガイドブックの目的及び概要

1.1 ガイドブックの目的

「コンテンツ作成ガイドブック」（以下、「ガイドブック」）は、平成 26 年度「クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証」において検討を行った学習・教育クラウド・プラットフォームについて、コンテンツ作成者がコンテンツ作成する際に満たすべき要件を示すことを目的とする。

1.2 学習・教育クラウド・プラットフォームの概要

学習・教育クラウド・プラットフォームの概要を図 1-1 に示す。

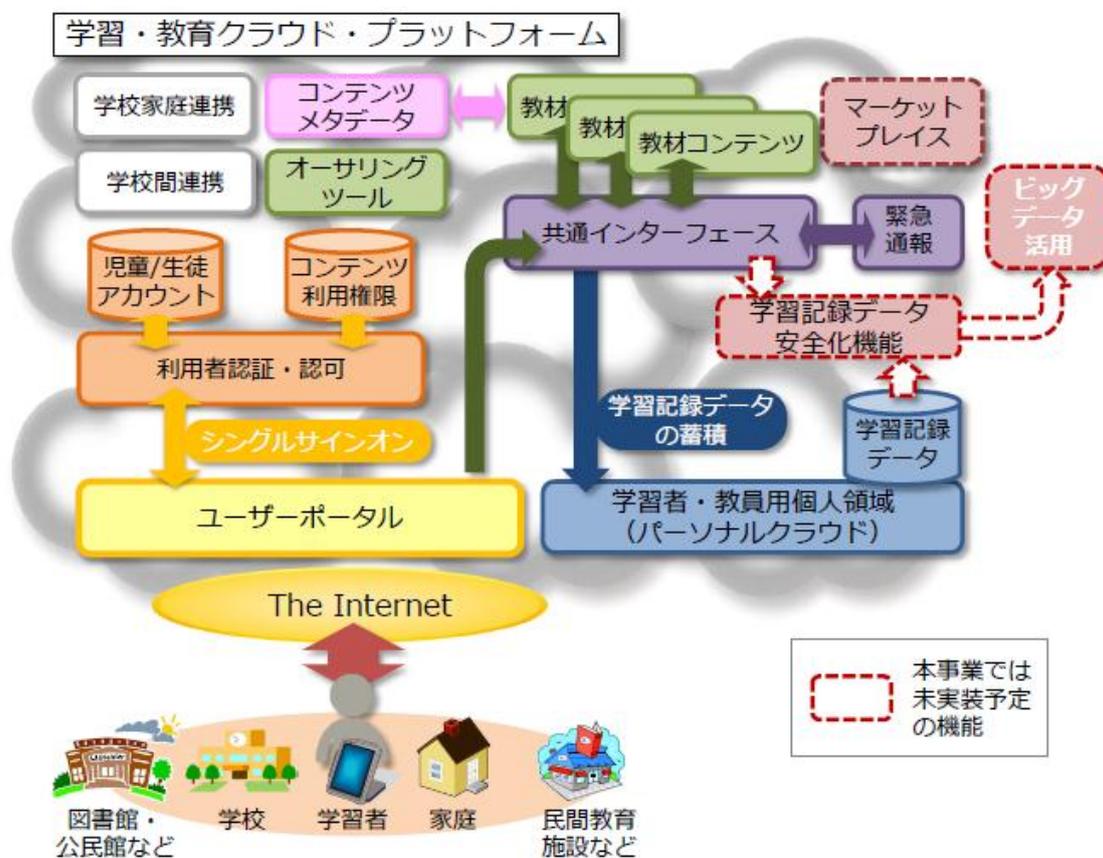


図 1-1 学習・教育クラウド・プラットフォームの概要

1.3 ガイドブックの適用範囲および前提条件

学習・教育クラウド・プラットフォームは、前項で示した各機能をパブリック・クラウドサービス（IaaS）上で実現することを前提とする。

教育用コンテンツの規格としては HTML5 を採用した。HTML5 とは、WWW に関する各

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

種技術の標準化を進める団体である W3C により策定が進められている HTML の最も新しい標準規格である。ただ、広義には、厳密に W3C が定義した HTML5 仕様のみではなく、Web API と呼ばれるウェブアプリケーション向けの機能に関わる規格と、CSS3 と呼ばれる表示に関わる規格など、ウェブ技術全般の急速な進化を指すことが多い。

HTML5 の規格自体は W3C によりほぼ策定を終えたと言える一方、ブラウザの対応は進行中で、ブラウザの種類やバージョンによって実装のレベルは異なっている。

このため、HTML5 の規格に厳密に則ってコンテンツを制作したとしても、すべてのブラウザで正常に動作するわけではないのが現状である。

本事業では、HTML5 のコンテンツを制作する上で、W3C の策定した仕様を前提としているが、ブラウザの差異に対する対応等、コンテンツに共通するノウハウはドキュメントにまとめて共有することとし、コンテンツ作成ガイドブックを制作した。また、共通インターフェースやユーザー認証、学習記録データの書き出しなど、本事業のプラットフォームに適合させるための基準も必要であり、それらもコンテンツ作成ガイドブックにまとめた。

2. コンテンツとして最低限必要な要素

本事業のコンテンツは共通のシステム上で動作する。そのため、コンテンツには、コンテンツ本体の他にシステムとの接続のために指定のファイルを含める必要がある。

本章では、共通システムとの接続に必要なファイルの説明と、個別のコンテンツ毎に必要なカスタマイズについて記述する。

2.1 認証

2.1.1 ユーザ認証結果情報の利用

以下のコードを JavaScript ファイル内に記述（ペースト）する。

記述例

```
var bInit = false;
var oComIf = null;
var szVendorPfx = "Comp1";
var szProductPfx = 'Product1';
var nComIfType = ComIF.typeMin;
var szComifColor = 'b';
var szAppRoom = 'A';
var bGroup = false;
var szName = '匿名';

document.onreadystatechange = function () {
    if (document.readyState == "complete") {
        init();
    }
};

window.onunload = function() {
    if (oComIf != null) {
        oComIf.closeBar();
        oComIf = null;
    }
};

function init() {
    if (bInit == false) {
        bInit = true;
        oComIf = new ComIF(szVendorPfx, szProductPfx, nComIfType, szComifColor,
            szAppRoom, bGroup);
        oComIf.loadBar(function(szRes) {
            if (szRes !== 'success') {
                alert(szRes);
            } else {
                oComIf.getAuthorizedInfo(szVendorPfx, szProductPfx, function(szRes) {
                    if (szRes !== 'success') {
                        alert(szRes);
                    }
                });
            }
        });
    }
}
```

```
    } else {  
        var arAuthInfo = oComIf.getUserInfo(); // ←※  
        szName = arAuthInfo.uname;  
    }  
});  
}  
});  
}  
}
```

※で取得可能なユーザー認証結果情報は以下となる。

group	// ユーザー認証機能による学校間連携用グループ情報
org	// 半角文字 組織名称情報
jaorg	// 全角文字 組織名称情報
gr	// 学年情報
cl	// クラス情報
uname	// 半角文字 ユーザー表示名情報
jauname	// 全角文字 ユーザー表示名情報
opid	// 学外一意 ユーザー (ID ハッシュ化) 情報
aff	// 教員, 保護者, 生徒 識別情報

※の例では、「getUserInfo()」が返した連想配列から「uname」項目を szName 変数に格納している。

認証システムに関連するポリシーから、必ず全ての情報が参照可能とは限らない。教材コンテンツによっては、null 値となる項目が混在することがある。

opid は、ユーザー認証フェデレーションの範囲内にて一意性、および、永続性が保証される情報である。

各項目のフォーマット等に関しては、ユーザー認証機能より提供される仕様書等を参照すること。

student / teacher / parent 等

2.2 共通インターフェイス

2.2.1 共通ユーザーインターフェイスの利用について

HTML5 教材コンテンツ画面上に共通ユーザーインターフェイス機能を搭載する手順について説明する。

共通ユーザーインターフェイスは、所謂「ツールバー」として、教材コンテンツ画面に被る形で、ブラウザのクライアント領域下部の高さ値 42 ピクセル を占有する。

パラメータにて、クライアント領域下部の全幅を占有するか、または、右端の最低利用域のみとするかを選択できる。

クライアント領域下部を占有する要件が教材コンテンツに不都合な場合には、iFrame を

併用することで、描画領域の重なりに関する問題を回避することも可能である。

(1) ファイル構成とライブラリの読み込み定義

① 教材コンテンツのファイル構成について

3 種のファイル（my_app.html / my_app.css / my_app.js）にて教材コンテンツが構成されること（②参照）を前提に以降の説明をする。既存の教材コンテンツにおいて、js（Java Script）プログラムを別ファイルとしては持たずに html 内に記述している場合には、該当箇所に対する説明として読み替えること。また、js プログラムを一切含まない教材コンテンツの場合には、新たに my_app.js ファイルを作成し、script タグを以下のように <head>・・・</head> タグ間等に加えること。

[例]

```
<script type="text/javascript" src="my_app.js" ></script>
```

（上記では、my_app.html と同じ PATH 内に「my_app.js」名の Java Script ファイルを設置している）

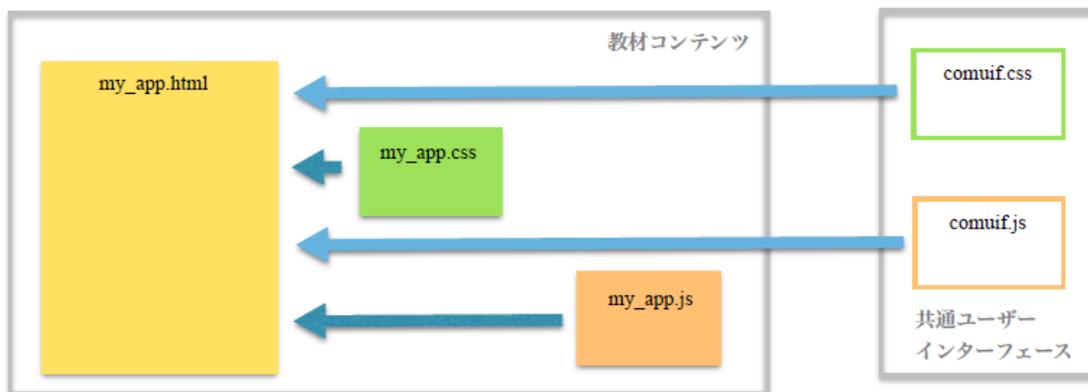
② 共通ユーザーインターフェースを利用するには、以下の 2 行を <head> 前のどこか、もしくは、</body> と </html> の間に記述する。

```
<link rel="stylesheet" href="https://comif.ledup.jp/cdn/sdk/comuif.css" />
```

```
<script type="text/javascript" src="https://comif.ledup.jp/cdn/sdk/comuif.js"></script>
```

複数の HTML ファイルにて構成される教材コンテンツの場合には、それぞれの HTML ファイルに上記定義を加えること。また、②以降で説明される作業も同じようにおこなう必要がある。

ドメイン名や PATH 等については、今後、変更される場合がある。また、教材コンテンツを配置するプラットフォームが複数存在する場合には、プラットフォーム毎の指定ドメイン名や PATH 等を書き換えた上で教材コンテンツの配置をお願いする可能性が考えられる。



ここまでの説明にて、my_app.html の <head> タグ内定義は以下のようなになる。

```
<head>
  .
  .
  <link rel="stylesheet" href="https://comif.ledup.jp/cdn/sdk/comuif.css" />
  <link rel="stylesheet" href="my_app.css" />
  .
  .
  <script type="text/javascript" src="https://comif.ledup.jp/cdn/sdk/comuif.js"></script>
  <script type="text/javascript" src="my_app.js"></script>
</head>
<body>
  .
```

(2) ユーザー識別情報、教材コンテンツ識別情報の登録

「https://comif.ledup.jp/secure/user/」にアクセスし「ユーザー識別情報」および「教材コンテンツ識別情報」を登録する。

上記ページを利用するには、別途提供されるユーザー認証システムへの ID、パスワードが必要である。「ユーザー識別情報」のユーザーとは、共通ユーザーインターフェース機能を利用する教材コンテンツを提供される方を指す。

1. はじめて利用するときには「ユーザー識別情報の登録ページに移動」をクリックします。
2. ご担当者様所属先等の代表メールアドレス（2カ所）＜※＞と、ご希望のユーザー識別情報（半角英数字3～8文字）を入力し「新規登録」をクリックします。
3. 表示された「パスワード」を忘れずに記録します（参照、および、再発行手続き機能は未搭載です）。「サインインページに移動」をクリックすると「1」画面に戻ります。
4. メールアドレスとパスワードを入力してサインインします。
5. 教材コンテンツ単位で教材コンテンツ識別情報を登録します。

※ 個人情報保護の観点から、可能な限り個人のメールアドレスは利用しないでください。

(3) JavaScript プログラムの記述（既存 JavaScript プログラムへの追記）

以下のコードを JavaScript ファイル内に記述（ペースト）する。
 (JavaScript コードが無い教材コンテンツの場合には、先に作成した my_app.js ファイル内に記述する。)

記述例

```
var bInit = false;
var oComIf = null;
var szVendorPfx = "Comp1"; // Comp1 に(2)で登録した「ユーザー識別情報」を記述← (ア)
var szProductPfx = 'Product1'; // Product1 に(1)で登録した「教材コンテンツ識別情報」を
記述← (イ)
var nComIfType = ComIF.typeMin; // 共通ユーザーインターフェースのタイプや利用機
能の指定← (ウ)
var szComifColor = 'b'; // 共通 I/F の色 (黄(y), 青(b), 赤(r))の指定← (エ)
var szAppRoom = 'A';
var bGroup = false;
var szName = '匿名';

document.onreadystatechange = function () {
  if (document.readyState === "complete") {
    init();
  }
};
window.onunload = function() {
  if (oComIf != null) {
    oComIf.closeBar();
    oComIf = null;
  }
};
function init() {
  if (bInit == false) {
    bInit = true;
    oComIf = new ComIF(szVendorPfx, szProductPfx, nComIfType, szComifColor,
szAppRoom, bGroup);
    oComIf.loadBar(function(szRes) {
      if (szRes !== 'success') {
        alert(szRes);
      } else {
        oComIf.getAuthorizedInfo(szVendorPfx, szProductPfx, function(szRes) {
          if (szRes !== 'success') {
            alert(szRes);
          } else {
            var arAuthInfo = oComIf.getUserInfo();
            szName = arAuthInfo.uname;
          }
        });
      }
    });
  }
}
```

(ア) (イ) 共、登録済みの識別情報を設定する必要がある。

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

(エ) にて、青・赤・黄のいずれかで共通ユーザーインターフェースの色を指定する。

教材コンテンツにて既に「document.onreadystatechange」「window.onunload」イベントをフックされている場合には、既存のプログラム内に前頁の記述例部を追記すること。

JQuery 等を利用されている場合には「document.onreadystatechange」「window.onunload」イベントフック部を読み替えて参照すること。尚、共通ユーザーインターフェースでは、jQuery 等のライブラリを一切利用していない。

(4) 共通ユーザーインターフェースへのパラメータ

共通ユーザーインターフェースの利用機能やタイプについては (ウ) にて 設定する。(今後、変更や追加が予定される情報である)

ComIF.typeMin //リアルタイム Web 機能を利用しない

ComIF.typeDef //リアルタイム Web 機能を最低限利用 (IWB 連携, 防災アラートにて利用)

ComIF.typeFloating // 共通ユーザーインターフェースのツールバーを画面右下に最低限のサイズで表示

(設定例) `var nComIfType = ComIF.typeMin | ComIF.typeFloating; // 上記 2 つを共に指定`

なお平成 26 年度事業における実証環境では「リアルタイム Web 機能」を利用しないこととなった。については、上記パラメータの「ComIF.typeDef」は利用せずに、必ず「ComIF.typeMin」を指定すること。

2.2.2 iFrame を利用した共通ユーザーインターフェースの搭載方法について

画面レイアウトへの影響等にて教材コンテンツが全てのクライアント領域を占有する必要がある場合には、教材コンテンツを iFrame 内に読み込むことで共通ユーザーインターフェース機能を搭載することが可能である。

ここでは、サンプル「https://comif-test.ledup.jp/secure/example/rp_test/cifwrapper.html」をベースに説明する。

(1) 共通ユーザーインターフェース搭載 iFrame 教材コンテンツラッパーの利用

以下の手順にて利用できる。

- ① 2.2.1(2)の内容を参考に「ユーザー識別情報」および「教材コンテンツ識別情報」を登録する。
- ② サンプル「rp_test」フォルダー内の「cifcontwrapper.html」を、既存教材コンテンツのトップページファイルのパス下に配置し「cifcontwrapper.html」をエディタで開く。
- ③ 以下の 2 行に①での登録情報を設定する。

```
var cif_szVendorPfx = ""; // 企業情報) 共通ユーザーI/F MBaaS 利用情報管理にて登録したプリフィクスを渡す
```

```
var cif_szProductPfx = ""; // 教材コンテンツ情報) <上記同様>
```

- ④ 共通ユーザーインターフェースの色等、必要に応じ、他のパラメータを調整する。
- ⑤ 教材コンテンツのトップページのファイル名が「index.html」では無い場合には、以下の「index.html」部を書き換える。

```
<iframe id="contframe" src="index.html" frameborder="0"></iframe>
```

- ⑥ ブラウザにて「cifcontwrapper.html」にアクセスし、動作を検証する。

(2) iFrame ラッパー利用時の注意点

教材コンテンツ内に「target= “_top”」等の iFrame から脱出するリンク定義が混じる場合には、該当箇所を変更いただく等の対応が要求されることが考えられる。

2.2.3 教材コンテンツ画面上の終了ボタン等と共通ユーザーインターフェースの連動について

教材コンテンツ画面上に、学習を終了するためのボタン等が設置されている場合、共通ユーザーインターフェースの「マイポータル画面への復帰ボタン」や「ログアウトボタン」と連動させることが可能である。

(1) 教材コンテンツと共通ユーザーインターフェースの連動

ここでは、サンプル「<https://comif-test.ledup.jp/secure/example/cifcontmsg/>」をベースに説明する。

共通ユーザーインターフェースの「マイポータルへの復帰ボタン」「ログアウトボタン」と連動するには、教材コンテンツの JavaScript プログラム内に以下のコードを加える。

```
oComIf.loadBar(function(szRes) {  
  if (szRes !== 'success') {  
    alert(szRes);  
  } else {  
    oComIf.getAuthorizedInfo(szVendorPfx, szProductPfx, function(szRes) {  
      if (szRes !== 'success') {  
        alert(szRes);  
      } else {  
        // ハンドラーの登録  
        oComIf.setEvHandler(retMenu, ComIF.ev2Portal); // ①  
        oComIf.setEvHandler(doLogout, ComIF.evLogout); // ②  
        (以降、省略)
```

①では、共通ユーザーインターフェースの「マイポータルへの復帰ボタン」が利用者によって選択された時に起動してもらい関数を登録しており、②も同様に「ログアウトボタン」押下時の関数を登録している。「retMenu」は、教材コンテンツ側にて、マイポータルへの復帰ボタンが押下された時におこなう処理が記述された関数名であり、「doLogout」は、ログ

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック
アウトボタン押下時のそれであることを示している。

それぞれの関数内は、以下となる。

```
function retMenu() {  
    .  
    . // 教材コンテンツにて必要な処理  
    .  
    oComIf.onMyPortal(false); // 共通ユーザーインターフェースに「マイポータル画面への  
    復帰」処理を実行させる (③)  
}
```

(ログアウト処理 (④) も同様に「oComIf.onLogout(false);」をコールする)

尚、教材コンテンツ画面にて「終了」等の処理が選択された場合には、教材コンテンツ側にて必要な後処理をおこなった上で、③、または、④のように共通ユーザーインターフェース側の関数を呼ぶが、その前に、①、②にて登録したハンドラーを外す手続きが以下のように必要となる。

```
oComIf.removeEventHandler(ComIF.ev2Portal); // ①のハンドラーを削除  
oComIf.removeEventHandler(ComIF.evLogout); // ②のハンドラーを削除
```

教材コンテンツ画面上の終了等の機能を、プラットフォーム内においては「マイポータル画面への復帰」に置き換えるのか、または「ログアウト」なのかについては、教材コンテンツの制作者が選択、もしくは、双方を網羅可能な仕様としている。

(2) iFrame ラッパー内教材コンテンツと共通ユーザーインターフェースの連動

ここでは、サンプル¹をベースに説明する。

iFrame 内教材コンテンツと、iFrame 外設置共通ユーザーインターフェース間においてメッセージのやりとりをおこなうにはサンプルの js フォルダ内に格納されている「cifmsg.js」ライブラリーが必要である。コピーして利用すること。

教材コンテンツへのプログラム追記内容については、概ね、iFrame ラッパーが利用されていない前ページまでの説明とさほど変わりはない。

以下のように、教材コンテンツがブラウザにロードされたタイミングにて、共通ユーザーインターフェースの「マイポータルへの復帰ボタン」「ログアウトボタン」機能に対する教材コンテンツ側のハンドラー関数を登録する。

教材コンテンツがブラウザに読み込まれた直後に処理をおこなう例：

```
document.onreadystatechange = function () {
```

¹ <https://comif-test.ledup.jp/secure/example/iframecontmsg/cifwrapperfull.html>

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

```
if (document.readyState == "complete") {  
    // ここに処理を記述する ①  
}  
};
```

①内の記述例：

```
cifSetEvHandler(false, retMenu); // マイポータルへの復帰処理に対する関数 (retMenu) を  
                                登録 ②  
cifSetEvHandler(true, doLogout); // ログアウト処理に対する関数 (doLogout) を登録 ③
```

教材コンテンツから共通ユーザーインターフェースに対し、マイポータルへの復帰、または、ログアウト処理を実施させるには、以下のように記述する。

```
cifSendMessage(false, 'launch'); // マイポータル画面への復帰  
cifSendMessage(true, 'launch'); // ログアウトの実行
```

尚、教材コンテンツ側の終了ボタン等が押下された場合には、②、③で登録したハンドラーを削除する必要がある。

```
cifSendMessage(false, 'remove'); // ②ハンドラーの削除  
cifSendMessage(true, 'remove'); // ③ハンドラーの削除
```

2.3 異なる画面サイズへの対応

2.3.1 概要

CSS3 では Media Queries という、様々なデバイスに対し、最適な表示を提供するための仕組みがある。また、ウェブブラウザに対して表示する幅を%で指定することで、ブラウザの幅によってコンテンツのサイズを変更する、いわゆるリキッドレイアウトと呼ばれる可変レイアウトで、種々の環境での表示を行う仕組みがある。学習・教育クラウド・プラットフォームで提供されるコンテンツは、あらゆる画面サイズで最適なプレゼンテーションを提供する必要がある。

2.3.2 Media Queries

記述例

```
@media screen and (max-width: 980px) { /* view port の横幅が 980px 以下の時に適当される CSS */  
    #hogeHoge{  
        width: 900px;  
    }  
}
```

@media screen and (max-width: 760px) { /* view port の横幅が 760px 以下の時に適当される CSS */

```
#hoge{
  width: 700px;
}
}
@media screen and (max-width: 480px) {...}
@media screen and (max-width: 320px) {...}
```

2.3.3 リキッドレイアウト

記述例

```
<div id="wrapper">
  <div id="main">
    <div id="mainContent">div#main</div>
  </div>
  <div id="sub">div#sub</div>
</div>
div#wrapper {
  width: 80%;
}
div#main {
  width: 100%;
  float: right;
}
div#mainContent {
  margin-left: 200px;
}
div#sub {
  width: 200px;
  margin-right: -200px;
  float: left;
}
```

2.4 個人情報

2.4.1 概要

コンテンツ内では、児童生徒の名前を入力させたり、手書きで書かせたりしてはならない。システム上では、児童生徒の名前は一切扱わない。

3. HTML 5 の要素に関して

HTML5 に関連する多くの規格の策定状況はかなり成熟してきており、規格を満たす実装を待つ状態と言える。しかし、現在は目下、ブラウザの実装が進行中であり、ブラウザの種類やバージョンによっては、未実装の箇所や、動作が異なる部分がある。

本章では、その差異を考慮に入れ、各端末で問題なくコンテンツを動作させるために必要な情報、手段をまとめている。

3.1 セキュリティ

3.1.1 概要

HTML5 を始めとしたウェブ技術の進展に伴い、ウェブアプリケーションのセキュリティ問題が心配される。2013 年 10 月 30 日に JPCERT/CC より「HTML5 を利用した Web アプリケーションのセキュリティ問題に関する調査報告書」が一般公開された。コンテンツ開発の前に、必ず、このドキュメントを一読し、セキュリティを考慮したアプリケーション設計を行うこと。

●JPCERT/CC HTML5 を利用した Web アプリケーションのセキュリティ問題に関する調査報告書

<http://www.jpcert.or.jp/research/html5.html>

また、概要を把握するために、JPCERT/CC 職員が行った過去の講演や記事を読むことを推奨する。

●JPCERT/CC 講演・執筆活動一覧

<http://www.jpcert.or.jp/kouen/2014.htm>

●今から始める HTML5 セキュリティ

<http://www.jpcert.or.jp/present/2014/20140129HTML5.pdf>

3.2 Video 要素

3.2.1 概要

video 要素は HTML5 で新たに導入された要素である。これまで、ウェブページに動画を組み込む場合は、プラグインが利用されてきた。しかし、HTML5 では、video 要素を使うことでウェブページに動画を組み込むことが可能になった。さらに、HTML の要素となることで、CSS で見た目の定義を行うことができ、JavaScript により制御を行えるようになった。本項では、コンテンツとして適合する video 要素の記述方法について記載している。

動画ファイル

動画ファイルは以下の形式を推奨する。

- ・【推奨】H.264 + AAC + MP4 (拡張子:.mp4, MIME タイプ:video/mp4)
- ・【任意】WebM (拡張子:.webm, MIME タイプ:video/webm)

H.264 は、Windows 8.1/IE11, Android/Chrome, iOS7/Safari のいずれもサポートしている。

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

また、Firefox も、Windows 8.1, MacOS X 上でならサポートしている。

H.264 は、スマートフォンやタブレットにおいては、ハードウェアデコードするため、CPU は PC と比べて非力ながらも再生パフォーマンスが期待できる。

WebM は、Chrome, Firefox, Opera がサポートしているが、コンテンツが対象とする Windows 8.1/IE11 と iOS7 Safari がサポートしていないため、特別な事情がない限りは推奨しない。

H.264 をサポートしていないブラウザおよび OS 環境においても再生させたい場合に、H.264 のフォールバックとして採用することを推奨する。

マークアップ例

```
<video controls width="360" height="240">
```

```
  <source src="v.mp4" type="video/mp4">
```

```
  <source src="v.webm" type="video/webm">
```

```
</p>ご利用のブラウザではビデオが再生できません。
```

```
  ビデオファイルは<a href="v.mp4">こちら</a>からダウンロードできます。</p>
```

```
</video>
```

もし動画ファイルがユーザーに対してダウンロードが許可されるのであれば、video 要素の開始タグと終了タグの間に、動画ファイルのダウンロードリンクを用意するのが良い。こうすることで、仮に video 要素そして対象の動画コーデックをサポートしていないブラウザからアクセスがあった際に、便利ではないが、動画を閲覧する手段を提供することが可能になる。

Web サーバ

動画ファイルの拡張子に合わせた MIME タイプをユーザーエージェントへ送信できること。もしウェブサーバーの設定が難しい場合、動画ファイルが存在するディレクトリ内に .htaccess ファイルを用意し、以下のディレクティブを記述することで、対処できる場合がある。

```
AddType video/mp4 .mp4
```

```
AddType video/webm .webm
```

3.2.2 要素固有の属性

src

動画ファイルのアドレスを指定する。(src 属性が存在しない場合、video 要素の中に 1 つ以上の source 要素を含む。)

crossorigin

CORS(Cross-Origin Resource Sharing)で使用される属性で、異なるサイトのファイルにアクセスする場合に、認証情報を使用するかどうかを指定する。

anonymous :

認証情報を使用しない

use-credentials :

認証情報を使用する

poster

ブラウザが動画ファイルのロードを開始して再生準備が整うまでの間に表示する画像フ

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

ファイルのアドレスを指定する。この画像は、指定の動画ファイルが存在しなかった、または、再生できなかった場合にも表示される。

preload

動画ファイルのデータの先読みの挙動の指針をブラウザに伝えることができる。なお、この属性は、あくまでもブラウザにヒント与えるに過ぎず、実際に指定の挙動を保証するものではない。指定可能な値は、以下の通りである。

none

音声ファイルを先読みしない。

metadata

動画の長さなどのメタデータのみを先読みする。

auto

動画ファイルを先読みする（デフォルト）。

preload

属性が指定されていない場合は、デフォルトである **auto** が採用される。

autoplay

論理属性である。属性を指定することで、動画の読み込みが行われると自動的に再生を開始する。（※Nexus10,iPad の標準ブラウザでは、自動再生は機能しない。なので、本属性は使用しない。）

mediagroup

複数の動画ファイルをグルーピングする際のグループ名を指定する。

loop

論理属性である。属性を指定することで、ループ再生する。（※Nexus10,iPad の標準ブラウザでは、自動再生は機能しない。なので、本属性は使用しない。）

muted

論理属性である。属性を指定することで、音をミュートする。（※Nexus10,iPad の標準ブラウザでは、自動再生は機能しない。なので、本属性は使用してはならない。）

controls

論理属性である。属性を指定することで、ユーザーエージェント標準のコントローラを表示する。動画の再生制御を JavaScript で行わないのであれば、必ず、この属性はセットすること。セットしないと、ユーザーが動画再生をコントロールすることができなくなるので注意すること。

width

横幅を指定する属性である。

height

高さを指定する属性である。

3.2.3 上記属性を除く IDL 属性とメソッド

videoWidth

(読み込み専用)動画の幅を返す。

videoHeight

(読み込み専用)動画の高さを返す。

currentSrc

(読み込み専用)現在の動画のアドレスを返す。動画が存在しない場合、空の文字列を返す。

audio

errore state

error

(読み込み専用)MediaError オブジェクトを返す。エラーが無ければ null を返す。

network state

networkState

(読み込み専用)ネットワークアクティビティを返す。

NETWORK_EMPTY = 0: まだ初期化されていない。すべての属性は初期状態のままである。

NETWORK_IDLE = 1: 要素のリソース選択アルゴリズムはアクティブで、かつ、リソースが選択された状態である。しかし、実際には、この時点ではネットワークを使っていない。

NETWORK_LOADING = 2: ユーザーエージェントは現在データをダウンロードしようとしている。

NETWORK_NO_SOURCE = 3: 要素のリソース選択アルゴリズムはアクティブであるが、まだ採用するリソースは見つかっていない。

bufferd

(読み込み専用) ユーザーエージェントがバッファした動画の範囲を表す TimeRanges オブジェクトを返す。

load()

要素をリセットする。そして、最初から新たな動画を選択し、それをロードし始める。

canPlayType(type)

空の文字列(否定的な回答)、“maybe”(確信がない)、“probably”(確信)を返す。指定された type の動画を再生できるかどうかを、ユーザーエージェントがどれほど確信できるかどうかに基づき返す。

ready state

readyState

(読み込み専用) 現在再生位置のレンダリングに関して、要素の現在の状態を表す値を返す。

HAVE_NOTHING = 0: 動画に関する情報はまだ何も入手できていない。現在再生位置のデータも何も入手できていない。networkState 属性が NETWORK_EMPTY にセットされたメディア要素は、常に、HAVE_NOTHING 状態にある。

HAVE_METADATA = 1: 十分なリソースが取得でき、そのリソースの尺がわかっている。video 要素の場合、そのビデオの寸法もわかっている。もうシーク時に API から例外が発出されることはない。現在再生位置の直近のメディア・データはまだ入手できていない。テキスト・トラックは準備できている。

HAVE_CURRENT_DATA = 2: 現在再生位置の直近のデータが取得できている。しかし、ユーザーエージェントは、再生方向に現在再生位置を進めようとする、すぐに HAVE_METADATA 状態に戻ってしまう状態、または、再生方向に取得すべきデータがない状態のいずれかである。例えば、ビデオでは、これは、ユーザーエージェントが現在のフレームのデータを持っているけれども、次のフレームは持っていない状態に、そして、再生が

終わったときに一致する。

HAVE_FUTURE_DATA = 3 : 現在再生位置の直近のデータが取得できている。さらに、ユーザーエージェントは、すぐに **HAVE_METADATA** 状態に戻ること無しに、少なくとも少しは再生方向に現在再生位置を進めることができる状態である。例えば、ビデオでは、ユーザーエージェントが少なくとも現在のフレームと次のフレームのデータを持ち合わせている状態である。ユーザーエージェントは、再生が終わったら、この状態に留まることはできない。

HAVE_ENOUGH_DATA = 4 : ユーザーエージェントは、実効再生レートで進めるとしても、再生がメディア・リソースの最後に到達する前に、現在再生位置が利用可能なデータを追い越してしまうことはないほどの割合でデータをフェッチしていると、見積もっている。

seeking

(読み込み専用) ユーザーエージェントが現在シーク中なら **true** を返す。

playback state

currentTime

現在再生位置を返す。単位は秒である。

duration

メディア・リソースの最後の時間を返す。単位は秒である。

paused

メディア要素が停止されているかどうかを返す。

defaultPlaybackRate

メディア・リソースを再生させたい速度を与える。これは、その本来の速度の倍数である。取得時においては、それにセットされた最後の値を返さなければいけない。

playbackRate

この属性は、実効再生レートを与える。これは、メディア・リソースが再生する速度のことで、その本来の速度の倍数である。取得時においては、最後にセットされた値を返さなければいけない。

played

この属性は、メディア・リソースの範囲があれば、これを表す新たな静的な正規化。

TimeRanges

オブジェクトを返さなければいけない。これは、ユーザーエージェントが、この属性が評価される時点で、すでにレンダリングした範囲である。

ended

メディア要素の再生が終了し再生方向が前進なら **true** を、そうでなければ、**false** を返す。

play()

動画の再生を開始する。

pause()

動画を一時停止する。

controls

volume

属性には、現在再生しているメディア・リソースの再生ボリュームの値を与える。(0.0~1.0) 取得時は、現在の再生ボリュームを返す。

muted

属性には、現在再生しているメディア・リソースのミュートの設定を与える。取得時は、現在の状態を返す。

3.2.4 記述例

コンテンツの要件を満たす video 要素の記述例

```
<video controls width="360" height="240">  
  <source src="video/video.mp4" type="video/mp4" />  
  <source src="video/video.ogv" type="video/ogg" />  
  <source src="video/video.webm" type="video/webm" />  
  <p> ご使用中のブラウザでは動画再生機能はご利用できません。 <a  
href="video/vide.mp4">こちら</a>からダウンロードしてください。 </p>  
</video>
```

3.2.5 外部リンク

W3C HTML5 - The video element (<http://www.w3.org/wiki/HTML/Elements/video>)

W3C (HTML5 Video Events and API)

3.3 Audio 要素

3.3.1 概要

audio 要素は HTML5 で新たに導入された要素である。ウェブページに音声を組み込む場合は、プラグインが利用されてきた。しかし、HTML5 では、audio 要素を使うことでウェブページに音声を組み込むことが可能になった。

さらに、JavaScript により制御を行えるようになった。本項では、コンテンツとして適合する video 要素の記述方法について記載している。

3.3.2 コンテンツとして audio 要素を利用する際に必要となる要件

音声ファイル 音声ファイルは以下の形式を推奨する。

- 【推奨】 AAC (拡張子:.aac, MIME タイプ:audio/aac)
- 【推奨】 MP3 (拡張子:.mp3, MIME タイプ:audio/mp3)
- 【任意】 Vorbis + Ogg (拡張子:.ogg, MIME タイプ:audio/ogg)
- 【任意】 WAVE (拡張子:.wav, MIME タイプ:audio/wave)

AAC と MP3 は、Windows 8.1/IE11, Android/Chrome, iOS7/Safari のいずれもサポートしている。また、Firefox も、Windows 8.1, MacOS X 上でならサポートしている。Ogg、WAVE は、コンテンツが対象とする Windows 8.1/IE11 と iOS7 Safari がサポートしていないため、特別な事情がない限りは推奨しない。AAC または MP3 のいずれもサポートしていないブラウザおよび OS 環境においても再生させたい場合に、AAC または MP3 のフォールバックとして採用することを推奨する。

マークアップ例

<audio controls>

<source src="a.mp3" type="audio/mp3">

<source src="a.ogg" type="audio/ogg">

<p>ご利用のブラウザではオーディオが再生できません。

オーディオファイルはこちらから

ダウンロードできます。</p>

</video>

もし音声ファイルがユーザーに対してダウンロードが許可されるのであれば、audio 要素の開始タグと終了タグの間に、音声ファイルのダウンロードリンクを用意するのが良い。こうすることで、仮に video 要素そして対象の音声コーデックをサポートしていないブラウザからアクセスが有った際に、便利ではないが、音声コンテンツを利用する手段を提供することが可能になる。

Web サーバ 音声ファイルの拡張子に合わせた MIME タイプをユーザーエージェントへ送信できること。もしウェブサーバーの設定が難しい場合、音声ファイルが存在するディレクトリ内に.htaccess ファイルを用意し、以下のディレクティブを記述することで、対処できる場合がある。

AddType audio/mp4 .acc

AddType audio/mpeg .mp3

AddType audio/ogg .ogg

AddType audio/wav .wav

3.3.3 要素固有の属性

src

音声ファイルのアドレスを指定する。(src 属性が存在しない場合、audio 要素の中に 1 つ以上の source 要素を含む。)

crossorigin

CORS(Cross-Origin Resource Sharing)で使用される属性で、異なるサイトのファイルにアクセスする場合に、認証情報を使用するかどうかを指定する。

anonymous : 認証情報を使用しない

use-credentials : 認証情報を使用する

preload

音声ファイルのデータの先読みの挙動の指針をブラウザに伝えることができる。なお、この属性は、あくまでもブラウザにヒント与えるに過ぎず、実際に指定の挙動を保証するものではない。指定可能な値は、以下の通りである。

none : 音声ファイルを先読みしない。

metadata : 音声の長さなどのメタデータのみを先読みする。

auto : 音声ファイルを先読みする (デフォルト)。

preload 属性が指定されていない場合は、デフォルトである auto が採用される。

autoplay

論理属性である。属性を指定することで、音声の読み込みが行われると自動的に再生を開始する。Nexus10,iPad の標準ブラウザでは、自動再生は機能しない。なので、本属性は使用

しない。

mediagroup

複数の音声ファイルをグルーピングする際のグループ名を指定する。

loop

論理属性である。属性を指定することで、ループ再生をする。Nexus10 標準ブラウザでは、ループは機能しない。なので、本属性は使用しない。

muted

論理属性である。属性を指定することで、音をミュートする。Nexus10,iPad の標準ブラウザでは、ミュートは機能しない。なので、本属性は使用しない。

controls

論理属性である。属性を指定することで、ユーザーエージェント標準のコントローラを表示する。音声の再生制御を JavaScript で行わないのであれば、必ず、この属性はセットすること。セットしないと、ユーザーが音声再生をコントロールすることができなくなるので注意すること。

3.3.4 記述例

コンテンツの要件を満たす audio 要素の記述例

```
<audio controls>
<source src="audio/audio.mp4" type="audio/aac" />
<source src="audio/audio.mp3" type="audio/mp3" />
<source src="audio/audio.ogg" type="audio/ogg" />
<source src="audio/audio.wav" type="audio/wav" />
<p>ご使用中のブラウザでは音声再生機能はご利用できません。<a href="audio/audio.mp4">こちら</a>からダウンロードしてください。</p>
</audio>
```

3.3.5 外部リンク

W3C HTML5 - The audio element (<http://www.w3.org/wiki/HTML/Elements/audio>)

3.4 Track 要素

3.4.1 概要

track 要素は HTML5 で新たに導入された要素である。video 要素や audio 要素に字幕やキャプションを標準化された方法で追加することができる。

・ track 要素記述例

```
<video src="v.mp4" controls>
  <track kind="subtitles" src="subtitle_ja.vtt" srclang="ja" label="日本語" default>
  <track kind="subtitles" src="subtitle_en.vtt" srclang="en" label="English">
</video>
```

track 要素をサポートしているのは、Windows8.1/IE11 と Chrome のみである。しかし、学習教材の動画において字幕やキャプションは生徒にとっては理解に役立つため、字幕やキャ

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

プションは出来る限りにおいて用意することを推奨する。すべての環境で確実に字幕やキャプションを表示させるには、track 要素を使わずに、字幕やキャプションを直接埋め込んだ形で動画を用意する。もし字幕やキャプションをテキストデータとして用意する場合、track 要素を使いつつも、多くの環境で字幕やキャプションをサポートするために、JavaScript ライブラリの利用を推奨する。

Captionator.js

Captionator.js は、字幕ファイルのフォーマットに WebVTT を採用している。

WebVTT: The Web Video Text Tracks Format

- WebVTT の記述例

WEBVTT FILE

00:00:01.342 --> 00:00:06.763

みなさん、こんにちわ。

00:00:08.211 --> 00:00:14.536

お集まり頂き、ありがとうございます。

00:00:16.325 --> 00:00:20.442

これから教材の使い方について紹介させていただきます。

Web サーバーは、WebVTT ファイルの拡張子に合わせた MIME タイプをユーザーエージェントへ送信する必要がある。もしウェブサーバーの設定が難しい場合、WebVTT ファイルが存在するディレクトリ内に .htaccess ファイルを用意し、以下のディレクティブを記述することで、対処できる場合がある。

```
AddType text/vtt .vtt
```

3.5 Canvas 要素

3.5.1 概要

canvas 要素は HTML5 で新たに導入された要素である。canvas 要素を使うことで、JavaScript によって動的に二次元グラフィックスを描画することができる。本項では、コンテンツとして適合する canvas 要素の記述方法について記載している。

3.5.2 コンテンツとして canvas 要素を利用する際に必要となる要件

(1) ソースコード

- 三次元グラフィックスを扱う WebGL も canvas 要素を使うが、実証研究用のデバイスの一部でサポートされていない。したがって、教材として無くてはならないコンテンツとして WebGL を採用してはならない。
- canvas につける ID は、canvas 保管時の JSON キー、もしくは、画像ファイル名の一

部として利用される。よって、JSON キーやファイル名に使用できない文字を使ってはならない。

(2) アニメーション

- canvas を使ってアニメーションを実現する際には、JavaScript の setInterval() や setInteval() といったタイマー関数ではなく、W3C の Timing control for script-based animations 仕様で規定された requestAnimationFrame() メソッドを採用することを推奨する。なめらかなアニメーションを実現しようとする、概ね 16 ミリ秒ごとに描画処理を実行 (約 60fps) する必要がある。タイマー関数では、16 ミリ秒間隔でタイマーをセットしても、その精度は低く、また、必ずしもディスプレイのリフレッシュレートに一致しないため、非常に非効率である。また、デバイスのバッテリー消費が激しいと言われている。

一方、requestAnimationFrame() は、デバイスのディスプレイのリフレッシュレートに合わせてコールバック関数を呼び出す。

- W3C - Timing control for script-based animations

・ サンプルコード

// 現在のタイムスタンプ

```
var start = window.performance.now();
```

// canvas にフレームを描画する関数

```
var startAnimation = function(now) {
```

```
    // canvas に描画
```

```
    ...
```

```
    // 次のリフレッシュ時に startAnimation() 関数を呼び出す
```

```
    window.requestAnimationFrame(startAnimation);
```

```
};
```

// アニメーション開始

```
window.requestAnimationFrame(startAnimation);
```

3.5.3 要素固有の属性

width

横幅を指定する属性である。

height

高さを指定する属性である。

3.5.4 TIPS

簡単に canvas を扱えるよう、Javascript ライブラリが公開されている。

CreateJS

CreateJS は、ActionScript に近いコーディングスタイルを採用している。Flash クリエイターには馴染みやすいライブラリといえる。また、Flash Professional CC には、Flash コンテンツを CreateJS から利用できる形に変換する「Toolkit for CreateJS」が用意されている。シン

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック
フルな Flash コンテンツなら、過去の Flash コンテンツ資産をそのまま変換できる可能性がある。
ある。

3.5.5 外部リンク

W3C HTML5 - The canvas element (<http://www.w3.org/wiki/HTML/Elements/canvas>)

W3C HTML Canvas 2D Context (<http://www.w3.org/TR/2dcontext/>)

3.6 Progress 要素

3.6.1 概要

progress 要素は HTML5 で新たに導入された要素である。何かしらの処理の進捗を表すために使用する。実証研究用のデバイスにおいては、progress 要素がサポートされているが、進捗バーの見た目はブラウザによって異なるので注意すること。進捗バーは、ユーザーにとって進捗の概略を直感的に把握することはできるが、詳細な値を読み取ることができない。もしユーザーに値を読み取らせる必要がある場合は、progress 要素の外にテキスト情報も用意することを推奨する。

・ コードサンプル

```
<progress id="pb" value="50" max="100"></progress> <span id="pt">50%</span>
```

スクリプトを使って動的に進捗を変化させる場合も、テキストの情報の更新も忘れないようにすること。

```
document.getElementById("pb").value = 30;  
document.getElementById("pt").textContent = "30%";
```

3.6.2 要素固有の属性

value

進捗を表す数値を指定する。max 属性に指定した値以下の数値を指定しなければいけない。max 属性を指定しない場合は、0.0 以上 1.0 以下の数値を指定しなければならない。

max

進捗の完了を表す数値を指定する。0 以上の数値を指定しなければならない。max 属性が指定されなければ、1 が指定されたとして処理される。

3.6.3 記述例

```
<p><progress value="3" max="5">60%</progress></p>
```

3.6.4 外部リンク

W3C HTML5 - the progress element (<http://www.w3.org/wiki/HTML/Elements/progress>)

3.7 Meter 要素

3.7.1 概要

`meter` 要素は HTML5 で新たに導入された要素である。最小値と最大値が指定された限定された範囲における値を表すために使用する。実際にはバーの形をしたメーターとしてレンダリングされる。実証研究用のデバイスにおいては、IE11 に `meter` 要素が実装されていない。それ以外のブラウザでは `meter` 要素がサポートされているが、メーターの見た目はブラウザによって異なるので注意すること。

JavaScript のライブラリを使うことで、サポート外のブラウザでも `meter` 要素のレンダリングを行うことが可能である。

(1) html5.jp(<meter> polyfill (jQuery プラグイン))

メーターは、ユーザーにとって概略を直感的に把握することはできるが、詳細な値を読み取ることができない。もしユーザーに値を読み取らせる必要がある場合は、`meter` 要素の外にテキスト情報も用意することを推奨する。

(2) コードサンプル

```
<meter id="mb" min="0" max="128" value="24"></meter>
```

```
<span id="mt">128 個の空きに対して 24 個が使用中である。</span>
```

スクリプトを使って動的に進捗を変化させる場合も、テキストの情報の更新も忘れないようにすること。

```
document.getElementById("mb").value = 30;
```

```
document.getElementById("mt").textContent = "128 個の空きに対して 30 個が使用中である。
```

```
";
```

3.7.2 要素固有の属性

value

メーターに表す値を指定する

min

メーターが表す範囲の下限値を指定する

low

低領域の上限値を指定する

high

高領域の下限値を指定する

max

メーターが表す範囲の上限値を指定する

optimum

最適値を指定する

3.7.3 記述例

`<p><meter value="3" min="0" max="5"> 5 分の 3 </meter></p>`

3.7.4 外部リンク

W3C HTML5 - the meter element (<http://www.w3.org/wiki/HTML/Elements/meter>)

3.8 MathML

3.8.1 概要

MathML は、これまで HTML 中に直接記述することができなかった。しかし、HTML5 の仕様では、直接記述することができるようになった。ただ、仕様上では規定されたが、実装されているユーザーエージェントは大変限られている。今回の実証実験で使われるデバイスの標準ウェブブラウザでは、サポートされていない。したがって、MathML は使用してはならない。

3.8.2 コンテンツ内で数式表現をする場合

コンテンツ内で数式表現をする場合、次の節で詳細を記述をする、代替案から条件に適したものを選んで適用すること。

1. 簡易な数式を利用する場合 (四則演算等) → HTML+CSS
2. 高等数学の数式を利用する場合(平方根、積分等) → MathJax
3. 初等数学指導のために独特な表現がある場合 (分数、割り算、掛け算の筆算) → 画像

3.8.3 数式表現の代替案

MathML を使用せずに、今回の実証実験で使用するデバイスにて数式の記述を行う方法は、以下の方法がある。

MathJax

MathJax は、世界最大の数学論文データベース MathSciNet で採用されている、数式表示の仕組みである。JavaScript で作られており、多くのブラウザで、同じ表示を実現することが可能である。数式の記載方法は、公式ページを参照のこと。初等数学 (割り算の筆算等) はサポートされていない。

- ・ MathJax
- ・ CSS を使った表現

W3C が CSS を使った MathML の表示を文書として公開している。CSS ファイルを読み込んで、MathML の記述を行うことで、ある程度の表現を行うことができる。MathML で初等数学の規定をしているため、初等数学の表現も可能である。

画像

想定する表現ができない場合、画像で対応することも可能である。画像で対応する場合、

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック
読み上げなどで同じ情報を伝えられるように配慮すること。

3.8.4 外部リンク

W3C(Mathematical Markup Language (MathML) Version 3.0)

3.9 ストレージ

3.9.1 概要

近年のブラウザには、データをブラウザに蓄積する API がいくつか追加されている。旧来の Cookie とは異なり、大きなデータや大量のデータの蓄積や、蓄積したデータを検索できる点が、大きな特徴である。

本項では、これらストレージ関連の API の概要に加え、コンテンツとして適合するストレージの活用について記載している。

3.9.2 コンテンツとしてストレージ系 API 利用する際に必要となる要件

- ストレージ系 API は、それぞれに特徴がある。用途を見極めて API を選択すること。詳細は「ストレージ系 API の概要と特徴」および「ブラウザ別の API サポート状況と対処方法」を参照すること。
- 異なるアプリがお互いのストレージデータに干渉しないよう、データ蓄積メカニズムを設計すること。詳細は「オリジン単位で保存されるデータ」を参照すること。
- ブラウザのストレージに蓄積されたデータは消去されてしまう場合があることを想定すること。詳細は「データの消去」を参照すること。
- ストレージには、決して個人を特定できる情報やプライバシーに関わる情報を保存してはならない。詳細は「プライバシーの対処」を参照すること。

3.9.3 ストレージ系 API の概要と特徴

近年のブラウザは、4 種類のストレージ系 API が実装されている。ここでは、それぞれについて、概要と特徴を解説する。

Web Storage

Web Storage は、テキストデータを一意の値となるキーに結び付けて蓄積するための枠組みを提供する非常にシンプルなストレージ API である。すでにすべてのメジャーブラウザに実装されている。

Web Storage には 2 種類のストレージが用意されている。一つは永続的にデータが保存されるローカルストレージである。ローカルストレージに保存されたデータは、ブラウザを閉じてでも残り続ける。もう一つは、ページがブラウザに表示されている間だけ存続するセッションストレージである。ブラウザを閉じれば、保存したデータは完全に消えて無くなる。

Web Storage は他のストレージ系 API とは異なり、書き込みと読み取りともに同期処理である。また、容量も 5MB~10MB に制限されている。Web Storage は、アプリケーションの設定データなどの、小さなデータを保存するのに適している。

Indexed Database API

Indexed Database API（以降は Indexed DB と呼ぶ）は、構造化データを保存する枠組みを提供する。検索高速化のためのインデックスや、排他処理を実現するトランザクションをサポートしている点が特徴である。大きなデータを保存する場合や、大量のレコードを扱う場合に有効である。

File API: Directories and System

File API: Directories and System（以降は File System と呼ぶ）は、ファイルシステムにアクセスするための API である。ブラウザが管理するサンドボックス化された仮想的なファイルシステムにのみアクセスすることができる。

File API: Directories and System は、他のストレージ系 API とは異なり、データをファイルという実体として保存する。さらに、ディレクトリを自由に作成することもできる。

SQL Database API

SQL Database API（以降は SQL DB と呼ぶ）は、SQL を使ってデータにアクセスする枠組みを提供する。W3C では SQL DB は廃止となったが、Safari や Chrome などのブラウザが実装していることもあり、現在でもよく使われている API である。

3.9.4 ブラウザ別の API サポート状況と対処方法

前述のストレージ系 API は、ブラウザによってサポート状況が大きく異なる。実証研究用のデバイスの対象ブラウザの API サポート状況は下表のとおりである。

ブラウザごとのストレージ系 API サポート状況（2014 年 2 月現在）				
ブラウザ	Web Storage	Indexed DB	File System	SQL DB
Internet Explorer 11 for Windows 8.1	○	○	×	×
Chrome for Android	○	○	○	○
Safari for iOS	○	×	×	○

すべてのブラウザでサポートされているストレージ系 API は、Web Storage のみである。もし小さなデータを保存するのであれば、Web Storage を利用することを推奨する。

もし大きなデータや大量のレコードを保存したい場合は、Safari では SQL DB を使わざるを得ない。一方、Internet Explorer 11 と Chrome では Indexed DB が利用可能である。面倒ではあるが、ブラウザによって、API を使い分ける必要がある。

SQL DB は、前述のとおり、すでに仕様が廃止になっている。Chrome でも SQL DB は利用可能であるが、余程の理由がない限り、Chrome においては、Indexed DB の採用を推奨する。Safari で SQL DB を採用するのは、あくまでも代替手段が無いため、と考えること。

File System は、技術的には、Indexed DB で代用可能である。現在のところ、File System API は、Chrome のみに実装されているが、他のブラウザベンダーは File System を実装する予定がない。一方、Indexed DB は、Safari を除いて、すべてのメジャーブラウザが実装している。従って、現段階では、File System と Indexed DB の選択に迫られた場合は、将来性も考慮し、Indexed DB の採用を推奨する。

3.9.5 オリジン単位で保存されるデータ

前述のストレージ系 API はいずれも、オリジン単位にデータ領域が用意される。オリジンとは、URL のうち、以下の範囲を指す。

`http://www.example.jp:80`

スキーム (`http:`) からポート番号 (`80`) までが、オリジンと呼ばれる。もしオリジンが異なると、別のデータ領域が用意され、お互いに参照しあうことはできない。

同じアプリケーションが複数のオリジンを扱う場合の注意点

もしアプリケーションが複数のオリジンにまたがって構成されている場合には注意すること。例えば、以下のオリジンは、すべて別オリジンとして扱われる。

`http://sub.example.jp:80` (ホスト名が異なる)

`http://www.example.jp:8080` (ポート番号が異なる)

`https://www.example.jp` (スキームが異なる)

もし上記のオリジンを一つのアプリケーションが使っている場合、同じアプリケーションといえども、それぞれで保存されたデータをお互いに読み取ったり書き込んだりすることはできない。もしストレージのデータを使う場合は、同じオリジンでアプリを構築すること。

異なるアプリが同じオリジンを扱う場合の注意点

逆に、複数のアプリが同じオリジンを使っている場合にも注意が必要である。以下の URL は、いずれも同じオリジンとして扱われる。

`http://www.example.jp/app1/`

`http://www.example.jp/app2/`

`http://www.example.jp/app3/`

このように、アプリケーションをディレクトリで分けてしまうと、それぞれのアプリケーションが同じデータ領域を参照してしまう。アプリケーションごとに、それぞれが影響しあわないように調整するのは効率的ではない。可能な限り、アプリケーションごとにオリジンを割り当てることを推奨する。もしアプリケーションごとにドメインを取得するのが難しい場合は、サブドメインを使うのがよい。

`http://app1.example.jp/`

`http://app2.example.jp/`

`http://app3.example.jp/`

3.9.6 データの消去

どのストレージ系 API で蓄積されたデータも、さまざまな理由で削除されてしまう可能性がある。削除されてしまうきっかけとしては、ユーザーが意図的にブラウザのメニューからデータを削除した場合や、ディスクの空き容量が無くなった場合が挙げられる。コンテンツにおいては、決して、保存したデータが永続的に保存されていることを前提にアプリを設計することがないよう注意すること。どうしてもデータを保存しておきたい場合は、クラウド (サーバー) 側にもデータを蓄積する方式を検討すること。

3.9.7 プライバシーの対処

ストレージ系の API を使えば、さまざまなデータをデバイス上に保存することが可能である。ゆえに、プライバシー問題を孕む場合がある。コンテンツにおいては、決して、ユーザー個人を特定できるような情報や、プライバシーに関わる情報を保存しないこと。

もし仮にアプリケーションがクロスサイトスクリプティングの脆弱性を孕んでいたとする。悪意のある第三者によって攻撃を受けると、ストレージに蓄積されたデータを盗みとられることが想定される。アプリケーションにセキュリティ上の脆弱性があるとはならないものの、完全に脆弱性を排除したことを保証することは不可能である。従って、万が一、未発見の脆弱性を突いた攻撃を受けた場合においても、ユーザーに迷惑をかけないようにするために、不必要なデータの保存は避けること。

3.10 フォーム

3.10.1 概要

これまで HTML にはテキスト入力フィールド、パスワード入力フィールド、ラジオボタン、チェックボックスなどのフォームコントロールが用意されているが、HTML5 では、アプリケーションとして求められるフォームコントロールがいくつか追加された。また、input 要素にも、新たな属性が追加されている。

本項では、HTML5 で新たに導入された新フォームコントロールの概要に加え、コンテンツとして適合する新フォームコントロールの記述方法について記載している。

3.10.2 コンテンツとして video 要素を利用する際に必要となる要件

- 該当のフォームコントロールをサポートしていないブラウザを利用するユーザーでも、コンテンツが利用できるよう配慮すること。後述の「コントロールをサポートしていないブラウザの対処」を参照すること。
- 入力制限を持つフォームコントロールを使う場合、利用者が、そのコントロールに何を入力すべきかがわかるよう、適切な説明を加えること。後述の「入力制約のエラーメッセージ」を参照すること。
- 入力制限をマークアップで用意したとしても、確実に期待する値がサーバーに送信されるとは限らない。該当のコントロールをサポートしていないブラウザだけでなく、意図的に不正な値を送信することは簡単である。従って、たとえマークアップで入力制限を用意したとしても、サーバー側でも、同様またはそれ以上の入力チェックを行うようにすること。

3.10.3 新たなフォームコントロール

以下に HTML5 に新たに加わったフォームコントロールの概要について解説する。

検索キーワード `<input type="search">`

検索キーワード入力を想定したテキスト入力フィールドである。該当の OS の検索キーワード入力欄と同じ形状のコントロールとして表示されるため、ユーザーはそれが検索キーワ

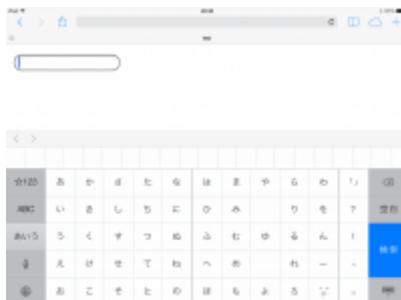
平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

ード入力欄であることが分かりやすくなる。

マークアップ例

```
<input type="search" name="q">
```

表示例



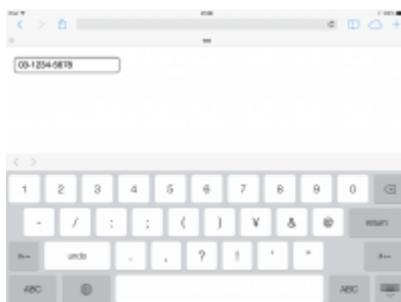
電話番号 <input type=tel>

電話番号入力を想定したテキスト入力フィールドである。見た目は通常のテキスト入力フィールドと同じであるが、タブレットなどでは、数字入力モードのキーパッドが表示されるため、ユーザーにとっては使いやすいコントロールになる。

マークアップ例

```
<input type="tel" name="telno" value="03-1234-5678">
```

表示例



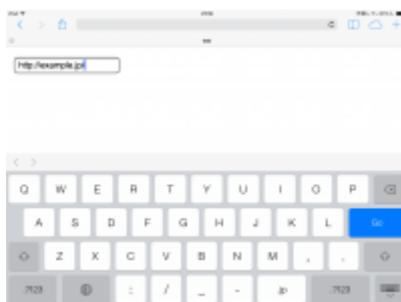
[編集]URL <input type=url>

URL 入力を想定したテキスト入力フィールドである。見た目は通常のテキスト入力フィールドと同じであるが、タブレットなどでは、アルファベット入力モードの入力パッドが表示されるため、ユーザーにとっては使いやすいコントロールになる。

マークアップ例

```
<input type="url" name="blog" value="http://example.jp/">
```

表示例



また、HTML5 仕様上、このコントロールは入力制約を持つ。HTML5 仕様に完全に準拠

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

したブラウザであれば、もし URL に適さない値を入力してフォームを送信しようとする
と、サーバーに送信する前に、該当のフォームコントロールにエラーメッセージが表示さ
れる。

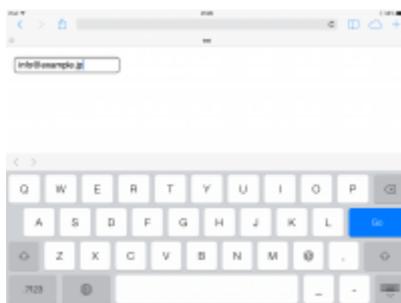
メールアドレス <input type=email>

メールアドレス入力を想定したテキスト入力フィールドである。見た目は通常のテキスト
入力フィールドと同じであるが、タブレットなどでは、アルファベット入力モードの入力パ
ッドが表示されるため、ユーザーにとっては使いやすいコントロールになる。

マークアップ例

```
<input type="email" name="mailaddress" value="info@example.jp">
```

表示例



また、HTML5 仕様上、このコントロールは入力制約を持つ。HTML5 仕様に完全に準拠
したブラウザであれば、もしメールアドレスに適さない値を入力してフォームを送信しよ
うとすると、サーバーに送信する前に、該当のフォームコントロールにエラーメッセ
ージが表示される。

日付 <input type=date>

日付入力を想定したフォームコントロールである。通常は入力時にカレンダーまたはセ
レクターが表示され、ユーザーは日付を選択できる。値がセットされた時点では、
"2004/07/23" のように表示されるが、サーバーに送信される値は"2004-07-23"の
ようにハイフン区切りの形式である。

マークアップ例

```
<input type="date" name="birthday" value="2004-07-23">
```

表示例



数値 <input type=number>

数値入力を想定したフォームコントロールである。見た目は通常のテキスト入力フィー
ルドと同じであるが、タブレットなどでは、数値入力モードの入力パッドが表示され
るため、ユーザーにとっては使いやすいコントロールになる。

マークアップ例

```
<input type="number" name="age" value="11">
```

表示例



数値ではない値を入力してフォームを送信しようとする、サーバーに送信する前に、該当のフォームコントロールにエラーメッセージが表示される。また、HTML5 仕様上、このコントロールは入力制約を持つ。HTML5 仕様に完全に準拠したブラウザであれば、もし数字に適さない値を入力してフォームを送信しようとする、サーバーに送信する前に、該当のフォームコントロールにエラーメッセージが表示される。

範囲限定の数値 <input type=range>

スライダーが表示され、ユーザーはマウスまたはタッチによってスライダーを動かすことで数値をセットする。

マークアップ例

```
<input type="range" name="level" value="3" min="0" max="5" step="1">
```

表示例



min 属性はスライダーの最小値を、**max** 属性はスライダーの最大値を表す。**step** 属性は、値の変化の幅を表す。上記の例では、0 から 5 の間で、1 ずつスライダーを動かせることを意味する。つまり、取りうる値は 0, 1, 2, 3, 4, 5 のいずれかとなる。

その他

2014 年 2 月現在、HTML5 仕様の最新のドラフトでは、上記の他に、以下の新コントロールが定義されている。

- ・ <input type=color>
- ・ <input type=datetime>
- ・ <input type=month>
- ・ <input type=week>
- ・ <input type=time>

ただし、現在のところ、これらのコントロールは"at risk"と呼ばれる状態になっており、

W3C の仕様策定において最終的にどうなるかが確定していない。仕様から削除される可能性もあるし、仕様に残ったとしても、内容が変更される可能性がある。これらのコントロールに関しては、HTML5 仕様の確定を待った上で、ブラウザの実装状況を踏まえて採用を判断すること。

3. 10. 4 新たな input 要素の属性

以下に HTML5 に新たに加わったフォームコントロールの概要について解説する。

required 属性

入力必須を指定する論理属性である。

マークアップ例

```
<input type="text" name="nickname" required>
```

HTML5 仕様に完全に準拠したブラウザであれば、もし何も入力せずにフォームを送信しようとする、サーバーに送信する前に、該当のフォームコントロールにエラーメッセージが表示される。

pattern 属性

入力値パターンを正規表現で指定することができる。

マークアップ例

```
<input type="text" name="zip" pattern="^\d{3}\Y-\d{4}$">
```

HTML5 仕様に完全に準拠したブラウザであれば、もしパターンに一致しない値を入力してフォームを送信しようとする、サーバーに送信する前に、該当のフォームコントロールにエラーメッセージが表示される。

min 属性と max 属性と step 属性

min 属性と max 属性は、それぞれ、最小値と最大値を指定する。step 属性は、入力できる値のステップ (刻み) を指定する。これらの属性は、type 属性が date、number、range の input 要素で指定可能である。

マークアップ例

```
<input type="date" name="reservation-date" min="2014-05-01" max="2014-05-10" step="1">
```

HTML5 仕様に完全に準拠したブラウザであれば、指定の範囲の値が入力できないよう制限されるか、または、指定の範囲を超えた値を入力してフォームを送信しようとする、サーバーに送信する前に、該当のフォームコントロールにエラーメッセージが表示される。

placeholder 属性

placeholder 属性は、そのテキスト入力フィールドに何を入力したら良いかをユーザーに気づかせるようなヒントを指定することができる。placeholder 属性にセットされたヒントは、該当のテキスト入力フィールドに何も値がセットされていない場合に限り、そのフィールド内に表示される。

このコンテンツ属性は、input 要素の type コンテンツ属性の値が text、search、url、tel、email、password の場合に利用することができる。

マークアップ例

```
<input type="search" name="q" placeholder="検索キーワード">
```

表示例



3.10.5 コントロールをサポートしていないブラウザの対処

ここで紹介したフォームコントロールは、すべて実証研究用のデバイス上のブラウザでサポートされているわけではない。これらのコントロールをサポートしていないブラウザでは、単なるテキスト入力フィールドとして表示される。たとえば、date タイプの input 要素は、IE11 ではテキスト入力フィールドとして表示されてしまう。

従って、コントロールの説明文は、どのブラウザに対しても、不自然にならないよう考慮すること。

不適切な説明文の例

```
<p>誕生日を選択してください : <input type="date" name="birthday"></p>
```

iPad の Safari では date タイプの input 要素はセレクトアとして表示されるため、説明文の「選択してください」は適切なのであるが、IE11 では単なるテキスト入力フィールドとして表示されるため、この説明文は不自然になる。

適切な説明文の例

```
<p>誕生日を指定してください : <input type="date" name="birthday"></p>
```

このように、どちらのブラウザでも不自然とならない文章を考えること。

どうしても文章を統一できない場合は、スクリプトを使って説明文を切り替える方法が使える。

マークアップ例

```
<p>  
  <span id="desc" data-desc2="誕生日を選択してください">  
    誕生日を YYYY-MM-DD 形式で入力してください</span>  
  : <input type="date" name="birthday">  
</p>
```

スクリプト例

```
var input_el = document.querySelector("input[name='birthday']");  
if(input_el.type === "date") {  
  var desc_el = document.querySelector("#desc");  
  desc_el.textContent = desc_el.getAttribute("data-desc2");  
}
```

上記の例では、date タイプをサポートしていないブラウザ向けの説明文が span 要素でマークアップされている。しかし、span 要素の data-desc2 属性に、date タイプをサポートしているブラウザ向けの説明文も記述されている。これを、スクリプトを使って、差し替える。

スクリプトでは、もし input 要素が date タイプをサポートしていれば、input 要素の DOM オブジェクトの type プロパティは、文字列"date"を返す。サポートしていなければ、文字列"text"を返す。これを応用して、date タイプをサポートしたブラウザの場合にのみ、説明文

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック
を差し替える。

3.10.6 入力制約のエラーメッセージ

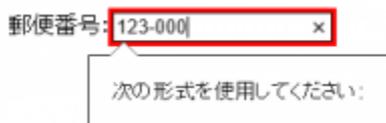
入力制約を持つコントロールに、不適切な値を入力して、フォームを送信しようとする、HTML5 仕様に準拠したブラウザでは、該当のコントロールにエラーメッセージが表示される。

マークアップ例

```
<p><label>郵便番号 : <input type="text" name="zip" pattern="^\d{3}\Y-\d{4}$"></label></p>
```

このコントロールは郵便番号を入力させるために用意されている。そのため、`pattern` 属性を使って、日本の郵便番号形式 (例: 123-0000) の値しか受け付けられないよう制限している。もしパターンが違う値を入力してフォームを送信しようとする、IE11 では、次のようなエラーが表示される。

パターンエラーの表示例



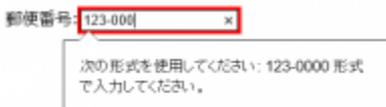
このように、ユーザーにとって、どうすれば良いかが分かりにくいエラーメッセージが表示される。このメッセージは、ブラウザによって異なる。これを回避するために、入力制約を持つコントロールには、必ず `title` 属性をマークアップすること。

title 属性のマークアップ例

```
<p><label>郵便番号 : <input type="text" name="zip" pattern="^\d{3}\Y-\d{4}$"  
title="123-0000 形式で入力してください。"></label></p>
```

`title` 属性をマークアップしておく、その値がエラーメッセージの最後に追加される。

パターンエラーの表示例



なお、2014 年 2 月現在で、iOS 版 Safari では、入力制約を持つコントロールのエラーメッセージ表示がサポートされていない。従って、このエラーメッセージが表示されることを前提としたアプリケーション設計は避けること。

もし、入力値チェックとエラーメッセージ表示を、ブラウザに頼らず、スクリプトを使って実現したい場合は、このブラウザのエラーメッセージを表示させないようにすることが可能である。form 要素に `novalidate` 属性をマークアップすると、ブラウザによる入力制約のチェックが無効になる。

novalidate 属性のマークアップ例

```
<form action="#" method="post" novalidate>  
...  
</form>
```

3.11 グラデーション

3.11.1 概要

CSS を利用したグラデーションでは、画像を利用したグラデーションと違い、サーバに画像を読み込みに行く必要がない。そのため、読み込み回数を減らすことができ、グラデーションの変更もコードを書き換えるのみなど、作業の負担が小さく済むメリットがある。

しかし、現状として、グラデーションの表示には、ユーザーエージェント毎の実装の差異があるため、同じコンテンツを表示するためには、ソースコードに工夫をする必要がある。

3.11.2 主要ユーザーエージェントに向けた実装方法

```
.gradient {  
    background: -webkit-linear-gradient(top, #1e5799 0%,#2989d8 50%,#207cca 51%,#7db9e8  
100%); /* Chrome10+,Safari5.1+ */  
    background: -ms-linear-gradient(top, #1e5799 0%,#2989d8 50%,#207cca 51%,#7db9e8  
100%); /* IE10+ */  
    background: linear-gradient(to bottom, #1e5799 0%,#2989d8 50%,#207cca 51%,#7db9e8  
100%); /* W3C */  
}
```

3.11.3 ツール

adobe Dreamweaver help(背景へのグラデーションの適用)
Ultimate CSS Gradient Generator

3.12 Writing-mode

3.12.1 概要

writing-mode プロパティは、文書表示方向の指定に使われるプロパティである。長らく、Internet Explorer の独自拡張だったが、W3C が定める、CSS3 に組み込まれた。本プロパティと direction プロパティを使用することにより、日本語の縦書表現を実現することが可能になる。

3.12.2 プロパティに指定できる値

horizontal-tb

行内のテキストは横に進み、行は上から下に進む。横書き。行内のテキストの左右の方向は direction プロパティによって指定する。

vertical-rl

行内のテキストは縦に進み、行は右から左に進む。縦書き。行内のテキストの方向は direction プロパティによって指定する。

vertical-lr

行内のテキストは縦に進み、行は左から右に進む。縦書き。行内のテキストの方向は `direction` プロパティによって指定する。

3.12.3 記述例

`writing-mode` プロパティは、CSS3 に規定されているが、現状として、ブラウザごとの独自実装となっている。そのため、で本プロパティを使用する場合は、次のような記述が必要である。

コンテンツの要件を満たす `writing-mode` の記述例

```
.verticalWrite {
    -ms-writing-mode:tb-rl;          /* Internet Explorer */
    -webkit-writing-mode:vertical-rl; /* Webkit Browser */
    writing-mode:vertical-rl;        /* CSS3 */
}
```

3.12.4 ウェブブラウザ別表示例

今回の実証研究で使用されるウェブブラウザ別の <code>writing-mode</code> プロパティ表示例			
表示コード(該当部分のみ)	android(Chrome)	iPad(Safari)	windows8(Internet Explorer10)
<pre>.verticalText { -webkit-writing-mode:vertical-rl; writing-mode:vertical-rl; direction:ltr; line-height:200%; }</pre>			

3.12.5 TIPS

ruby 要素との併用について

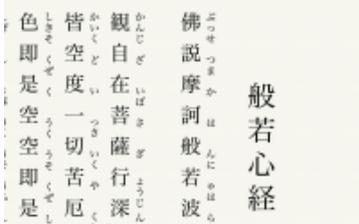
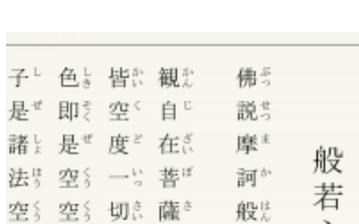
ruby 要素を利用した縦書コンテンツは、ウェブブラウザによっては、本文の列が左側にずれる。コンテンツ内でルビを振った縦書き文書を利用される場合は、画像化したテキストを使用すること。

今回の実証研究で使用されるウェブブラウザ別のルビふり縦書表示例

平成 26 年度クラウド等の最先端情報通信技術を活用した学習・教育システムに関する実証別冊
コンテンツ作成ガイドブック

android(Chrome)	iPad(Safari)	windows8(Internet Explorer10)
		<p>れているころでありま 、月のいい晩 ます。しずかな町のは 、ただひとり、窓 て、針しごとをしてい が、 に照らしていました。 ありましたから、</p>

今回の実証研究で使用されるウェブブラウザ別の SVG 縦書表示例

android(Chrome)	iPad(Safari)	windows8(Internet Explorer10)
		

SVG を用いた縦書き HTML : 般若心経

(<http://momdo.s35.xrea.com/web-html-test/vertical-text/svg-test.html>)

3.12.6 外部リンク

W3C Working Draft(CSS Writing Modes Module Level 3)(<http://www.w3.org/TR/css3-writing-modes/>)

以上