



OSC2017 Tokyo/Spring - LPI-Japan

HTML5プロフェッショナル認定試験 レベル1 ポイント解説無料セミナー

2017年3月



株式会社クレアコム
工藤 考司



クレアコムのご紹介



■会社概要

- ・ **設立**
2000年3月29日
- ・ **所在地**
神田
- ・ **事業内容**
IT技術者研修事業 / プロダクト企画・開発・販売事業
製造業向けコンサルティング事業 / プロジェクト支援事業

■教育/トレーニング事業

- ・ **ITスクール運営 (DTC : Developer Training Center)**
Java、PHP、HTML5を用いたWebアプリケーション開発技術の研修を提供

[教育スタイル]

オープンコース | E-Learning | オンサイト研修 | E-Learning + オンサイトフォローアップ
※特に例年4月開講の「JavaによるWebアプリ開発」の新人向けオープンコースは好評をいただいております。詳細は研修事業サイト (<http://www.devtc.com>) をご参照ください。

- ・ **IT技術研修支援受託**
お客様のご要望に応じたIT技術研修をオーダーメイドで企画・実施
研修コンテンツの作成、講師・研修会場アサイン・研修マシン手配まで、ご要望に応じて対応
- ・ **講師アサイン**
お客様の研修に開発経験豊富な講師・サブ講師をアサイン
- ・ **コンテンツ販売・E-Learningシステム販売**
弊社オリジナルのE-Learningシステム および 研修コンテンツを提供

■ 2014年10月に勧告されたHTMLのバージョン

- ・ 1999年勧告のHTML 4.01から実に15年ぶりのバージョンアップ！
- ・ 2016年11月にはpicture要素等を加えHTML5.1としてマイナーバージョンアップ

■ 広義では、非常に幅広い分野における技術拡張を示す

- ・ HTMLマークアップ要素の拡張
- ・ CSS(CSS3として定義)
- ・ JavaScriptのAPI
- ・ その他XMLやSVG (Scalable Vector Graphics)
WebGLといった関連技術

ブラウザをはじめとするWeb全般のプラットフォームに対する機能拡張である！

■ HTML5で拡張された技術要素は、 以下8つの分野にカテゴライズされている

1	グラフィック	2D/3Dグラフィックス、SVG等の描画関連技術
2	コネクティビティ	Web Socketによるサーバ⇄クライアント双方向通信などの技術
3	デバイスアクセス	デバイスが持っているGPSや加速度センサー等の各種センサーを取り扱う技術
4	マルチメディア	音声や動画等のメディアファイルの利用技術
5	オフラインストレージ	ブラウザ上のストレージやデータベースなどにデータを保持するための技術
6	パフォーマンス	Web Workersによるマルチスレッドプログラミングなどの高速化技術
7	セマンティクス	文書構造を定義するための新マークアップ要素
8	スタイリング(CSS3)	CSS3によるアニメーションやグラデーション等の表現機能



HTML5プロフェッショナル認定資格

LPI-Japanが、
HTML5, CSS3, Javascriptなど
最新のマークアップに関する技
術力と知識を、公平かつ厳正
に、中立的な立場で認定する
認定資格



HTML5プロフェッショナル認定制度は二つのレベル

※ 2017年3月1日に出題範囲を改定(Ver1.0→Ver2.0)。本内容はVer2.0となります。



HTML 5 Level.1

マルチデバイスに対応したWebコンテンツをHTML5を使って企画・制作できる。

対象		
Webデザイナー	グラフィックデザイナー	フロントエンドプログラマー
HTMLコーダー	Webディレクター	Webシステム開発者
スマートフォンアプリ開発者	サーバーサイドエンジニア	



HTML 5 Level.2

最新のAPIを駆使したWebアプリケーションを設計・開発できる。

対象		
Webデザイナー		フロントエンドプログラマー
HTMLコーダー	Webディレクター	Webシステム開発者
スマートフォンアプリ開発者	サーバーサイドエンジニア	



HTML5プロフェッショナル認定試験 レベル1 (Ver2.0)

※2017年3月1日に出題範囲を改定(Ver1.0→Ver2.0)。本内容はVer2.0となります。



HTML 5 Level.1

マルチデバイスに対応したWebコンテンツをHTML5を使って企画・制作できる。

対象		
Webデザイナー	グラフィックデザイナー	フロントエンドプログラマー
HTMLコーダー	Webディレクター	Webシステム開発者
スマートフォンアプリ開発者	サーバーサイドエンジニア	

出題範囲

Webの基礎知識	重要度
HTTP, HTTPSプロトコル	8
HTMLの書式	9
Web関連技術の概要	6
CSS3	
スタイルシートの基本	7
CSSデザイン	9
カスケード (優先順位)	2
要素	
要素と属性の意味 (セマンティクス)	10
メディア要素	6
インタラクティブ要素	7

レスポンシブWebデザイン

レスポンシブWebデザイン	重要度
マルチデバイス対応ページの作成	4
メディアクエリ	5
スマートフォンサイト最適化	3

APIの基礎知識

APIの基礎知識	重要度
マルチメディア・グラフィックス系API概要	5
デバイスアクセス系API概要	4
オフラインストレージ系API概要	8
通信系API概要	3



HTML5プロフェッショナル認定試験 レベル2 (Ver2.0)

※2017年3月1日に出題範囲を改定(Ver1.0→Ver2.0)。本内容はVer2.0となります。



HTML 5 Level.2

最新のAPIを駆使したWebアプリケーションを設計・開発できる。



出題範囲

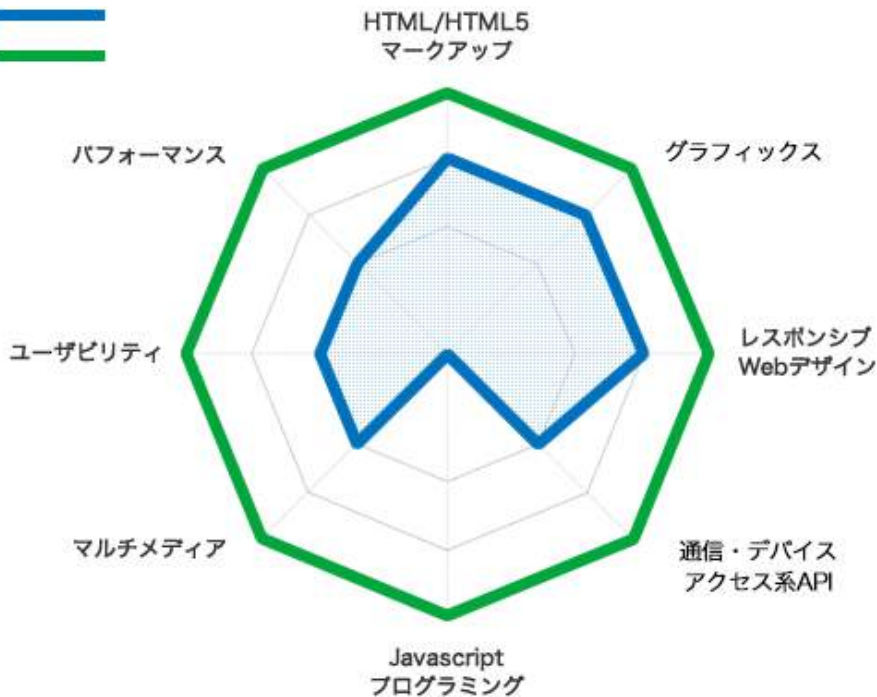
JavaScript	重要度
JavaScript文法	10
WebブラウザにおけるJavaScript API	
イベント	8
ドキュメントオブジェクト/DOM	6
ウィンドウオブジェクト	8
Selectors API	7
テスト・デバッグ	6
グラフィックス・アニメーション	
Canvas(2D)	8
SVG	2
Timing control for script-based animat	2
マルチメディア	
メディア要素のAPI	5
ストレージ	
Web Storage	7
Indexed Database API	
File API	5

通信	重要度
WebSocket	5
XMLHttpRequest	5
Server-Sent Events	1
デバイスアクセス	
Geolocation API	5
DeviceOrientation Event	1
パフォーマンスとオフライン	
Web Wokers	5
High Resolution Time	2
オフラインアプリケーションAPI	3
Page Visibility	2
Navigation Timing	1
セキュリティモデル	
クロスオリジン制約とCORS	4
セキュリティモデルとSSLの関係	4



レベル1とレベル2の資格体系 (Ver2.0)

Level.1 
Level.2 



試験実施形式

コンピュータベーステスト (CBT) で実施します。
マウスによる選択方式がほとんどですが、キーボード入力
問題も多少出題されます。

- ※ 試験は「ピアソンVUE」より配信されています。
- ※ 団体受験用にペーパーテスト (PBT) も実施します。

**HTML5プロフェッショナル認定試験
レベル1**

所要時間：90分 (アンケート等の時間を含む)
出題数：約60問
受験料：¥15,000 (税別)
認定条件：HTML5 レベル1試験の合格
認定の有効性の期限：5年間



**HTML5プロフェッショナル認定試験
レベル2**

所要時間：90分 (アンケート等の時間を含む)
試験問題数：40~45問
受験料：¥15,000 (税別)
認定条件：HTML5 レベル2試験に合格し
、
かつ有意なHTML5レベル1認定を保有し
ていること。
認定の有効性の期限：5年間



認定の証明

認定証



認定カード



認定者ロゴ(名刺用)



Level.1



Level.2

認定者ロゴは、認定後すぐに名刺等でご利用いただけます。

認定証・認定カードは、認定されてから2週間程度でご登録されたご住所にお届けしています。



本セミナーの解説内容

1.1 Webの基礎知識

- 1.1.1 HTTP, HTTPSプロトコル
- 1.1.2 HTMLの書式
- 1.1.3 Web関連技術の概要

1.2 CSS

- 1.2.1 スタイルシートの基本
- 1.2.2 CSSデザイン**
- 1.2.3 カスケード（優先順位）

1.3 要素

- 1.3.1 要素と属性の意味（セマンティクス）
- 1.3.2 メディア要素**
- 1.3.3 インタラクティブ要素**

1.4 レスポンシブWebデザイン

- 1.4.1 マルチデバイス対応ページの作成
- 1.4.2 メディアクエリ**
- 1.4.3 スマートフォンサイト最適化

1.5 APIの基礎知識

- 1.5.1 マルチメディア・グラフィックス系API概要
- 1.5.2 デバイスアクセス系API概要
- 1.5.3 オフラインストレージ系API概要
- 1.5.4 通信系API概要

！ピックアップ！
本セミナーでは、Level1の出題範囲の中から、実用的な内容をピックアップし解説します

HTML5によるバリデート (入力検証)

■ HTML5で加わったフォーム部品

※ `<input type="xxx">`のxxxに該当

フォーム部品	説明
search	入力時右側にクリア用×ボタンがついた検索テキスト入力ボックスを表示
tel	電話番号入力ボックスを表示
url	submit時、デフォルトでURL形式のチェックをするURL入力ボックスを表示
email	submit時、デフォルトでメールアドレス形式のチェックをするメールアドレス入力ボックスを表示
datetime	協定世界時による日時入力ボックスを表示
date	日時入力ボックスを表示。カレンダーダイアログによる入力。日付形式以外入力できません。
month	月入力ボックスを表示
week	週入力ボックスを表示
time	時間入力ボックスを表示
number	数値入力ボックスを表示。右側のスピンボタンを押すと値が増減。数字以外入力できません。
range	マウスで操作できるレンジバーを表示
color	色入力ボックスを表示。色選択ダイアログによる入力

※ ブラウザにより対応状況が異なります

■フォーム部品の「url」「email」は、使用するだけでバリデートの機能が働く

URL: type[url]	<input type="text" value="適当な値"/>
メールアドレス: type[email]	<input type="text" value="samp@san"/>
日付: type[date]	<input type="text" value="2016/07/2"/>
日: type[month]	<input type="text" value="2016年07月"/>

! URL を入力してください。

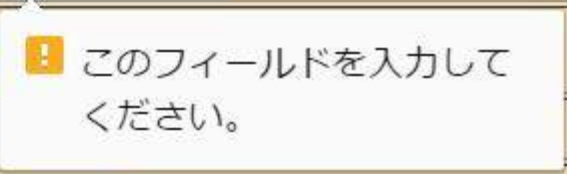
メール: [type=email]	<input type="text" value="sample.co.jp"/>
値必須: [required]	<input type="text" value="OK"/>
5文字以上10文字以下: [minlength=5 maxlength=10]	<input type="text" value="ABCDEFG"/>
20以上: [type=number min=20]	<input type="text" value="100"/>
2015/12/01~2016/01/15:	<input type="text" value="2016/01/01"/>

! メールアドレスに「@」を挿入してください。
「sample.co.jp」内に「@」がありません。

フォームのサブミット時に、デフォルトで入力値の検証が実施され、結果がNGであればサブミットはキャンセルされ、正しい入力を促すメッセージが表示されます

■ 値の入力を必須にするための required属性

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

値必須:[required]	<input type="text"/>
5文字以上10文字以下: [minlength=5 maxlength=10]	ABCDEFGG 
20以上:[type=number min=20]	100
0015 /10 /01 ~ 0016 /01 /15	

未入力の場合、サブミット時にエラーメッセージが表示され、サブミットがキャンセルされます

■ 入力文字数を制限する minlength属性とmaxlength属性

```
<input type="text" minlength="5" maxlength="10" >
```

5文字以上10文字以下: [minlength=5 maxlength=10]	ABC
20以上:[type=number min=20]	100
2015/12/01~2016/01/15: [type=date min=2015-12-01 max=2016-01-15]	2016/01/0
半角英字のみ:[pattern="^[a-zA-Z]+\$"]	XYZ

! このテキストは 5 文字以上で指定してください（現在は 3 文字です）。

入力文字数がminlengthに指定の文字数未満、またはmaxlengthに指定の文字数を超える場合、サブミット時にエラーメッセージが表示され、サブミットがキャンセルされます

■ 数値や日付の範囲を制限する min属性とmax属性

```
<input type="number" min="20 " max="100" >
```

```
<input type="date" min="2015-12-01 " max="2016-01-15" >
```

20以上:[type=number min=20]	5	
2015/12/01~2016/01/15: [type=date min=2015-12-01 max=2016-01-15]	2016/01/31	
半角英字のみ:[pattern="^[a-zA-Z]+\$"]	XYZ	
2015/12/01~2016/01/15: [type=date min=2015-12-01 max=2016-01-15]	2016/01/31	
半角英字のみ:[pattern="^[a-zA-Z]+\$"]	XYZ	
入力値同一か確認1	
入力値同一か確認2	

入力値がminに指定の最小値未満、またはmaxに指定の最大値を超える場合、サブミット時にエラーメッセージが表示され、サブミットがキャンセルされます

■ 入力形式を正規表現で指定する pattern属性

```
<input type="text" pattern="^[a-zA-Z]+$"  
title="半角英字のみ！">
```

半角英字のみ:[pattern="^[a-zA-Z]+\$"]	<input type="text" value="ああ"/>
入力値同一か確認1	<input type="text" value="....."/>
入力値同一か確認2	<input type="text" value="....."/>
スペルチェック	<input type="text"/>

! 指定されている形式で入力してください。
半角英字のみ!

入力した値の形式が指定の正規表現に適合しない場合、サブミット時にエラーメッセージが表示され、サブミットがキャンセルされます。

pattern属性に違反した場合は、title属性に指定したテキストがエラーメッセージに利用されます。

■ setCustomValidityによる独自バリデーション

```
<input type="text" id="user" onchange="checkValue();">
```

[JavaScript]

```
function checkValue() {  
    document.querySelector("#user").setCustomValidity("");  
    if (document.querySelector("#user").value == "taro"){  
        document.querySelector("#user").setCustomValidity("taroは既に存在  
しているよ!");  
    }  
}
```

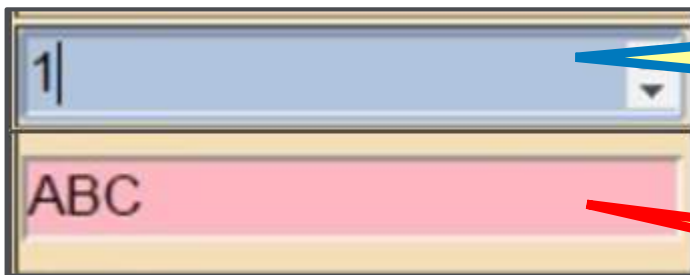


setCustomValidityによりエラーメッセージがセットされている場合、サブミット時にエラーメッセージが表示され、サブミットがキャンセルされます。

※ querySelectorは、HTML5で加わった要素を選択するためのJavaScript APIです

■ バリデートの状況を示すCSS3擬似クラス「:valid」「:invalid」

```
input:valid {  
    background-color: lightsteelblue;  
}  
input:invalid {  
    background-color: lightpink;  
}
```



「valid=検証OK」状態なら背景色が薄い青

「invalid=検証NG」状態なら背景色がピンク

■「:valid」「:invalid」以外のバリデート状態を示す CSS3擬似クラス

:in-range	min、max属性で指定した範囲内である状態
:out-of-range	min、max属性で指定した範囲外である状態
:required	必須入力（required）が設定されている状態
:optional	必須入力（required）が設定されていない状態



マルチメディア

■ マルチメディア関連のHTML5マークアップ要素

canvas 要素	図形描画する際に使用。グラフや2D/3Dグラフィックスなどの描画が可能。
audio 要素	音声を再生する際に使用
video 要素	動画を再生する際に使用
source 要素	動画や音声のURLなどを指定する際に使用。video要素またはaudio要素の子要素として記述。source要素を複数記述することで、フォーマットの異なる複数のメディアファイルを指定することが可能。
track 要素	動画や音声に字幕や頭だし用のチャプターを設定するためのタグ。video要素またはaudio要素の子要素として記述。
embed 要素	外部アプリケーションやFlashなどのプラグインを必要とするデータをHTML文書に埋め込む際に使用

■ audio要素で扱える音声形式

2016年8月時点

形式	Chrome	IE	Safari	Firefox	Opera
MP3	○	○	○	○	×
Ogg	○	×	×	○	○
AAC	○	○	○	×	×
WAVE	○	×	○	○	×

1.MP3:

MPEGが開発した音声フォーマット規格「MPEG-1 Audio レイヤー3」。拡張子は「.mp3」。歴史が古く汎用性があり一般に普及している代表的な音声ファイル形式。特許制限がある。

2.Ogg:

Xiph.Org Foundationが規格化した音声フォーマット。拡張子は「.ogg」。コーデックはOgg Vorbisが有名。オープンソースでライセンスフリー、またmp3よりも高音質。

3.AAC:

ISO 13818-7として標準化され、MPEG-2およびMPEG-4で採用されている音声ファイル形式。拡張子は「.m4a」。iTunesで使用されており、mp3よりも高音質。

4.WAVE:

MicrosoftとIBMにより開発された音声フォーマット。拡張子は「.wav」。Windowsで一般的に使われ普及した。最も高品質だが、非圧縮形式で容量が大きい。特許制限がある。

video要素で扱える動画形式

2016年8月時点

形式	Chrome	IE	Safari	Firefox	Opera
MP4 H.264	○	○	○	○	×
Ogg Vorbis	○	×	×	○	○
WebM	○	×	×	○	○

1.MP4 H.264:

MPEG-4フォーマットとともに利用される高品質な動画コーデック。一般に広く普及している。拡張子は「.mp4」。特許制限がある。

2.Ogg Vorbis:

Ogg Theoraとともにオープンソースでライセンスフリーのコーデック。拡張子は「.ogv」。

3.WebM:

Google社が開発したオープンソースでライセンスフリー、高品質な動画フォーマット。拡張子は「.webm」。VP8という動画コーデックを使用。

■ audio要素



↑ **controls**属性で再生コントロールを表示

ブラウザは、sourceタグを記述順で上から順に確認し、**type**属性に書かれたMIMEタイプの音声フォーマットの再生が可能であれば、リソースとしてそのファイルを読み込みする

```
<audio controls>
  <source src="audio/sample.ogg" type="audio/ogg">
  <source src="audio/sample.mp3" type="audio/mp3">
  <source src="audio/sample.wav" type="audio/wav">
  <p>※お使いのブラウザはaudioタグをサポートしてません</p>
</audio>
```

■ video要素



controls属性で再生コントロールを表示

ブラウザは、sourceタグを記述順で上から順に確認し、**type**属性に書かれたMIMEタイプ並びにコーデックの動画フォーマットの再生が可能であれば、リソースとしてそのファイルを読み込みます

<video controls>

```
<source src="video/rapiro.webm" type="video/webm;codecs='vp8,vorbis'">
```

```
<source src="video/rapiro.mp4" type="video/mp4;codecs='avc1.42E01E'">
```

```
<source src="video/rapiro.ogv" type="video/ogg;codecs='theora, vorbis'">
```

<p>※お使いのブラウザはvideoタグをサポートしてません</p>

</audio>

! Audio要素もvideo要素も、基本的な属性や記法は同じ!

■preload属性

再生前に指定の音声または動画ファイルを事前に読み込むか否かの属性。事前に読み込んでおけば、再生ボタン押下時の再生動作がスムーズになる一方で、ユーザーが動画を再生しない場合も通信上の負荷がかかることとなります。

```
<audio src="音声のURL" preload="下記指定値"></audio>  
<video src="動画のURL" preload="下記指定値"></video>
```

指定値	内容
none	事前の読み込みを行わず、再生時に初めて読み込みする
metadata	再生時間、トラックリスト、横/縦サイズなどのメタデータのみ事前に取得する
auto	スマートフォンのブラウザ以外は、音声または動画ファイルを事前に読み込みする ※省略時のブラウザの規定値(ブラウザにより異なる場合もある)

■ autoplay属性

Webページを読み込んだ時点で、指定の音声または動画を自動で再生する

```
<audio src="音声のURL" autoplay></audio>  
<video src="動画のURL" autoplay></video>
```

※ 省略時は、自動再生は行われません

■ loop属性

自動的に再生を繰り返す

```
<audio src="音声のURL" loop></audio>  
<video src="動画のURL" loop></video>
```

※ 省略時は、ループ再生は行われません

■ muted属性

ミュート(消音)状態で再生する

```
<audio src="音声のURL" muted></audio>  
<video src="動画のURL" muted></video>
```

※ 省略時は、ミュート再生は行われません

■ mediagroup属性

メディアリソースをグループ化する。グループ化したリソースはブラウザによって同期再生されます。グループ名が同じaudioまたはvideo要素が同じグループと見なされます。

```
<audio src="音声のURL" controls mediagroup="任意の名前"></audio>  
<video src="動画のURL" mediagroup="任意の名前"></video>
```

「音声と動画」「動画と動画」などを同期して再生することにより、解説音声と手話動画、教材動画と講師解説動画といった連動して再生する必要があるコンテンツへの対応が可能です。

■ poster属性

動画の再生前、またはブラウザが動画フォーマットの再生をサポートしていない場合に表示しておく画像のURLを指定します。



※ videoタグでのみ使用可能

```
<video src="動画のURL" poster="表示画像のURL"></video>
```

■ embed要素

プラグインを利用するコンテンツを組み込む

```
<embed src="組み込むコンテンツのURL" width="コンテンツの幅"  
height="コンテンツの高さ" type="コンテンツのMIMEタイプ"  
pluginspage="プラグインのダウンロード先URL">
```

以下のように、Flashコンテンツを指定したembedタグをvideoタグの子要素にすれば、videoタグ未対応の古いブラウザでもFlash Playerプラグインで動画を再生できます。この場合、videoタグに対応しているブラウザはembedタグを無視します。

```
<video controls>  
  <source src="video/rapiro.webm" type="video/webm;codecs='vp8,vorbis'">  
  <source src="video/rapiro.mp4" type="video/mp4;codecs='avc1.42E01E'">  
  <embed src="video/rapiro.swf" width="320px" height="300px"  
    type="application/x-shockwave-flash"  
    pluginspage="https://get.adobe.com/jp/flashplayer/">  
</video>
```


■ track要素

動画や音声に字幕や頭だし用のチャプターを設定する

```
<video src="video/rapiro.webm" controls>
  <track src="video/tracksample.vtt" kind="subtitles"
    srclang="ja" label="字幕" default>
</video>
```



src	トラック用ファイルのURL
kind	トラックの種類 ※ 次のいずれかを指定 subtitles : 字幕 (初期値) captions : 難聴用の字幕 descriptions : 音声合成用のテキスト chapters : 頭出し用のタイトル metadata : スクリプト用のデータ
srclang	トラックファイルの言語 (ja、en、zh、等)
label	トラックの名前 (「CCボタン」で切り替える際に使用される)
default	複数指定したトラックのうちの既定トラックに指定



■ track要素に使用するトラック用ファイル「テキストトラック」 TTML(拡張子「.ttml」)とWebVTT(拡張子「.vtt」)の2種類

TTML	XMLで記述するファイル
WebVTT	時間帯とテキストを並べて書くだけ簡単なテキスト形式のファイル

・WebVTTのサンプル

WEBVTT

00:00.000 --> 00:05.000 line:80%
□ロボットのrapiro君です！

00:51.000 --> 00:55.000
おっ！障害物が！！

00:56.500 --> 01:14.500 line:80%
回避ですー(^^)！

01:37.000 --> 01:42.000
ではバイバイ！

- 文字コードはUTF-8
- ファイルの冒頭は WEBVTT
- 空白行で区切り
- 始まりの時間と終わりの時間を「 --> 」でつなぐ
- 字幕の既定の表示位置をline、position、alignを指定することにより調整可能

■ JavaScriptによるaudioやvideoの再生

```
<audio id=" audioControl">  
<button onClick="playAudio('audio/sample.mp3')">視聴</button>
```

[JavaScript]

```
function playAudio(audioURL){  
    var audioObj = document.querySelector("#audioControl");  
    audioObj.src = audioURL;  
    audioObj.load();  
    audioObj.play();  
};
```

1. audioまたはvideoのメディア要素を取得
2. メディア要素のsrcプロパティに再生メディアのURLを設定
3. メディア要素のloadメソッドで音声ファイルの読み込みを実行
4. メディア要素のplayメソッドで再生を実行

■ audio/video要素のURLに再生範囲を指定する

```
src="メディアのURL#t=[再生開始時間][,再生終了時間]"
```

- ・ 50秒時点から最後まで再生

```
src="audio/sample.mp3#t=50"
```

- ・ 15秒時点から 40秒時点までの範囲を再生

```
src="audio/sample.mp3#t=15,40"
```

※ 再生範囲の指定により、音声や動画にもCSS Spriteのようなテクニック(1つの大きなメディアファイルを読み込み、部分的に再生する)を使用し、高速化を図ることができます。



CSS3

擬似クラスと擬似要素

■バリデート関連以外のCSS3で追加された擬似クラス

<code>:checked</code>	対象の要素がチェックされた状態 チェックボックスからラジオボタンで使用可能
<code>:not()</code>	当該要素が指定したセレクトクに該当しない（否定） 例えば、チェックされていない状態を指定する場合「 <code>:not(:checked)</code> 」
<code>:empty</code>	対象の要素が空の(テキストを含めた下位要素が一切存在しない)状態
<code>:enabled</code>	対象の要素が使用可能である状態
<code>:disabled</code>	対象の要素が使用不可の状態
<code>:first-of-type</code>	親要素にとって一番初めに現れた指定の要素
<code>:last-of-type</code>	親要素にとって一番最後に現れた指定の要素
<code>:nth-child()</code>	子要素として先頭から○番目の要素
<code>:nth-of-type()</code>	子要素として先頭から○番目の指定要素
<code>:nth-last-child()</code>	子要素として末尾から○番目の要素
<code>:nth-last-of-type()</code>	子要素として末尾から○番目の指定要素
<code>:only-child</code>	親要素にとって1つだけ存在している子要素
<code>:only-of-type</code>	親要素にとって1つだけ存在している指定要素
<code>:root</code>	文書の最上位要素(ルート要素) HTMLの場合、常にhtml要素

■ 擬似要素とは

- ・「要素の前後」や「最初の1文字」、「最初の1行」のように要素全体ではなく、その一部分を対象とする
- ・対象の部分に擬似的にタグを埋め込んだような効果や影響をあたえる

■ 擬似要素はコロンを2つ「::」記述するよう変更 (CSS3)

::first-letter	対象の要素の1文字目
::first-line	対象の要素の1行目
::before	対象の要素の直前
::after	対象の要素の直後
::marker	対象の要素のリストマーカ リスト系の要素のマーカ（先頭のマーク）となる部分

■ 擬似クラス/擬似要素を使用したサンプル

```
tr:nth-child(odd){
    background:lightblue;
}
tr:nth-child(even){
    background:bisque;
}
tr>td:nth-child(2)::first-letter{
    font-size:32px;
    color:red;
}
input[type="checkbox"]:checked~label{
    color:red;
}
input[type="checkbox"]:not(:checked)~label{
    color:silver;
}
```

■ Webコンテンツ HTML5 コース

【第1回】	終了	9月5日～9月6日	2日間	定員10名	<input checked="" type="checkbox"/> 終了
【第2回】	募集中	10月3日～10月4日	2日間	定員10名	<input type="checkbox"/> 終了
【第3回】	終了	11月1日～11月2日	2日間	定員5名	<input checked="" type="checkbox"/> 終了
【第4回】	募集中	12月14日～12月15日	2日間	定員5名	<input type="checkbox"/> 終了
【第5回】	募集中	1月4日～1月5日	2日間	定員15名	<input type="checkbox"/> 終了

■ nth-of-type()擬似クラスとnth-child()擬似クラス

数えてn番目にあたる要素を選択するためのセレクタ

※ 「nth-」という接頭値は、1th、2th、3th...の数値の部分をもnに置き換えたもの

セレクタ:nth-of-type(引数)

※セレクタの要素だけがカウントの対象

セレクタ:nth-child(引数)

※セレクタを含む全要素がカウントの対象

・引数

数値	数えて指定の数値番目の要素を対象とします。
間隔数値n 間隔数値n+数値 間隔数値n-数値	指定の間隔数値毎の要素を対象とします。例えば、「2n」とした場合、2番目、4番目、6番目...といったように偶数の要素が対象になります。「2n-1」とした場合、先の内容から1を引いた要素であるため、1番目、3番目、5番目...といったように奇数の要素が対象になります。
even	「2n」と同じで偶数の要素が対象になります。
odd	「2n-1」と同じで奇数の要素が対象になります。

■:checked擬似クラス

チェックされている状態を示すセレクトタ

■:not()擬似クラス

()の中に指定したセレクトタを否定するセレクトタ

・使用サンプル

1. `input[type="checkbox"]:checked~label`
2. `input[type="checkbox"]:not(:checked)~label`

※ 「**[]**(ブラケット)」は、属性値に条件を指定して抽出する**属性セレクトタ**です。
「**~**(チルダ)」は、ある要素と同じ階層にある要素を対象とする**兄弟セレクトタ**です。
どちらも、**CSS3**で追加されました。

1は、チェック状態のチェックボックスと同じ階層にあるラベル要素

2は、未チェック状態のチェックボックスと同じ階層にあるラベル要素
をターゲットにします

■ ::first-letter擬似要素

選択した要素の先頭文字を対象にするセレクタ

・使用サンプル

```
tr>td:nth-child(2)::first-letter
```

:nth-child(2)で、表2列目のフィールドをターゲットにし、
::first-letterで、該当フィールドテキストの先頭文字を対象にしています

※ 擬似クラスと擬似要素を併用する場合は、
擬似クラスが前で、擬似要素を後ろに記述する必要があります

■ ::before 模擬要素を使用したサンプル

・HTML

```
<h4>HTML5概要</h4>
<ul>
  <li>HTML5とは</li>
  <li>HTML5形式のドキュメント</li>
  <li>HTML5で追加・廃止・変更された要素</li>
</ul>
<h4>HTML5要素</h4>
<ul>
  <li>構造化タグ</li>
  <li>HTML5のフォーム部品</li>
  <li>バリデートの属性</li>
  <li>マルチメディア</li>
</ul>
<h4>CSS3</h4>
<ul>
  <li>CSSによるエフェクト効果</li>
  <li>CSS3アニメーション</li>
  <li>Media Queries</li>
  <li>模擬クラスと模擬要素</li>
</ul>
```

・CSS

```
h4{
  counter-increment:chapter;
}
h4::before {
  content:"第" counter(chapter) "章";
  color:blue;
  text-shadow:1px 1px 1px black;
}
ul{
  list-style-type:none;
  counter-reset:section;
}
li::before {
  counter-increment:section;
  content:"§" counter(section) ".";
  border-radius:10px;
}
```

・表示イメージ

第1章 HTML5概要

- §1. HTML5とは
- §2. HTML5形式のドキュメント
- §3. HTML5で追加・廃止・変更された要素

第2章 HTML5要素

- §1. 構造化タグ
- §2. HTML5のフォーム部品
- §3. バリデートの属性
- §4. マルチメディア

第3章 CSS3

- §1. CSSによるエフェクト効果
- §2. CSS3アニメーション
- §3. Media Queries
- §4. 模擬クラスと模擬要素

■ ::before/ ::after 擬似要素

セレクタ指定要素の前後にテキストや画像などを挿入する擬似要素

contentプロパティにテキストや画像へのURLを指定することで、セレクタで指定した要素の前後に、指定したコンテンツを指定のレイアウトで一括して付け加える。

元のHTML文書に記述されていない文字や記号等をcontentプロパティを使用して補助的に表示させたい場合に使用する。

- ・文字列を表示する場合

```
要素へのセレクタ::before { content:"任意の文字列"; }
```

- ・画像を表示する場合

```
要素へのセレクタ::before { content:url(画像のURL); }
```

- ・対象要素の属性値を表示する場合

```
要素へのセレクタ::before { content:attr(属性値名); }
```

- ・カウンターを表示する場合

```
要素へのセレクタ::before { counter-increment:任意のカウンター名;  
content:counter(上記のカウンター名); }
```

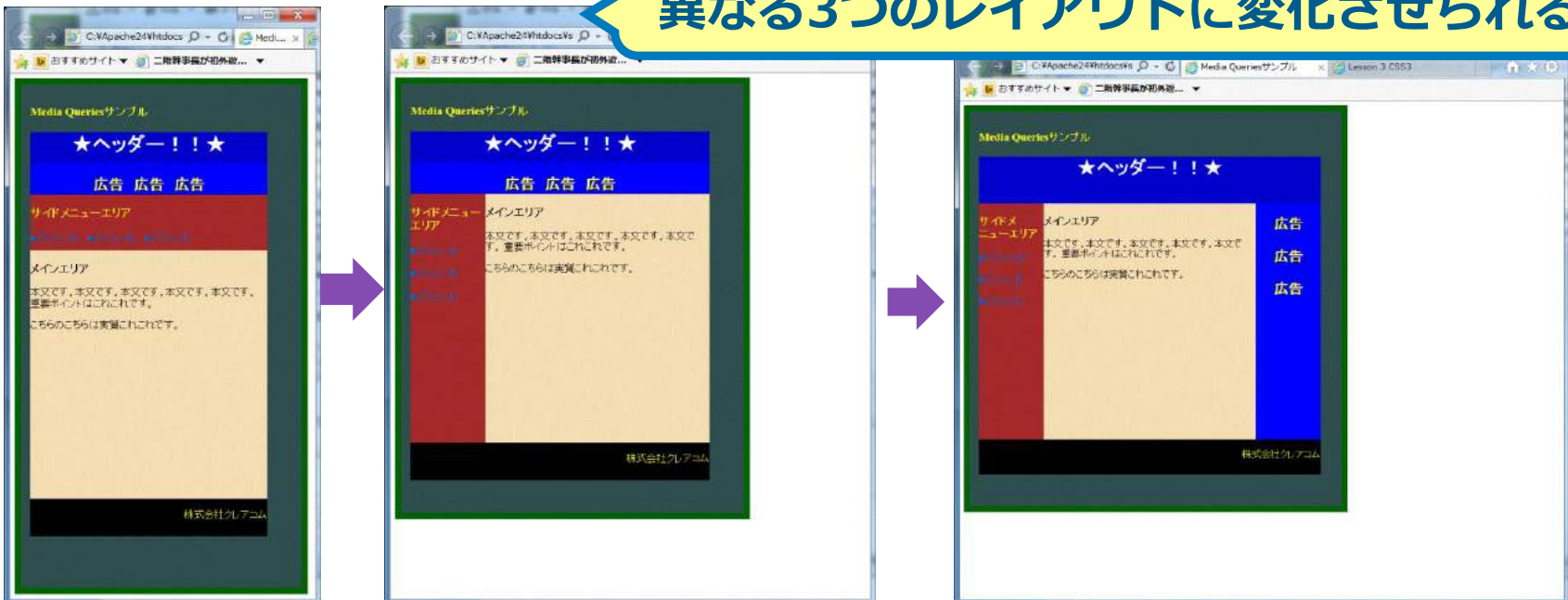


メディアクエリ

Media Queriesとは

- ・「Media Types」(ディスプレイやプリンタ等の出力先のデバイスに応じてレイアウトを分岐する仕組み)を拡張したもの
- ・出力先のデバイスの違いだけでなく、例えば同じディスプレイというデバイスの中で、さらに画面の解像度やサイズに応じて細かくレイアウトを振り分けすることができる

例えば、ブラウザのサイズ(幅)に応じて、異なる3つのレイアウトに変化させられる



Media Queriesの基本的な記述方法

メディアタイプ and (属性 : 値)

- ・上記構文を「,(カンマ)」で区切って複数並べ、**OR**条件を指定できる
- ・(属性 : 値)の部分を「and」で区切って複数並べ、**AND**条件を指定できる
- ・先頭に否定を表す「not」を指定できる
- ・Media Queries対応ブラウザのみに**限定**する「only」を指定できる

メディアタイプ

screen	ディスプレイを表す
print	印刷を表す
projection	プロジェクタを表す
tv	テレビを表す
handheld	モバイル機器を表す
all	全てのメディアを表す

属性

width	ウィンドウの横幅
height	ウィンドウの高さ
device-width	物理的なデバイスの横幅
device-height	物理的なデバイスの高さ
resolution	デバイスの解像度 (ピクセル密度)
aspect-ratio	ウィンドウのアスペクト比
device-aspect-ratio	物理的なデバイスのアスペクト比
orientation	画面が横長か縦長か
color	デバイス1pxのRGBビット数

※属性について、orientation以外は「以上」を表す“min-”、「以下」を表す“max-”の前置詞を付けることが可能

■スタイルシート内で記述

```
@media screen and (min-width:320px){  
    画面の横幅が320px以上の場合のスタイルの定義  
}  
  
@media screen and (min-width:768px) and (max-width:1024px) ){  
    画面の横幅が768px以上、1024px以下の場合のスタイルの定義  
}  
  
@media screen and (min-width:1024px) ){  
    画面の横幅が1024px以上の場合のスタイルの定義  
}
```

※ 上から順番に真偽が評価され、最後に真となったブロックのスタイルが適用されます

■ linkタグで指定

例えば、320px以上用として「min320.css」、768px以上用「min768.css」、1024px以上用「min1024.css」といったように、環境毎にスタイルシートのファイルを用意

```
<link rel="stylesheet" media="screen and (min-width:320px)" href="min320.css">
```

```
<link rel="stylesheet" media="screen and (min-width:768px) and ( max-width:1024px)" href="min768.css">
```

```
<link rel="stylesheet" media="screen and (min-width:1024px)" href="min1024.css">
```

※ media属性に指定したMedia Queriesの条件に合致しているスタイルシートのファイルが読み込まれます



最後に！ クレアコムでのHTML5関連 研修コースのご紹介



DTC HTML5関連コースのご紹介

HTML5 既存技術コース

※ HTML5勧告以前から
変わらない技術要素

Webコンテンツ Basic (2日)

1日目	1	Webの仕組み
	2	HTML
	3	CSS (Cascading Style Sheets)
2日目	3	CSS (Cascading Style Sheets)
	4	JavaScript
	5	HTML、CSS、JavaScript組み合わせ演習

JavaScript実践

- DOM、Ajax、jQueryによる
Webリッチクライアント技術 - (3日)

1日目	1	Webリッチクライアント技術概要
	2	JavaScript基礎
2日目	3	JavaScriptによるDOMと画面制御
	4	オブジェクトとJSON
3日目	5	Ajax
	6	jQuery

HTML5 新技術コース

Webコンテンツ HTML5 (2日)

1日目	1	HTML5とは
	2	HTML5形式のドキュメント
	3	HTML5で追加されたタグ
	4	HTML5で廃止されたタグ
	5	HTML5で変更されたタグ
	6	構造化タグ
	7	HTML5のフォーム部品
	8	バリデートの属性
	9	マルチメディアタグ
	10	HTML Media Capture
	11	CSS3によるエフェクト効果
2日目	12	CSS3 アニメーション
	13	Media Queries
	13	CSS3の模擬クラス
	14	HTML5タグとCSS3を使用した総合演習



HTML5 試験対策コース

HTML5プロフェッショナル認定
試験レベル1対策コース (2日)

1日目	1	Webの基礎知識
	2	HTML
	3	HTML5の新しい要素と属性
	4	HTML5で廃止・変更された要素と属性
	5	スタイルシートの基本1
2日目	6	スタイルシートの基本2
	7	CSSプロパティ
	8	カスケード
	9	マルチデバイス対応
	10	オフラインWebアプリケーション

お問合せ

株式会社クレアコム デベロッパー・トレーニングセンター
〒101-0047 東京都千代田区内神田2-6-9
翔和鎌倉橋ビル2F
TEL:03-3251-2226 FAX:03-3251-2227
URL <http://www.devtc.com>
E-Mail devtc@creacom.co.jp



LPI-JAPAN HTML5 Professional Certification

Open the Future with **HTML5**.