



LPI Japan 主催

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4月

講師：伊藤 眞

(株式会社クリーク・アンド・リバー認定講師)



本日の内容

1. HTML5プロフェッショナル認定試験の概要

- 1-1. HTML5 試験とは
- 1-2. HTML5 資格を取得する意義
- 1-3. HTML5 レベル1の範囲の概要(Ver2.0)
- 1-4. 受験に向けた学習方法

2. 項目解説

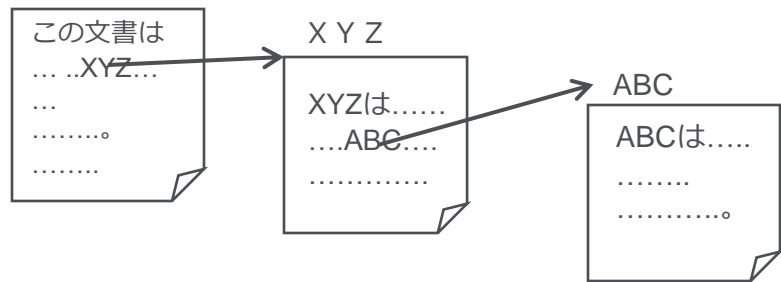
- 2-1. 1.2.3 CSS カスケード
- 2-2. 1.4レスポンシブWebデザイン
- 2-3. 1.5.1 マルチメディア・グラフィックス系API概要
(Ver2.0の新規項目)

1. HTML5プロフェッショナル認定試験の概要

1-1. HTML5 試験とは

Hyper Text Markup Language

複数の文書を相互に関連付け、結び付ける仕組み。hyperというのは「テキストを超える」という意味



- World Wide Webはハイパーテキストの実装のひとつと言える。
- HTTPは「Hyper Text Transfer Protocol」の略。

視覚表現や文章構造などを記述するための言語

HTML5 の例

```
<!DOCTYPE HTML>

<h1>これはレベル1の見出しです</h1>
<body> ここが文書本体です
      :
</body>
```

その他のマークアップ言語の例

- SGML,
- TeX (数式など)
- XML

“Markup”という語は英語圏の出版業界で著者、編集者、印刷者の中で指示を伝える方法を意味していた



**HTML4.01 : 文書閲覧
(1999年12月勧告)**



**HTML5 : アプリケーション実行環境
(2014年10月勧告)**



HTML5プロフェッショナル認定資格とは

- 次世代のWebプロフェッショナルのスキルの向上に貢献するために、HTML5を活用したWebページやWebアプリケーションなどのデザイン、設計、構築に関する体系だった知識とスキルを備えたHTML5のプロフェッショナルを中立的な立場で公平かつ厳正に認定する資格制度です。
- Webデザイナー、Webプログラマー、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などの、Web開発プロジェクトやWebサービスに関わるあらゆるプロフェッショナルが対象です。
- 多くの企業が推進する次世代Web言語の認定資格として、HTML5のプロフェッショナルのスキルの向上に役立ちます。また、企業内や研修機関での『技術力を担保する客観的基準』としても活用できます。



はじめに：試験の出題範囲の改定について

- 2017年3月1日より出題範囲が改訂されました。
- **出題範囲改訂の背景とメリット**
 - 試験の提供から約3年が経過し、その間HTML5を取り巻く環境が変化した。
 - 適用分野の拡大
 - 部分技術の陳腐化
 - 新技術の盛り込み
- **レベル1では出題範囲の一部を以下の概要にてアップデートした。**
 - サーバー関連技術に関する出題を減らした。
 - ⇒クライアント系のみのお仕事をされる方に配慮した内容とした。
 - HTML5で何が出来るかの知識やWebアプリの知識を問う質問を増やした。
 - ⇒認定者の対象として、Webディレクターをより意識した内容とした。
- **移行措置**
 - 新バージョン(Ver2.0) リリースの6ヶ月間は旧バージョンを選択可能。
(但しこれから受ける方はVer2.0で準備することが良いと思われます。)



二つのレベル



HTML5 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・制作ができる。



HTML5 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。

・グラフィックデザイナーは Level.2 の対象外。

対象

Webデザイナー

HTMLコーダー

フロントエンドプログラマー

スマートフォンアプリ開発者

Webシステム開発者

サーバーサイドエンジニア

Webディレクター

グラフィックデザイナー

対象

Webデザイナー

HTMLコーダー

フロントエンドプログラマー

スマートフォンアプリ開発者

Webシステム開発者

サーバーサイドエンジニア

Webディレクター



レベル1とレベル2の資格体系

ベーシックレベル
HTML5プロフェッショナル向け

所要時間：90分
試験問題数：約60問
受験料：¥15,000（税抜）
認定条件：HTML5 レベル1試験に合格すること
認定の有意性の期限：5年間



アドバンストレベル
HTML5プロフェッショナル向け

所要時間：90分
試験問題数：約40-45問
受験料：¥15,000（税抜）
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。
認定の有意性の期限：5年間

認定名：HTML5 Level1 (Markup Professional)
試験名：HTML5 Level1 Exam

この資格の認定者は、下記のスキルと知識を持つWebプロフェッショナルであることを証明できます。

- HTML5を使ってWebコンテンツを作成することができる。
- ユーザー**体験を考慮した**WEBコンテンツを設計・作成することができる。
- スマートフォンや**組み込み機器**など、ブラウザが利用可能な様々なデバイスに対応したコンテンツを制作できる。
- HTML5で何ができるか、こういった技術を使うべきかの広範囲の基礎知識を有する。**

認定名：HTML5 Level2 (Application Development Professional)
試験名：HTML5 Level2 Exam

この資格の認定者は、下記のスキルと知識を持つWebプロフェッショナルであることを証明できます。

- 動的に動作させて高いユーザビリティを実現するリッチユーザーインターフェイスアプリケーションを作成することができる。
- マルチデバイスに対応し高パフォーマンスで動作する動的コンテンツを作成することができる。
- システム間連携を行いリアルタイムな情報を提供するアプリケーションを作成することができる。
- APIのセキュリティモデルを理解したうえで開発することが出来る。**

試験方式

- コンピューターベーステスト (CBT)
- 受験会場・受験日時は、全国各地から好きな会場や日時を自由に選択して受験できる。
- 申し込みは ピアソン VUEへ
 - <https://www.pearsonvue.co.jp/Clients/HTML5.aspx>
- 出題方式はマウスによる選択方式がほとんど。キーボード入力問題も多少出題される。
- 出題形式は以下の3つ
 - **知識問題**：問題を読んで解答を複数選択肢の中から1つないし複数選ぶ。
 - **記述問題**：キーボードで解答を入力する。
 - **コードリーディング問題**：示されたコードを読み解いて複数選択肢の中から1つないし複数の解答を選ぶ。
- 試験時間は90分だが機密保持契約を結ぶ時間も込みなので正味90分ではない。
- 合否は終了と同時にわかる。

認定証

※サンプルのため
斜線が入っています。



認定カード



認定者ロゴ (名刺用)



1.HTML5プロフェッショナル認定試験の概要

1-2. HTML5 資格を取得する意義



今 HTML5プロフェッショナル認定試験を取得する意義

試験範囲外

- Web資格の中で最も求められている理由
- いまIT系エンジニアに最も人気のある資格とは何だろうか？
- スマートフォンアプリの開発方法の変化
- 組み込み機器へのWeb技術の浸透

既にHTML5は次世代Web標準として多くの分野で活用されています。

HTML5は、PC/タブレット端末、スマートフォン、電子書籍端末、スマートテレビ、カーナビなどの車載システム、ゲーム機器など、多くのデバイスに対応しています。

このようなワンソース・マルチユース/マルチデバイスの対応だけでなく、企業向けのWebアプリやモバイルアプリの開発、ゲームの開発、電子書籍コンテンツの制作など、HTML5が活用される分野はどんどん広がっています。



LPI Japan ウェブサイトより引用



いまIT系エンジニアに最も人気のある資格とは何だろうか？

試験範囲外

アイティメディア株式会社が2015年9月～11月に行った調査の結果を分析し紹介する。Webエンジニアとしての価値を高めるヒントになるはずだ。

@ITで告知したWebアンケート

調査期間 2015/9/29-2015/11/2 有効回答数 1,064件

今回の調査では、「今後取得したいベンダーニュートラル資格」で「HTML5プロフェッショナル認定試験」が1位、LPICが2位となった。現在必要なスキル分野というだけでなく、今後伸び続けるスキル分野として、「HTML5プロフェッショナル認定試験」が高く評価されていることの証といえる。

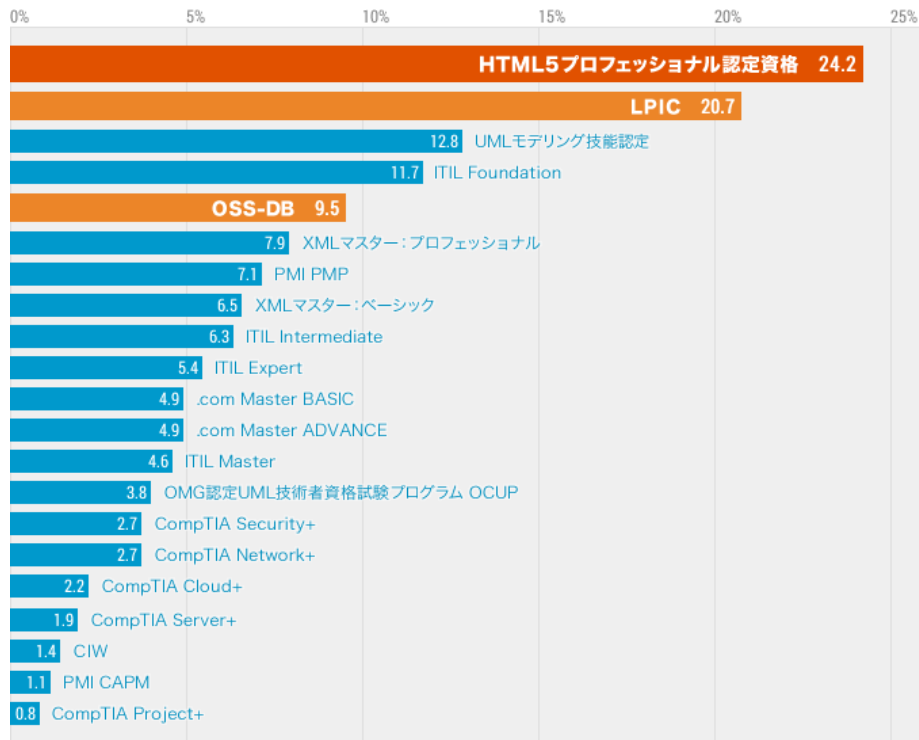
1位：HTML5プロフェッショナル認定試験（24.2%）

2位：LPIC（20.7%）

3位：UMLモデリング技能認定（12.8%）

4位：ITIL Foundation（11.7%）

5位：OSS-DB技術者認定試験（9.5%）



LPI Japan ウェブサイトより引用

アプリのタイプ	ネイティブ	ハイブリッドアプリ	ウェブアプリ
言語	Objective-C , Swift Java, C# 等の OO 言語	HTML/CSS/JavaScript	HTML/CSS/JavaScript
開発ツール	Xcode (iOS) Android Studio (Android) Eclipse (Java) Visual Studio (Windows他)	ウェブアプリのものに加え Cordova(PhoneGap), Monaca, appian, Intel XDX などの ビルドツール。	テキストエディター、ブラウザ、 各種ライブラリー、 専用開発ツール： YEOMAN, Visual Studio, WebStorm Dreamweaver など
メリット	ハードウェアの機能を最大に 使用できる。	Webのスキルで開発できる	幅広いハードウェアに対応できる。 Webのスキルで開発できる。
デメリット	エンジニアの確保が最も困難。 多くの機種に対応するための 工数が増大する。 配布にプラットフォームの承認 が必要。	最終的にネイティブアプリを作る がネイティブ程の自由度はなく 性能も通常はネイティブより 劣る	ブラウザの機能を使用するためデバイスの 機能を全て利用できない。 パフォーマンスがネイティブより劣る。 コードがユーザーに見える。

近年はネットワークの高速化とハードウェアの高速化、ブラウザによる実装の充実によりウェブアプリ、ハイブリッドアプリの優位性が増大してきている。

*製品名等は例示のためのもので全てではなく、また推奨を意味するものではありません。

- 組み込み機器とはPCやサーバー、スマートフォンなど、一般にコンピューターの形態をしていない機器でマイコンなどのコンピューターを内蔵している物の総称。最近は自動車、家電、工作機器、航空機などあらゆる機器にCPUが使用されている。
- 最近のTVは既にWeb端末になっている。
- IoT (Internet of Things) デバイスにもブラウザを搭載するものが出てきた。

1.HTML5プロフェッショナル認定試験の概要

1-3. HTML5 レベル1の範囲の概要(Ver2.0)

この章では出題構成の各項目を概観します。受験対策セミナーではありませんので全ての項目に言及するわけではありませんのでご理解ください。

レベル1(Ver2.0)の出題構成

主題	項目	重要度*
1.1 Webの基礎知識	1.1.1 HTTP,HTTPSプロトコル	8
	1.1.2 HTMLの書式	9
	1.1.3 Web関連技術の概要	6
1.2 CSS	1.2.1 スタイルシートの基本	7
	1.2.2 CSSデザイン	9
	1.2.3 カスケード（優先順位）	2
1.3 要素	1.3.1 要素と属性の意味（セマンティックス）	10
	1.3.2 メディア要素	6
	1.3.3 インタラクティブ要素	7
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	4
	1.4.2 メディアクエリ	5
	1.4.3 スマートフォンサイト最適化	3
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	5
	1.5.2 デバイスアクセス系API概要	4
	1.5.3 オフラインストレージ系API概要	8
	1.5.4 通信系API概要	3

変更
変更
変更

新規
新規
変更
新規

*重要度：試験における各分野における重要度の相対値で、およその問題比率となります。

1.1.1 HTTP,HTTPSプロトコル

重要度 8

知識問題

記述問題

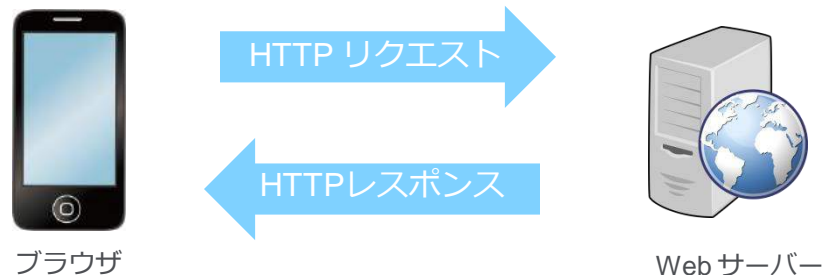
望まれるスキル (概要)

- HTTPおよびHTTPSプロトコルに関する知識を有している。
- ブラウザでアクセスした時に返ってくるエラーコードの意味を理解できて、問題を解決するヒントとする事ができる。

主要な知識範囲 (一部)

- 通信内容や手順
- HTTPリクエストにおけるメソッド
- リクエストURI
- レスポンスのヘッダ項目およびステータス
- HTTPプロトコルに規定されている認証方式

HTTP プロトコルによる通信の概要



主なメソッドの種類

GET	指定されたリソースを取得する
POST	サーバーにデータを送信する
DELETE	指定したリソースを削除する

主なステータスコードと意味

200	OK
400	Bad Request
403	Forbidden
404	Not Found

重要度 9

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

- HTMLの仕様に沿った書式でHTMLコードを記述する事ができる。

主要な知識範囲 (一部)

- HTMLバージョン情報を含む文書型宣言に関する記述方法
- 要件に合わせた言語コードと、文字コード (符号化方式)
- HTMLで使用可能な文字参照の記述方法
- ヘッダ内の記述方法(メタ情報、外部リソースの指定)

HTML5のDOCTYPE 宣言

```
<!DOCTYPE html>
```

HTML5以前の例

[HTML 4.01 Transitional]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

文字コードの設定の例

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title> HTML5 レベル 1 試験へようこそ</title>
```

```
</head>
```

1.1.3 Web関連技術の概要

重要度 6

知識問題

記述問題

望まれるスキル (概要)

- 動的なWebコンテンツのための技術の知識。
- SEO

主要な知識範囲 (一部)

- Webコンテンツを作成する際に使うスクリプト言語や画像ファイル、規格の概要
- Webに関する、セキュリティ脅威に関する技術の概要
- Web関連技術の概要

画像ファイルフォーマットの例

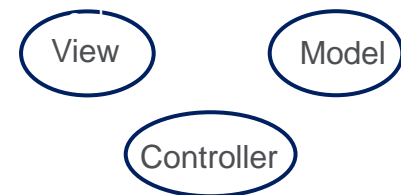
BMP	ビットマップ、無圧縮で保存される
PNG	フルカラー、可逆圧縮方式、半透明可
JPEG	フルカラー、非可逆圧縮方式
GIF	256色以下の画像、可逆圧縮形式

セキュリティ脅威の例

SQLインジェクション: WebアプリケーションのSQL文に不正なSQL文を注入する。

Web関連技術の例

MVC アーキテクチャー: ソフトウェア設計方法のひとつでプログラムを右のMVCの3個に分割しプログラムの構造をわかり易くする手法。



知識問題

HTML5に準拠したHTML文書に記述するDOCTYPE宣言として以下の中から正しいものを選びなさい。

- A. `<!DOCTYPE html >`
- B. `<!DOCTYPE html>`
- C. `< !DOCTYPE html>`
- D. `<!doctype HTML>`
- E. `<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">`

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。

1.2.1 スタイルシートの基本

重要度 7

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

- 見やすく効率的なコードを記述し、複数のページで共有させるために必要な記述を適切に行うことができる。

主要な知識範囲 (一部)

- HTMLから利用したいスタイルシートを指定する際の記述方法
- スタイルシートに正しいセクタを指定しプロパティ値を指定する際の記述方法

セクタとは：

どのHTML要素にCSSのスタイルを適用させるか指示するもの。

style.cssでは

pがセクタ、colorがプロパティ、redが値。

link 要素：CSSファイルを読み込む

example.html

```
<!DOCTYPE html>
<html>
<head>
<title>example</title>
<link rel="stylesheet"
href="style.css">
</head>
<body>
<p>CSS example</p>
</body>
</html>
```

style.css

```
p {
  color: red;
}
```

style 要素：HTML内にCSSを記述

example_style.html

```
<!DOCTYPE html>
<html>
<head>
<title>example</title>
<style>
  p {
    color: red;
  }
</style>
</head>
<body>
<p>CSS example</p>
</body>
</html>
```


重要度 9

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

- 要件に沿ったデザインをCSSを利用して実現する際に、どのような実現方法が適切か判断でき、仕様に沿った正しいコードを記述する事ができる。

主要な知識範囲 (一部)

- レイアウトに関する記述方法
- 色、背景、罫線に関する記述方法
- テキスト、リスト、テーブルに関する記述方法
- 変形、アニメーションに関する記述方法

・レイアウト、ボックスモデルの例

```

<div id="box">
  <p> ボックスモデル </p>
</div>

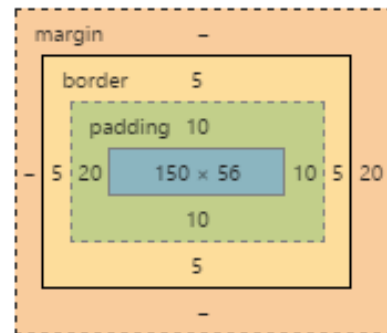
<style>
  div#box {
    width: 150px;
    padding: 10px 10px 10px 20px;
    border: 5px solid gray;
    margin: 0;
    margin-right: 20px;
    color: red;
  }
</style>

```

ブラウザでの表示



ブラウザの開発ツールの表示例



1.2.3 カスケード（優先順位）

重要度 2

知識問題

コードリーディング問題

望まれるスキル (概要)

- HTMLの一つの要素に対して複数の記述が対象となる事象（プロパティの衝突）が発生した場合における適用の優先順位を理解している。

主要な知識範囲 (一部)

- 外部・内部スタイルシート、インラインスタイルシートの指定に関して記述場所による優先順位
- セレクタの優先順位に関する計算方法

CSSは**Cascading Style Sheets**の頭文字をとったもの。HTMLの文脈では文書の親子構造に従ってスタイルシートの定義が段階的に引き継がれていく仕組み。

CSSスタイルの優先ルール	
スタイルの指定方法	重み
!important	最重要
インラインスタイル	1.0.0.0
id	0.1.0.0
クラス、属性セレクタ、疑似クラス	0.0.1.0
要素、疑似要素	0.0.0.1

- インラインスタイルの例:
`<h2 style=color:blue;> レベル2のヘッダは青</h2>`

重要度 10

知識問題

コードリーティング問題

記述問題

望まれるスキル (概要)

- HTML要素や属性のセマンティックスを理解し、コンテンツの意味を解釈しながら適切なHTML要素や属性を使ってHTMLコーディングができる。

主要な知識範囲 (一部)

- HTML5.1仕様で規定されたHTML要素と属性の意味
- セクションの概念

要素 : 実際のWebサイトの中味になるもの。

html要素:HTML文書全体を表す。

```
<html></html>
```

script要素: JavaScriptなどのスクリプトを組み込むためのもの。

```
<script src="myjsfile.js"></script>
```

など多数。

属性 : 要素に付加的な情報を持たせるもの。

例) ボタンを作成する際にボタンの種類を指定する場合。

```
<p> <button type="submit"> 送信 </button> </p>
```

送信

- グローバル属性とはどの要素にも使用できる属性。

セクション : 区分や区分けを意味するもの

<body>,<header>,<nav>,<section>,<article>,<aside>,<address>,<h1>~<h6> 等で形成される。

重要度 6

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

- ビデオやオーディオをHTMLコンテンツとして適切に活用できる。

主要な知識範囲 (一部)

- ビデオ再生のためのHTMLマークアップ知識
- オーディオ再生のためのHTMLマークアップ知識
- ビデオファイルとオーディオファイルの知識
- 字幕表示のためのHTMLマークアップ知識

video 要素:

ビデオやオーディオファイルを読み込み再生する。

audio 要素:

音楽ファイルを再生する。

source 要素:

ビデオやオーディオファイルの種類、場所を指定。
video要素またはaudio要素の子要素。

例)

```
<video controls>
  <source src="HTML5ビデオ.mp4" type="video/mp4">
  <source src="HTML5ビデオ.webm" type="video/webm">
  <p>お使いのブラウザはHTML5ビデオを再生できません。</p>
</video>
```

注)controls は再生コントロールを表示する属性。この場合ブラウザが再生できるフォーマットを見つけた時点で再生が始まりvideo要素から抜ける。

字幕表示: track 要素で字幕やキャプションが表示可

重要度 7

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

- ユーザーの操作を伴うHTML要素を効果的に活用できる。

主要な知識範囲 (一部)

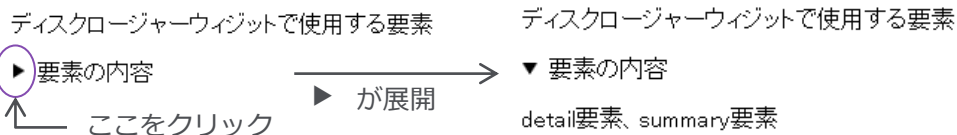
- ハイパーリンク
- フォーム
- フレーム
- コンテキストメニュー
- ディスクロージャーウィジット
- コマンドメニュー

ディスクロージャーウィジット

ユーザーに追加の詳細情報を提供する

```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <title> インタラクティブ要素 </title>
</head>
<p> ディスクロージャーウィジットで使用する要素 </p>

<details>
  <summary>要素の内容</summary>
  <p>detail要素、summary要素</p>
</details>
</html>
```



記述問題

下記のHTMLファイルでp要素の色を赤にしたい。 に入る要素をそれぞれ記述しなさい。

```
<!DOCTYPE html>
<html>
<head>
<title>example</title>

  p {
    color: red;
  }

</head>
<body>
<p>CSS example</p>
</body>
</html>
```

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。

コードリーディング問題

右記で示されている id名がboxの div要素の幅を選びなさい。

- A. 150px
- B. 175px
- C. 180px
- D. 200px
- E. 220px

```
<div id="box">
  <p> ボックスモデル </p>
</div>

<style>
  div#box {
    width: 150px;
    padding: 20px 20px 20px 20px;
    border: 5px solid gray;
    margin: 0;
    margin-right: 20px;
    color: red;
  }
</style>
```

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。

1.4.1 マルチデバイス対応ページの作成

重要度 4

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

- どのような画面サイズであってもデザインが仕様どおりになるようなページの実現方法を理解している。
- マルチデバイス対応のページを作ることができる。

主要な知識範囲 (一部)

- マルチデバイス対応ページを作成する際に考慮すべき点
- レスポンシブWebデザインにおけるページ構成要素の配置手法
- レスポンシブWebデザインにおける画像の扱い

フルードデザイン

画面の幅の比率で要素を記述する手法

```
<p class="one-fourth">フルードデザインされたスタイル</p>
<p>フルードデザインされないスタイル </p>

<style>
.one-fourth {
    width: 25%;
    background-color: silver; border: 1px; solid red;
}
</style>
```

Viewport

スマートフォンなどの機器のために画面幅を指定する機能。

```
<meta name="viewport" content="width=device-width">
```


1.4.2 メディアクエリ

重要度 5

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

- メディアクエリを利用して、画面サイズなどの様々な環境に合わせて表示を変えるページを作成することができる。

主要な知識範囲 (一部)

- メディアタイプ、メディア特性を指定し、正しい構文で条件毎に適用するスタイルを切り替える場合の記述
- エラーが発生した際の、エラーハンドリングに関する記述

メディアクエリ

メディアの特性を基に表示方法を変化させる手法

```
<p> スクリーン幅が600ピクセル以下になると文字が赤くなる。 </p>

<style>
  @media screen and (max-width: 600px) {
    p {color:red}
  }
</style>
```

メディアタイプの例

screen	一般的なディスプレイ
braille	点字ディスプレイなど
print	印刷用のプリンタ
tv	テレビでの表示用

1.4.3 スマートフォンサイト最適化

重要度 3

知識問題

コードリーディング問題

記述問題

望まれるスキル (概要)

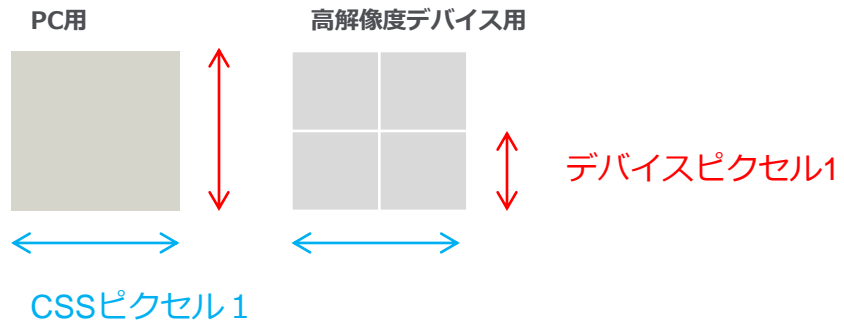
- スマートフォンでWebコンテンツを閲覧する時を考慮し、回線速度などスマートフォン特有の環境でも快適に閲覧することができるコンテンツを設計・開発することができる。

主要な知識範囲 (一部)

- スマートフォンにおける画像の扱い
- スマートフォン特有の設定について、注意すべき点
- スマートフォンにて閲覧する際のパフォーマンス向上技術

デバイス・ピクセル比

PC用のウェブページ：デバイスピクセル=CSSピクセル
高解像度のスマートフォンではデバイス：CSSが2:1になるものがある。



重要度 5

知識問題

記述問題

新規

望まれるスキル (概要)

- ビデオやオーディオの活用するための知識を有し、具体的に何ができるのかを理解している。
- 適切なグラフィックスを扱うための基礎知識を有し、適切な技術を選択できる。
- ビデオとグラフィックスを組み合わせ何ができるのかを理解している。

主要な知識範囲 (一部)

- メディア関連要素のAPIが提供する機能
- ストリーミング
- DRM
- ビットマップグラフィックスの知識
- ベクターグラフィックスの知識

MPEG-DASH (Dynamic Adaptive Streaming over HTTP)

- HTTP を使った動画配信プロトコルの規格。
- 様々な機器に対して、動画が途切れないように帯域などの環境に応じてビットレートを切り替え、最適な高品質ビデオストリームを提供する。
- 2012年4月(ISO/IEC 23001-6) ,2014年5月(ISO/IEC 23009-1:2014)
- 従来技術として下記があるが互換性がない。

Apple: HTTPライブストリーミング (HLS)
Microsoft: スムースストリーミング (SS)
Adobe: HTTPダイナミックストリーミング (HDS)

DRM (Digital Rights Management)

- デジタルコンテンツの利用や複製を制限するために設けられている仕組みの総称。

重要度 4

知識問題

記述問題

望まれるスキル (概要)

- スマートフォンやパソコンに備え付けられているセンサーなどのデバイスに関する技術を理解し、それらをJavaScriptからAPIを使って何ができるのかを理解している。

主要な知識範囲 (一部)

- 位置情報
- 加速度センサー
- ジャイロ
- 入力デバイス

Geolocation API: ユーザーの位置情報を扱うための A P I

メソッド	機能概要
getCurrentPosition()	現在の位置情報を 1 回だけ取得する。
watchPosition()	位置情報を定期的 to 取得する。
clearWatch()	位置情報の監視をクリアする。

- 取得できる位置情報には、緯度、経度、高度、緯度経度の精度、高度の精度、方角、速度がある。

Touch Events : タッチ面への接触状態の変化発生するイベント

touchstart	タッチ面がさわられた
touchmove	タッチ面がドラッグされた
touchend	タッチ面から離れた

重要度 8

知識問題

記述問題

望まれるスキル (概要)

- JavaScriptからデータをブラウザー内に保存する仕組み、オフラインアプリケーション、最新のバックグラウンドによる処理の仕組みを理解し、Webアプリケーションで何が可能になるのかを理解している。

主要な知識範囲 (一部)

- データストレージの仕組み
- オフラインアプリケーションに必要な知識
- バックグラウンド処理に関する知識

Webブラウザのデータストレージの種類

名称	主な特徴
Cookie	一つあたり4KB、文字列のみ
Web Storage	ローカルストレージは2MB~4MB、セッションストレージは2MB~文字列のみ
Indexed Database	JavaScriptのオブジェクトをそのまま保存できる。容量は無制限。

アプリケーションキャッシュ

- オフラインウェブアプリケーションを実現するための仕組み。マニフェストファイルを使って制御する。

Web Workers

- ウェブアプリケーションでバックグラウンドの処理を行わせるための仕組み。

重要度 3

知識問題

記述問題

望まれるスキル (概要)

- JavaScriptからさまざまな通信プロトコルを使ってクラウドと通信する仕組みと特性を理解し、利用シーンに応じて適切なAPI選択ができる。

主要な知識範囲 (一部)

- AJAX
- 双方向データリアルタイム通信
- サーバープッシュ

XMLHttpRequest

- クライアントがサーバーとデータをやり取りすることを可能にするAPI。ページの全てを送信しなくても部分的にデータをやり取りしてページの表示を変えることができる。

WebSocket API

- ブラウザとサーバ間でインタラクティブな通信セッションを開くことを可能にするAPI。サーバとクライアントが一度コネクションを行った後は、必要な通信を全てそのコネクション上で専用のプロトコルを用いて行う。

Server-Sent Events

- サーバー側で何らかの出来事が起きたことを、リアルタイムでクライアント側に知らせることを可能にする技術。HTTPを使う。

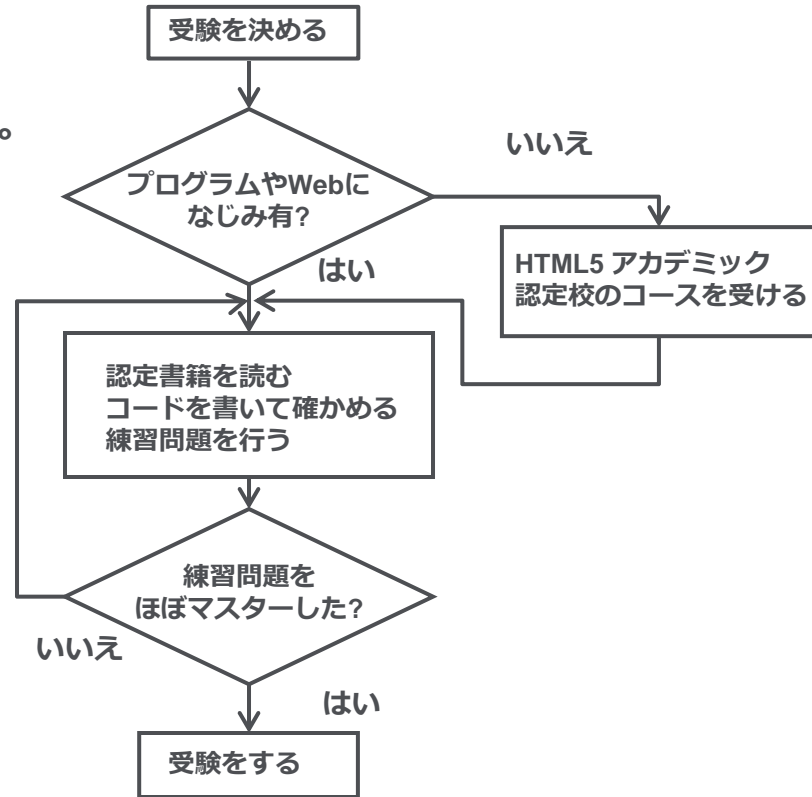
1.HTML5プロフェッショナル認定試験の概要

1-4. 受験に向けた学習方法

試験のための勉強には下記のようなものがあります

- LPI-Japan HTML5 アカデミック認定校のコースを受ける。
- LPI-Japan 認定書籍を読んで演習問題を解く。
- 自分で実際にコードを書いて表示や動作を確認する。
- 公開されている試験範囲を理解する。

- 過去問は非公開のため、HTML/CSSに詳しい方でも試験の傾向をつかむために認定書籍は必須です。
- 試験なので暗記はやはり必要。これも手を動かすことが近道。
- 合格ラインは約70%のため満点を取る必要はありませんがわからない項目を残したままだと70%をとるのは難しいと思われます。
- 公開されている試験範囲の項目は理解しておくべき。





LPI-Japan HTML5アカデミック認定校

<http://html5exam.jp/measures/learning.html>

• LPI Japan 認定 書籍

<http://html5exam.jp/measures/textbook.html>



HTML5
プロフェッショナル認定資格
レベル1 教科書

NTTソフトウェア株式会社 [著]



LPI-Japan HTML5 認定教材

「HTML5 プロフェッショナル認定試験レベル1」の
出題範囲を、体系的・体系的にまとめた教科書!

【Contents】

1. Web の基礎知識
 2. HTML5 の要素
 3. CSS3
 4. マルチデバイス対応
 5. オフライン Web アプリケーション
- 想定問題集



1. HTML/CSSを記述するテキストエディタ

直接HTML/CSSコードをタイプできるもの。

- [Visual Studio Code \(マイクロソフト\)](#)
- Atom <https://atom.io/>
- Text その他、多数
- emacs
- vi

2. HTML/CSSを実行するブラウザ

開発ツールのモードにすると挙動がより分かり易くなります。

ブラウザ	開発者ツールの呼び出し方
Chrome	ページ内で右クリック->検証。またはF12キー
Firefox	ページ内で右クリック->要素を調査。またはF12キー
Safari	メニューバーから「開発」->「webインスペクタを表示」 (メニューバーに開発がない場合はSafari->環境設定でメニューバーに「開発」メニューを表示をチェックする。)
IE	ページ内で右クリック->要素の検査。またはF12キー

*製品名等は例示のためのもので全てではなく、また推奨を意味するものではありません。



LPI Japanのサイトも活用しましょう

試験サイト

<http://www.html5exam.jp/>

試験範囲

http://html5exam.jp/outline/objectives_lv1_v2.html

サンプル問題

<http://html5exam.jp/measures/sample.html>

無料セミナー

<http://html5exam.jp/news/event/>

HTML5道場

<http://html5exam.jp/measures/dojo.html>



学習に役立つサイト

- Web 技術に関する情報、tutorialなど

MOZZILA DEVELOPER NETWORK <https://developer.mozilla.org/ja/>

- ブラウザーの実装状況がわかるサイト

<http://caniuse.com/>



2.項目解説

2-1. 1.2.3 CSSカスケード

CSSカスケードとは

- HTMLのDOM(ドキュメント・オブジェクト・モデル)は親子構造を持つ。
別の言葉で言うとHTML文書は階層化された木構造を持った要素の集合体。
カスケードという考え方は親に指定したCSSスタイルが段階的に子へ引き継がれていくこと。
したがって親で指定しておけばいちいち子の要素に指定する必要がなく便利。

例)

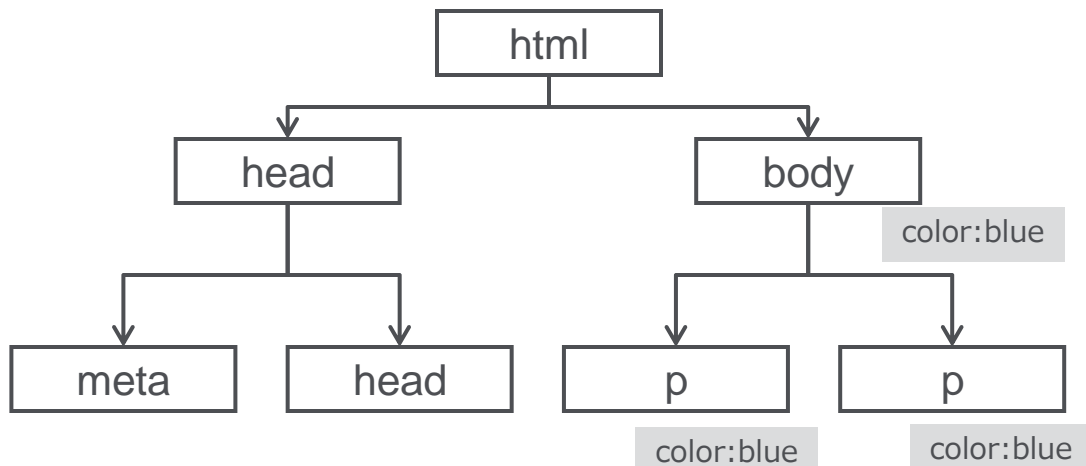
```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> CSSカスケード </title>
  </head>
  <body>
    <p> これからCSSカスケードの説明をします。
  </p>
    <p> p 要素にスタイルを定義しなくても上位の
body要素の指定が引き継がれます </p>
  </body>

  <style>
    body {
      color: blue;
    }
  </style>
</html>

```

左記のコードは以下のような木構造になる



前の例ではbodyという要素がCSSのセレクタに指定される。
これを要素セレクタと言い、そのHTML文書の該当する要素全てに適用される。

```
<style>
  body {
    color: blue;
  }
</style>
```


要素セレクトタ、別の例

```
<div>
  <p> 名前 <input type="text" > </p>
  <p> 住所 <input type="text" > </p>
  <p> 電話番号 <input type="tel" > </p>
</div>

<style>
  div{
    color:red;
  }
</style>
```

名前

住所

電話番号

左記のコードで電話番号だけ色を変えないようにするには。右のような方法もある。
これはdivとpの親子関係を利用した方法。

```
<div>
  <p> 名前 <input type="text" > </p>
  <p> 住所 <input type="text" > </p>
</div>
<p> 電話番号 <input type="tel" > </p>
<style>
  div p{
    color:red;
  }
</style>
```

名前

住所

電話番号

主要要素セレクタの例

名前	記法	機能
全数セレクタ	* (アスタリスク)	全てのHTML要素に対してスタイルが適用される。
子孫セレクタ	スペース区切り	例) h1 p の場合 h1を親に持つ全てのpに 適用される。
直近の子要素	>	子孫セレクタは親と子の間に何があっても良いがこの場合は直下にある子要素のみ。なお逆(<)はない。
隣接する兄弟要素	+	例) h3+pの場合、h3要素の次にあるp要素のみ適用される。
後続する兄弟要素	~	+の場合一つだけになるがこの場合は~以降の同じレベルの要素が全て選択される。

さきほどの電話番号の例ではDOMの木構造を変化させて色を変えている。
本来のCSSの目的はHTMLの構造とは別に表示を指定することにあるので避けるべき。

```
<div>
  <p> 名前 <input type="text" > </p>
  <p> 住所 <input type="text" > </p>
  <p class="telno_class"> 電話番号 <input type="tel" > </p>
</div>

<style>
  p.telno_class {
    color:blue;
  }
  div p{
    color:red;
  }
</style>
```

- その場合色を指定したい要素にクラス名を指定する方法がクラスによる指定。
- CSSではクラスの前に.(ピリオド)をつけてクラス名を書く。

名前

住所

電話番号

*先にp.telno_classが指定されているところにも注目。

クラスによる指定と似たなものにidセレクトタがある。

```
<div>
  <p> 名前 <input type="text" > </p>
  <p> 住所 <input type="text" > </p>
  <p id="home_telno"> 電話番号 <input type="tel" > </p>
</div>

<style>
  p#home_telno {
    color:green;
  }
  div p{
    color:red;
  }
</style>
```

- idの前に#をつけてid名を書く。

名前

住所

電話番号

クラスによる指定とidセレクトタとの違い (1/2)

クラスは同じHTML文書にいくつあっても良いがidは1つでなければならない。

クラスによる指定では、あるグループに同じクラス名を指定して共通のスタイルを適用できる。

```
<div>
  <p> 名前 <input type="text" > </p>
  <p> 住所 <input type="text" > </p>
  <p class="telno_class"> 自宅電話番号 <input type="tel" > </p>
  <p class="telno_class"> 携帯電話番号 <input type="tel" > </p>
  <p class="telno_class"> 会社電話番号 <input type="tel" > </p>
</div>

<style>
  p.telno_class {
    color:blue;
  }
  div p{
    color:red;
  }
</style>
```

名前

住所

自宅電話番号

携帯電話番号

会社電話番号

クラスによる指定とidセレクトタとの違い (2/2)

idは同一のHTML文書では1つでなければならない

```
<div>
  <p> 名前 <input type="text" > </p>
  <p> 住所 <input type="text" > </p>
  <p id="home_telno"> 自宅電話番号 <input type="tel" > </p>
  <p id="cel_telno"> 携帯電話番号 <input type="tel" > </p>
  <p id="office_telno"> 会社電話番号 <input type="tel" > </p>
</div>

<style>
  p#home_telno {color:green;}
  p#cel_telno {color:blueviolet}
  p#office_telno {color:grey}
  div p{
    color:red;
  }
</style>
```

名前

住所

自宅電話番号

携帯電話番号

会社電話番号

但し同じidを複数作成してしまってもブラウザで複数に適用されることが多い。しかしながら混乱の元になるのでそのような使い方はすべきではない。

属性セレクト (1/2)

一般的には上記のクラス、やidなど開発者が作成したものを使用してCSSを適用させる要素を指定するが、HTMLに定義されている属性を指定してCSSを適用させることが出来る。

```
<div>
  <p> お客様の情報を入力してください オレンジ色は必須入力です</p>
  <p title="mustfill"> 名前 <input type="text" required> </p>
  <p> 住所 <input type="text" > </p>
  <p> 自宅電話番号 <input type="tel" > </p>
  <p title="mustfill"> 携帯電話番号 <input type="tel" > </p>
  <p class="telno_class">会社電話番号 <input type="tel" > </p>
</div>

<style>
  p.telno_class {color:blue;}
  div p{ color:red; }
  p[title]{ color:orange; }
</style>
```

お客様の情報を入力してください オレンジ色は必須入力です

名前

住所

自宅電話番号

携帯電話番号

会社電話番号

*** この例ではtitle属性を持つp要素だけオレンジ色になる。**

属性セレクト (2/2)

前の例では「title」という属性をそのまま使用した。
その他にも以下のような属性の指定方法がある。

記法	機能	使用例
=	属性が特定の値に合致する場合に適用させる。	上記の例で p[title="mustfill"]など
^=	属性が指定の文字列で始まる要素のみ適用させる。	p[href^="http://"]でhttpのみ色を変えるなど
\$=	属性が指定の文字列で終わる要素のみ適用させる。	p[href\$=".pdf"]でPDFファイルのみ色を変えるなど
=	属性の値に指定の文字列が含まれている場合のみ適用させる。	p[href="LPI"]でLPIの文字の入ったリンクのみ色を変えるなど。

***これが全てではありません**

個別の要素の中で スタイルの指定を記述する方法。
適用されるのはその要素のみ。

```
<div>
  <p> お客様の情報を入力してください オレンジ色は必須入力です </p>
  <p> インラインスタイルシートは黒のボールドを指定しています </p>
  <p>
    <p title="mustfill"> 名前 <input type="text" required> </p>
    <p> 住所 <input type="text" > </p>
    <p> 自宅電話番号 <input type="tel" > </p>
    <p style="color:black; font-weight: bold" title="mustfill"> 携帯電話番号 <input
type="tel" > </p>
    <p class="telno_class">会社電話番号 <input type="tel" > </p>
  </div>

<style>
  p.telno_class {color:blue;}
  div p{ color:red; }
  p[title]{ color:orange; }
</style>
```

お客様の情報を入力してください オレンジ色は必須入力です

インラインスタイルシートは黒のボールドを指定しています

名前

住所

自宅電話番号

携帯電話番号

会社電話番号

最も優先度が高くなるのが!important。
全てに優先するのでなるべく使うべきではない。

```
<div>
  <p> お客様の情報を入力してください オレンジ色は必須入力です</p>
  <p title="mustfill"> 名前 <input type="text" required> </p>
  <p> 住所 <input type="text" > </p>
  <p> 自宅電話番号 <input type="tel" > </p>
  <p title="mustfill"> 携帯電話番号 <input type="tel" > </p>
  <p class="telno_class">会社電話番号 <input type="tel" > </p>
</div>

<style>
  p {color: black !important}
  p.telno_class {color:blue;}
  div p{ color:red; }
  p[title]{ color:orange; }
</style>
```

お客様の情報を入力してください オレンジ色は必須入力です

名前

住所

自宅電話番号

携帯電話番号

会社電話番号

疑似要素は、ドキュメントの特定のパーツをスタイル付けする。例えば `::after`
疑似要素は、セレクタで指定された要素の後ろにスタイルを追加できる。

```
例)
<div>
  <p>200</p>
</div>

<style>
  p::after{
    content: "円";
    font-size: 1.0em
  }
</style>
```

疑似要素の例 `::after`, `::before`, `::first-letter` など

疑似クラスは選択される要素に対して特定の状態を指定する。

例えば `:hover` 疑似クラスは、選択した要素上にマウスカーソルがあるときにスタイルを適用する。

```
<div>
  <p>マウスを乗せると色が赤に変わります。</p>
</div>

<style>
  p:hover{
    color: red;
  }
</style>
```

- この例ではp要素の上にマウスが来ると色が変わる。
- その他に
 - `:link`
 - `:visited`
 - `:active`
 - `:focus`
 - `:first-child`
 - `:last-child`
 - `:target`
 - `:enabled/deisabled`など多数。
試験範囲としては1.2.1 スタイルシートの基本の領域。

優先順位の必要性とそのルール

- 実際の開発では複数ファイル、複数のコーダーが存在する。
- 別の場所で自分の指定が上書きされたり、想定外の指定が引き継がれたりする。
- 優先度の理解は必要だが、そのうえで複雑にならないような仕掛けが重要。
- CSSの指定は最後に指定されたものが優先される上書きモデル。但し下記の優先ルールの重みの桁が変わるとそちらが優先される。

CSSスタイルの優先ルール	
スタイルの指定方法	重み
!important	最重要
インラインスタイル	1.0.0.0
id	0.1.0.0
クラス、属性セレクタ、疑似クラス	0.0.1.0
要素、疑似要素	0.0.0.1

一般論としては、指定の範囲が狭くなる方が優先度が高くなる。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> サンプル問題4 </title>
  </head>
  <div>
    <p> 名前 <input type="text" > </p>
    <p> 住所 <input type="text" > </p>
    <p class="telno_class"> 自宅電話番号 <input type="tel" > </p>
    <p class="telno_class"> 携帯電話番号 <input type="tel" > </p>
    <p class="telno_class" id="company"> 会社電話番号 <input type="tel" > </p>
  </div>
  <style>
    p#company{ color:green; }
    p.telno_class { color:blue; }
    div p{ color:red; }
  </style>
</html>
```

左記のHTML5のコードを実行した場合、「会社電話番号」は何色になるか正しいものを選びなさい。

- A.黒
- B.赤
- C.緑
- D.青

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> サンプル問題5 </title>
  </head>
  <div>
    <p> 名前 <input type="text" > </p>
    <p> 住所 <input type="text" > </p>
    <p class="telno_class"> 自宅電話番号 <input type="tel" > </p>
    <p class="telno_class"> 携帯電話番号 <input type="tel" > </p>
  </div>
  <h3>CSSカスケーダ</h3>
  <a> 本日は </a>
  <a> 2017/04/15</a>
  <style>
    p.telno_class { color:blue; }
    div p{ color:red; }
    h3+a{ color:green;}
  </style>
</html>
```

左記のHTML5のコードを実行した場合、
「2017/04/05」は何色になるか正しい
ものを選びなさい。

- A.黒
- B.赤
- C.緑
- D.青

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。



2.項目解説

2-2. 1.4レスポンスWebデザイン

レスポンシブWebデザインとは

もともとWebページはPCのブラウザーが中心であった。
携帯電話が出てくると携帯電話用のサイトとPC用とを別に作成していた。
スマートフォンではPC用のページも表示できるが全く同一というわけではない。

マルチデバイスに対応するためのWebページの制作方法

手法の概要

特徴

レスポンシブウェブデザイン

CSSのメディアクエリによってページのレイアウトを調整する。HTML/CSSファイルは同一。

同一のサイト名でデバイス毎のHTMLを振り分ける方法

サイトは同じだが別のHTML/CSSを作成する。

PC用とスマートフォン用に別のサイトを作成する方法

全く別のサイトを作成する。

・それぞれの方法にメリット・デメリットがあるが通常の業務用サイトではレスポンシブウェブデザインが主流になると考えられる。

レスポンシブWebデザインを実現するための技術要素としては以下のものがある。

- メディアクエリ
- フルードグリッド、
- フルードイメージ
- Viewport
- リセットCSS
- スマートフォンサイト最適化

- メディアクエリは、**メディアタイプ**と、**メディア特性**を利用してスタイルシートの適用条件を決定する一つ以上の式からなる。
- **メディア特性**には、メディアの幅と高さ、色数などがある。
- メディアクエリにより、コンテンツそのものを変更せずに、出力デバイスに合わせてコンテンツを表現することができる。

メディアタイプ

all	すべて	projection	プロジェクター
braille	点字用ディスプレイ	screen	PCなど通常のディスプレイ
embossed	点字用プリンター	speech	音声読み上げブラウザ
handheld	ハンドヘルド	tty	テキスト端末 (コマンドライン)
print	プリンタ	tv	テレビ

メディア特性には以下のものがある。min/maxとはmin-widthのように使用し、以上/以下を指定できる。

メディア特性	説明	min/maxの使用
color	出力デバイスのカラーコンポーネントあたりのビット数。カラー表示可能なものでない場合、この値は0。	可
color-index	出力デバイスのカラー検索テーブルのエントリ数を指示する。	可
aspect-ratio	出力デバイスの対象とする表示領域のアスペクト比を記述する。水平ピクセル数 / 垂直ピクセル数を記述する。	可
device-aspect-ratio	出力デバイスのアスペクト比を記述する。水平ピクセル数 / 垂直ピクセル数を記述する。	可
device-height	出力デバイスの高さを記述する。(描画領域ではなく、スクリーンや用紙全体を意味する)	可
device-width	出力デバイスの幅を記述する。(描画領域ではなく、スクリーンや用紙全体を意味する)	可

メディア特性	説明	min/maxの使用
height	出力デバイスの描画域の高さを記述する。(viewportの高さやプリンタのページ印刷領域の高さなど)	可
monochrome	モノクロ(グレースケール)デバイス上のピクセルあたりのビット数を指示する。デバイスがモノクローム表示でない場合、デバイスの値は0になる。	可
orientation	デバイスが横置き(landscape:高さより幅が広い表示)と縦置き(portrait:幅より高さが高い表示)のどちらのモードなのかを指示する。	不可
resolution	出力デバイスの解像度(ピクセル密度)を指示する。解像度は、1インチあたりのドット数(dpi)または1センチメートルあたりのドット数(dpcm)で指定される。	可
scan	テレビ出力デバイスの走査方式を記述する。(値: progressive interlace)	不可
width	出力デバイスの描画域の幅を記述する。(viewportの幅やプリンタのページ印刷領域の幅など)	可
grid	出力デバイスがグリッドデバイスとビットマップデバイスのどちらなのかを決定します。デバイスがグリッドベース(ttyターミナルや電話機のディスプレイなど表示部が一行ずつ)の場合、値は1になる。ビットマップデバイスの場合は0になる。	不可

外部のCSSファイルを条件によって振り分ける場合

HTMLファイル

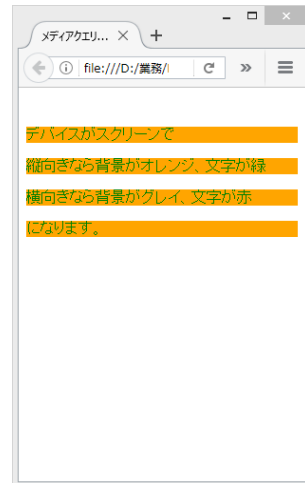
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> メディアクエリーCSSファイルの読み込み </title>
  </head>
  <link rel="stylesheet" media="screen and (orientation:portrait)"
href="portrait.css"/>
  <link rel="stylesheet" media="screen and (orientation:landscape)"
href="landscape.css"/>
  <div>
    <p> デバイスがスクリーンで </p>
    <p> 縦向きなら背景がオレンジ、文字が緑 </p>
    <p> 横向きなら背景がグレイ、文字が赤 </p>
    <p> になります。 </p>
  </div>
</html>
```

portrait.css

```
p {
  color: green;
  background-color: orange;
}
```

landscape.css

```
p {
  color:red;
  background-color: gray;
}
```



CSSファイル内で条件によって振り分ける場合

- 次の例ではviewport の幅が 500 ピクセルから 800 ピクセルの場合に使用されるスタイルシートを指定する。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> メディアクエリー同一ファイル </title>
  </head>
  <style>
    @media screen and (min-width: 500px) and (max-width:
800px) {
      p {color:red;}
    }
  </style>
  <div>
    <p> スクリーンの表示幅が500ピクセル以上、800ピクセル以下なら文字が赤になります。 </p>
  </div>
</html>
```

500 ピクセル未満

スクリーンの表示幅が500ピクセル以上、800ピクセル以下なら文字が赤になります。

500 ピクセル以上、800ピクセル以下

スクリーンの表示幅が500ピクセル以上、800ピクセル以下なら文字が赤になります。

800ピクセルより大

スクリーンの表示幅が500ピクセル以上、800ピクセル以下なら文字が赤になります。

フルードグリッド、フルードイメージ

- **フルードグリッド**とは、画面をグリッドデザインの手法を使ってデザインする際に、グリッドのサイズをピクセルなどの絶対値ではなく、%やem/remを用いて要素やテキストのサイズを調整する方法。
- **フルードイメージ**とは画像がどのようなスクリーンサイズのデバイスであってもそのデバイス内に収まるように比率を保持したまま拡大・縮小するように指定する方法。

```
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <style>
    div {display:flex;width:100%}
    div p:nth-child(1) {background-color:red;width:30%}
    div p:nth-child(2) {background-color:blue;width:30%}
    div p:nth-child(3) {background-color:yellow;width:40%}
    img {max-width: 50%}
  </style>
  <div>
    <p> Grid-1 </p> <p> Grid-2 </p> <p> Grid-3 </p>
  </div>
  
</html>
```



Viewport

- ブラウザは初期値で表示される画面の幅が決まっている。スマートフォンでは980pxが多い。しかし何も画面の表示に関して指定しないとPC用のサイトの大きさで見てしまう。
- Viewportの指定によりスマートフォンで最適な表示となるような設定が可能になる。
- HTMLのヘッダ要素内のmeta要素として以下の例のように書くとデバイスの幅に合わせて表示されるようになる。次ページのcontent の値の一覧を示している。

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

リセットCSS

- ブラウザはそれぞれに各タグの初期値を持っている。初期値が違う場合同じコードでも表示が異なる可能性がある。
- 最初に設定したい初期値を持つスタイルシートを適用し表示の違いをなくすことが可能になる。この最初に読み込ませるCSS ファイルのことをリセットCSSと呼ぶ。

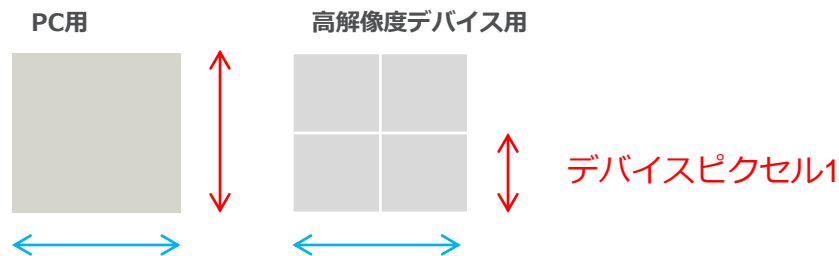
<meta name="viewport"> の content の値

値	設定可能な値	説明
width	正の整数またはリテラル device-width	viewport の幅をピクセル値で定義します。
height	正の整数またはリテラル device-height	viewport の高さをピクセル値で定義します。
initial-scale	0.0 から 10.0 までの、正の数値	デバイスの幅 (ポートレートモードでの device-width またはランドスケープモードでの device-height) と viewport のサイズの比率を定義します。
maximum-scale	0.0 から 10.0 までの、正の数値	ズームの最大値を定義します。この値は minimum-scale と同じまたはより大きくしなければなりません。そうしないときの動作は不定です。
minimum-scale	0.0 から 10.0 までの、正の数値	ズームの最小値を定義します。この値は maximum-scale と同じまたはより小さくしなければなりません。そうしないときの動作は不定です。
user-scalable	論理値 (yes または no)	no を設定すると、ユーザはページのズームができなくなります。デフォルト値は yes です。

出典:MDN meta要素の解説

デバイス・ピクセル比

PC用のウェブページ：デバイスピクセル=CSSピクセル
 高解像度のスマートフォンではデバイス：CSSが2:1になるものがある。



CSSピクセル1

CSSスプライト

- 複数の画像をまとめて、CSSで違う位置を指定することで違う画像を表示させるテクニック。
- HTTPリクエストを減らす効果がある。

電話番号

スマートフォンでは電話番号をタップすると通話できるが下記の記述をmeta要素にすることでその設定を無効にできる。

```
<meta name="format-detection" content="telephone=no">
```

スタンドアロンモード

スタンドアロンモード指定するとアドレスバーが非表示になりアプリケーションのように表示させることができる。

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

ファビコン

Favorite iconのこと。アドレスバーの左にあるようなアイコンのこと。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> サンプル問題6 </title>
  </head>
  <style>
    @media screen and (min-width: 500px) and (max-width: 800px) {
      p {color:red;}
    }
    @media print and (min-width: 500px) and (max-width: 800px) {
      p {color:blue;}
    }
    @media screen and (orientation:landscape) and (min-width: 800px) {
      p {color:yellow;}
    }
    @media screen and (orientation:portrait) and (min-width: 500px) and (max-width: 800px) {
      p {color:green;}
    }
  </style>
  <div>
    <p> サンプル問題6 </p>
  </div>
</html>
```

右のコードで横長の800pxの画面に表示される「サンプル問題6」は何色になるか選びなさい。

- A. 赤
- B. 青
- C. 黄色
- D. 緑
- E. 黒

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。



2.項目解説

2-3. 1.5.1 マルチメディア・グラフィックス系API概要 (Ver2.0の新規項目)

video 要素

映像コンテンツを埋め込む。この要素は、複数の映像ソースを含むことができる。映像ソースは `src` 属性または `<source>` 要素を使用して表し、ブラウザはもっとも適切なソースを選択する。`video` 要素をサポートしないブラウザ向けのコンテンツも追加できる。

src属性を使用した例

```
<video src="HTML5ビデオ.mp4" controls>  
<p>お使いのブラウザ5はHTMLビデオを再生できません </p>  
</video>
```

<source> 要素を使用した例

```
<video controls>  
<source src="HTML5ビデオ.mp4" type="video/mp4">  
<source src="HTML5ビデオ.webm" type="video/webm">  
<p>お使いのブラウザ5はHTMLビデオを再生できません。 </p>  
</video>
```

- `src`属性を持たない場合、0個以上の`source`要素を持つことが出来る。
- `controls`属性は再生コントロールの表示を指示する。

Video要素の主な属性

属性名	内容
width	映像の表示領域の幅を指定。CSS ピクセルで指定する。
height	映像の表示領域の高さを指定。CSS ピクセルで指定する。
poster	映像データが利用不可能な場合に表示される画像データを URI 形式で指定する。この属性が指定されない場合、再生可能な映像データがない場合は何も表示されない。
controls	再生・一時停止、音量、シーク、ポーズの各機能を制御するコントロールを表示する。
autoplay	ブール型。この属性が指定された場合、データの読み込みが完了し、再生可能な状態になった時点で即座にコンテンツの再生が始まる。
loop	ブール型。この属性が指定された場合、コンテンツの終端に達すると自動的に先頭位置に戻る。
muted	映像に含まれる音声のデフォルト設定を示すブール型属性。この属性を設定すると、初期状態が消音になります。既定値は false であり、映像再生時に音声も再生することを表す。
src	埋め込む映像コンテンツへの URL を指定する。

audio 要素

<audio> 要素は、ドキュメント内に音声コンテンツを埋め込むために使用する。この要素は、複数の音声ソースを含むことができる。音声ソースは `src` 属性または `<source>` 要素を使用して表し、ブラウザはもっとも適切なソースを選択する。

`<audio>` 要素をサポートしないブラウザ向けのコンテンツも追加できる。

Web Audio API を用いて JavaScript コードから直接、音声を生成 / 操作することが可能。

source 要素

HTML5 文書における `<picture>` 要素、`audio` 要素及び `video` 要素に対し、複数のメディアリソースを指定する場合に用いる。この要素は一般的に、各ブラウザ毎にサポートの異なる複数のメディア形式 を指定する場合に用いる。

この要素自体は空要素。開始タグは必須。終了タグを記述してはならない。

track 要素

メディア要素 (`<audio>` および `<video>`) の子として使用する。この要素は自動的に処理される字幕など、時間指定されたテキストトラック (または時系列データ) を指定できる。

この要素自体は空要素。開始タグは必須。終了タグを記述してはならない。

コンテンツの再生制御

<audio> 要素あるいは <video> 要素を用いて HTML 文書にコンテンツを埋め込むと、JavaScript を通じてそれらのコンテンツを制御できる。以下はその主な例。

- **再生を始める (または再開する)**
- **再生、一時停止、音声の増減を簡単に制御できます。**
- **メディアのダウンロードを停止する**
- **メディアをシークする**
メディアコンテンツの再生位置を特定の場所へ移動することをサポートしている。
- **再生範囲を指定する**
<audio> 要素または <video> 要素でメディアの URI を指定する際、メディアで再生する範囲を指定する追加情報を任意で含めることができる。
- **フォールバックオプション**
HTML5 メディアをサポートしないブラウザがある場合、それらのブラウザ向けの代替メディアを提供することができる。
- **エラーのハンドリング**
どのソースが読み込み失敗したのかを検出できる。



Media Source Extensions

Media Source Extensions は、JavaScript で再生用のメディアストリームを生成できるようにするため<video>要素および<audio>要素を拡張しようとする仕様、2016/11/17 W3C勧告になった。JavaScript でストリームを生成可能にすることで、適応性があるストリーミングやライブストリームのタイムシフトなど、さまざまな活用方法を容易に実現できる。

MPEG-DASH (Dynamic Adaptive Streaming over HTTP)

- HTTP を使った動画配信プロトコルの規格。
- 様々な機器に対して、動画が途切れないように帯域などの環境に応じてビットレートを切り替え、最適な高品質ビデオストリームを提供する。
- 2012年4月ISO/IEC 23001-6) ,2014年5月(ISO/IEC 23009-1:2014)
- 従来技術として下記があるが互換性がない。

Apple: HTTPライブストリーミング (HLS)

Microsoft: スムースストリーミング (SS)

Adobe: HTTPダイナミックストリーミング (HDS)

Encrypted Media Extension

Encrypted Media Extensions とはJavascript APIの仕様で、Webアプリケーションはコンテンツ保護システムとやりとりして、暗号化された音声や映像を再生することが可能になるもの。著作権付きの映画などの配信をする必要性から提案されている。

デジタル著作権管理 (DRM) システムの一部であるコンテンツ暗号解除モジュール (CDM) にアクセスするAPIを定義するものであり、

2016年7月時点でW3C勧告候補になっている。

<https://www.w3.org/TR/encrypted-media>

DRM (Digital Rights Management)

デジタルコンテンツの利用や複製を制限するために設けられている仕組みの総称

CDM (Content Decryption Module)

暗号化されたコンテンツの復号を行うモジュール (ハードまたはソフトウェア)

<canvas> 要素は、HTML上にグラフィックスを描く要素。FlashやJava アプレットなどのプラグインを使わずにJavaScriptによりグラフィックを描画することが出来るようになった。

<canvas> </canvas>で HTML上にcanvas要素を用意する。
主な属性はwidthとheight ,単位はCSS ピクセルで初期値はwidth=300, height=150 。

終了タグ (</canvas>) が必要。

<canvas> のブロック内で、代替コンテンツを提供することが可能

```
<canvas id="canvas" width="300" height="300">  
  申し訳ありませんがご使用中のブラウザはcanvasを実装していません。  
</canvas>
```

JavaScriptにより、線、円、円弧、矩形、ベジエ曲線、文字の描画や色の指定、回転、変形等が可能。
ただしアニメーションの機能はないので、アニメーションを作成したい場合は一コマずつ描画が必要。

canvasは ピクセル方式で描画を行う。

S V G (Scalable Vector Graphics)

S V G はベクター形式の画像フォーマットのこと。

SVGのメリット	SVGのデメリット
変形に強い。 描画範囲や解像度の変化に強い。 データとしては通常少ない量になる。	どこに描画するかを計算する必要があり、通常ピクセル形式に比べ描画速度が遅い。

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>SVG sample red square </title>
</head>

<body>
<h1> 赤い正方形を S V G で描画 </h1>
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" >
  <rect x="10" y="10" width="100" height="100" fill="red">
</svg>
</body>
</html>

```

- 左記のコードは単純なSVGの例。
- SVGはグラフィックス用のXML文法のため左記のような名前空間の宣言を行っている。



サンプル問題7

映画や音楽などのデジタルコンテンツの利用や複製を制限するために設けられている仕組みの総称の略称は何か。以下の中から正しいものを選びなさい。

- A. DASH
- B. DRM
- C. HLS
- D. CDM
- E. NDA

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。

サンプル問題8

SVGの説明で正しいものを選びなさい。

- A. ピクセル形式で描画を行う。
- B. ベクター形式で描画を行う。
- C. どこに描画するかを計算をする必要がある。
- D. JavaScriptを利用して描画を行う。
- E. XMLを利用して描画を行う。

※この例題は実際のHTML5プロフェッショナル認定試験とは異なります。

問題1. 正解は A,B,D の3つ

問題2. は `<style>`

は `</style>`

問題3. 正解はD の200px

問題4. 正解はC の緑

問題5. 正解はA の黒 (但しブラウザの初期値が黒が前提)

問題6. 正解はCの黄色

問題7. 正解はBのDRM

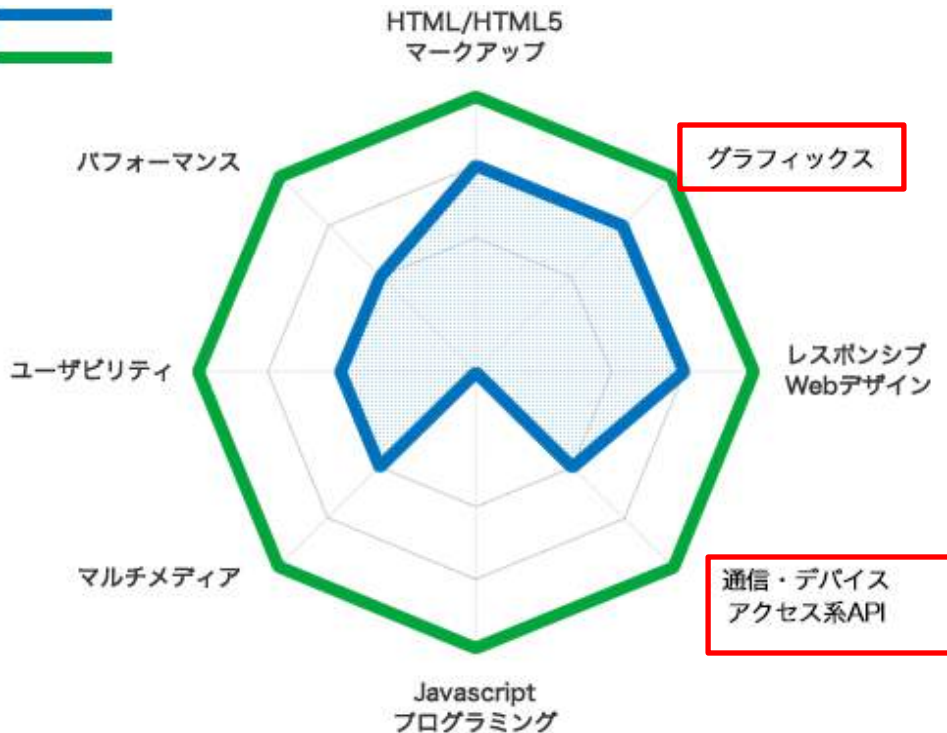
問題8. 正解は B,C,E

LPI-JAPAN HTML5 Professional Certification

Open the Future with **HTML5**.

レベル1とレベル2の資格体系 (Ver2.0)

Level.1
 Level.2



赤枠・赤字がVer1.0 からの変更点

HTML/HTML5マークアップ

HTML5に関するタグの用途、構造の組み立て方に関する技術

グラフィックス

JavaScriptやCSSなどを用いて、動的にグラフィックスを生成したりアニメーションを実現したりする技術

レスポンスWebデザイン

一つのソースで、スマートフォンなどの様々なデバイスの画面サイズに対応させるための技術

通信・デバイスアクセス系API

JavaScriptからクラウドと通信をして情報の送受信を行ったり、センサーなどのデバイスにアクセスしたりする技術

JavaScriptプログラミング

JavaScriptを使って、動的なWebコンテンツを作成する技術

マルチメディア

3D・動画・音声ファイルなどのマルチメディアコンテンツの表示・再生に関する技術

ユーザビリティ

JavaScriptやCSSなどを用いて、デザイン仕様に沿った見やすい表示や操作しやすいコンテンツを作成するための技術

パフォーマンス

ストレージや並列処理を使ってコンテンツを効率よく高速に動作させたり、オフラインでも動作する仕組みを作るための技術

レベル1の人材像

HTML5を使ってWebコンテンツを作成することができる。

ユーザー**体験を考慮した**WEBコンテンツを設計・作成することができる。

スマートフォンや**組み込み機器**など、**ブラウザが利用可能な**様々なデバイスに対応したコンテンツを制作できる。

HTML5に関連するプロジェクトにおいて、業務遂行に必要な総合的な知識を保有しており、チーム内で連携して業務を遂行できる。