



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

2018年3月25日

株式会社富士通ラーニングメディア

抜山雄一 結城陽平



富士通ラーニングメディアのご紹介

1. 会社概要

設立	1977年6月30日
資本金	3億円（全額 富士通株式会社）
売上高	92億円（2016年度）
従業員	432名（2017年3月末現在）
事業内容	人材育成・研修サービス（公開コース 1,520 コース、年間 90,000 名受講） 個人のお客様向けパソコン教室（富士通オープンカレッジ）
関係会社	株式会社富士通ラーニングメディア沖縄（研修サービス・研修サービスサポート） 株式会社富士通ラーニングメディア・スタッフ（人材派遣）
出資会社	株式会社アクト・ブレーン・ベトナム（ソフトウェア開発など）
事業所 関連施設	東京・名古屋・大阪・沖縄に5拠点（約 40 教室、 900 名以上の定員数）



品川本社



品川LC



名古屋事業所/LC



関西事業所/LC



沖縄事業所/LC

2. 富士通ラーニングメディアが目指すこと

ICT人材を支える人材育成のパートナーでありつづける

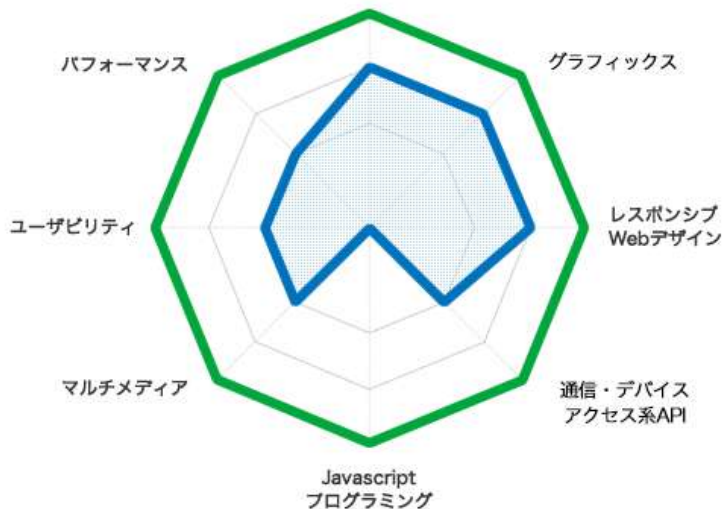
- 最高水準の『知』のサービスを提供することにより、お客様の真のパートナーを目指します。
- お客様の「成長のスパイラル」をサポートします。

HTML5 Professional Certification レベル1



Level.1 
Level.2 

HTML/HTML5
マークアップ



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

レスポンシブ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、デザイン仕様に合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

- 学習環境・各種ツールの紹介
 - ・ Visual Studio Code
 - ・ ブラウザの開発者ツール
 - ・ 推奨サイト

- 通信・グラフィックス系APIの紹介
 - ・ マルチメディア・グラフィックス系API概要
 - ・ 通信系API概要



学習環境・各種ツールの紹介

```
index.html - HTML5_LV1_Seminar_20180325 - Visual Studio Code
ファイル(F) 編集(E) 選択(C) 表示(V) 移動(G) デバッグ(D) タスク(T) ヘルプ(H)

エクスプローラー
  開いているエディター
    index.html samples/XMLHttpRequest
    JS server.js samples/XMLHttpRequest
  HTML5_LV1_SEMINAR_20180325
    node_modules
    samples
      Canvas
      MPEGDASH
      ServerSentEvents
      SVG
      WebRTC
      WebSocket
    XMLHttpRequest
      JS client.js
      JS server.js
      index.html
      JS server.js
    .gitignore
    package-lock.json
    package.json
    README.md

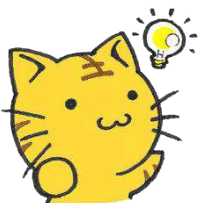
index.html
1 <!DOCTYPE html>
2 <html lang="ja">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="author" content="Fujitsu Learning Media">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <meta http-equiv="X-UA-Compatible" content="ie=edge">
9   <link rel="stylesheet" href="/node_modules/bootstrap/dist/css/bootstrap.min.css">
10  <title>XMLHttpRequest Sample</title>
11 </head>
12
13 <body>
14   <header>
15     <div class="navbar navbar-dark bg-dark box-shadow">
16       <div class="container d-flex justify-content-between">
17         <strong class="text-white">XMLHttpRequest Sample</strong>
18       </div>
19     </div>
20   </header>
21   <div class="container">
22     <div class="row">
23       <div class="col">
24         <div class="jumbotron">
25           <h1 class="display-4">XMLHttpRequest Sample</h1>
26           <p class="lead">入力した数値の階乗を行います。</p>
27         </div>
28       </div>
29     </div>
30     <div class="row justify-content-center">
31       <div class="col">
32         <form class="form-inline" id="fm">
33           <div class="form-group row">
34             <input type="text" class="form-control" value="10">
35             <input type="button" value="計算" class="btn btn-primary">
36           </div>
37         </form>
38       </div>
39     </div>
40   </div>
41 </body>
42 </html>
```

Visual Studio Code

<<<https://code.visualstudio.com/>>>

軽量で高機能なテキストエディタ。

拡張機能を追加すれば自分の好みにカスタマイズできるよ。

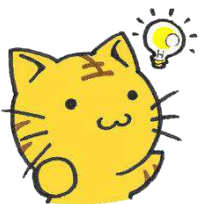




Visual Studio Codeの機能

- インテリセンスによるタイプミス撲滅
入力補完機能によって、**打ち間違いを防止**。
表示される候補を選択するだけです。
- 定形コードを簡単に挿入可能
長いソースコードも簡単に書けます。
同じタブを何度も書かずに済みます。
※細かい使い方はチートシート(<https://docs.emmet.io/cheat-sheet/>)で!
- 拡張機能でより使いやすく
機能が足りないなら、**追加しましょう**。

Windows、Linux、Mac OS Xで動作するよ。
複数言語に対応しているので、
HTML/CSS/JavaScriptに学習に使えるよ！
上記以外にも便利な機能がたくさん使えるんだ。



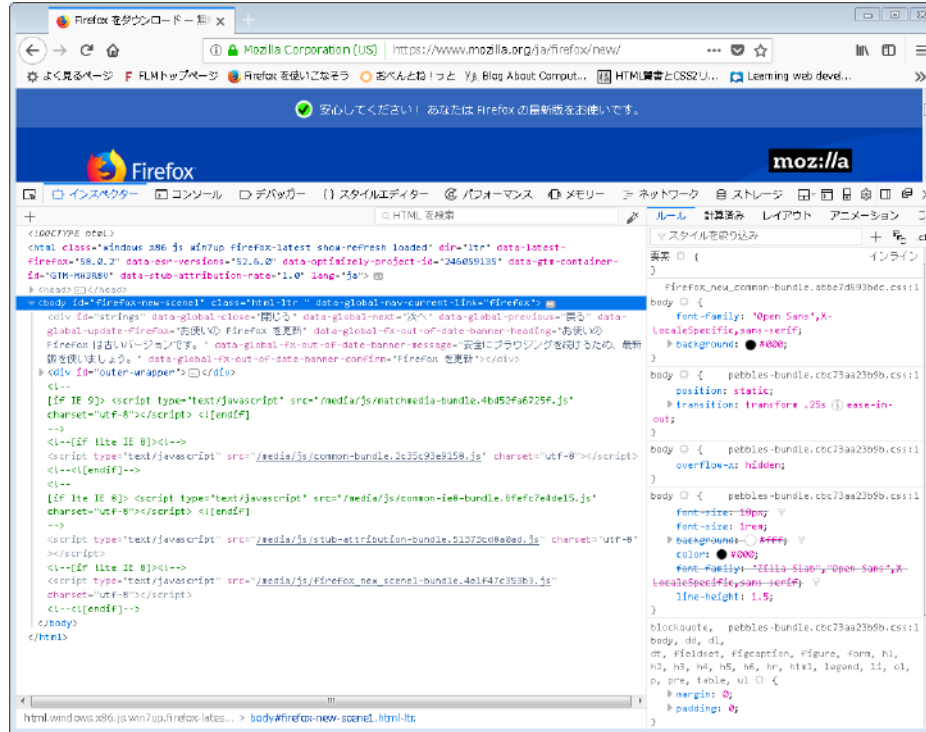


オススメの拡張機能

- Beautify
ソースコードを**自動で整形**してくれる
- HTML Hint
HTMLの**静的解析**ツール
- Can I use
ブラウザごとの**対応状況**を調査してくれる
- Preview on Web Server
Webサーバを用いた**ライブプレビュー**

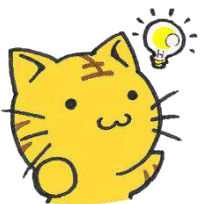


ブラウザの開発者ツール



Firefox

<<<https://www.mozilla.org/ja/firefox/>>>
 最近のアップデートで高速化したよ。
 開発者向けの機能が日本語で使えるのが便利！



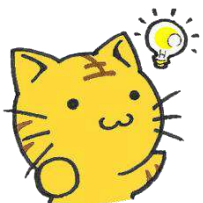


開発者ツールの機能

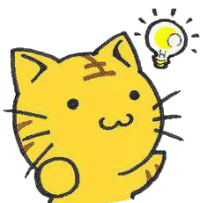
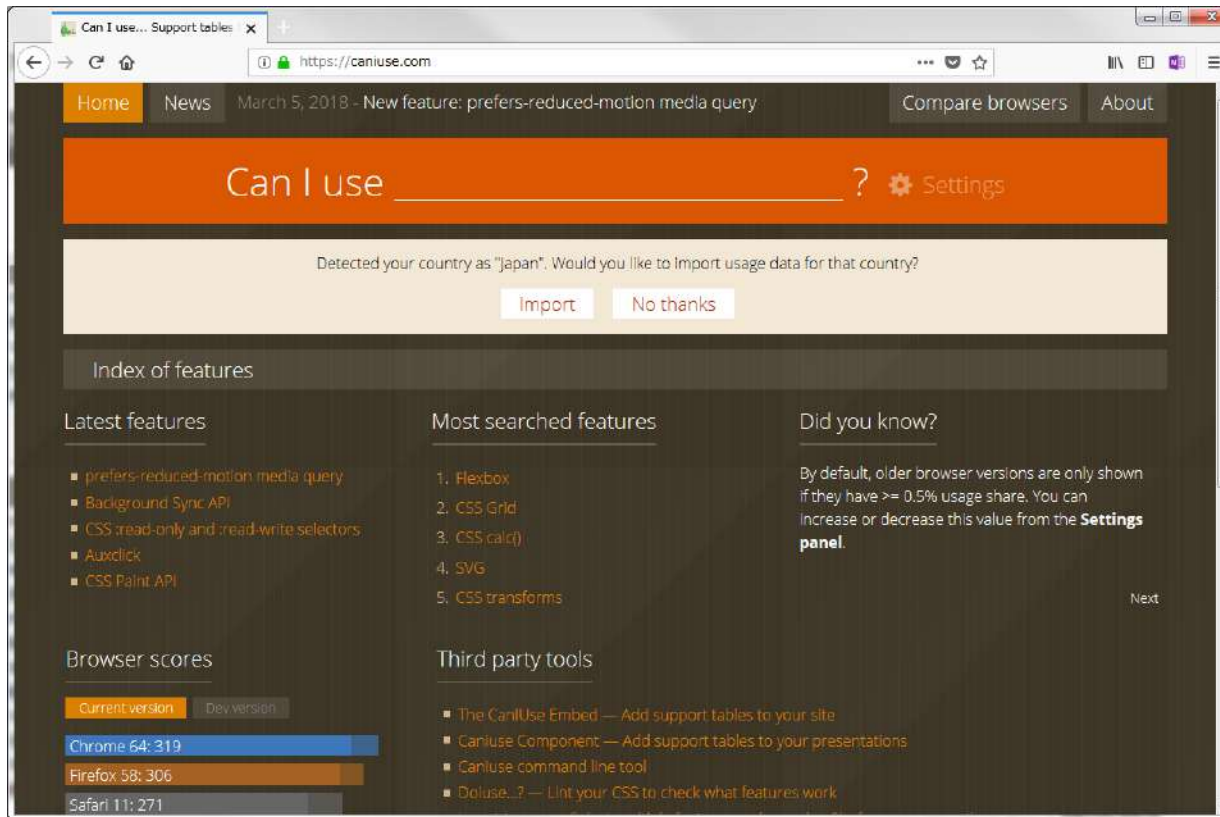
- インスペクター
閲覧しているWebページの文書構造（HTML）、デザイン(CSS)の**確認・編集**ができます。
- コンソール
エラー等のメッセージが表示されます。
メッセージの詳細へのリンクも用意されている。
- ネットワーク
HTTPステータスコードやリクエスト・レスポンスヘッダ等、ブラウザ/サーバ間の**通信情報を調査**できます。



Mozilla Developer Network



Mozilla Developer Network
<<<https://developer.mozilla.org/ja/>>>
Web標準技術に関する情報が充実しているよ。



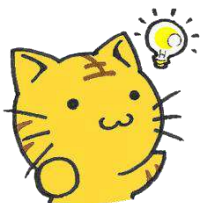
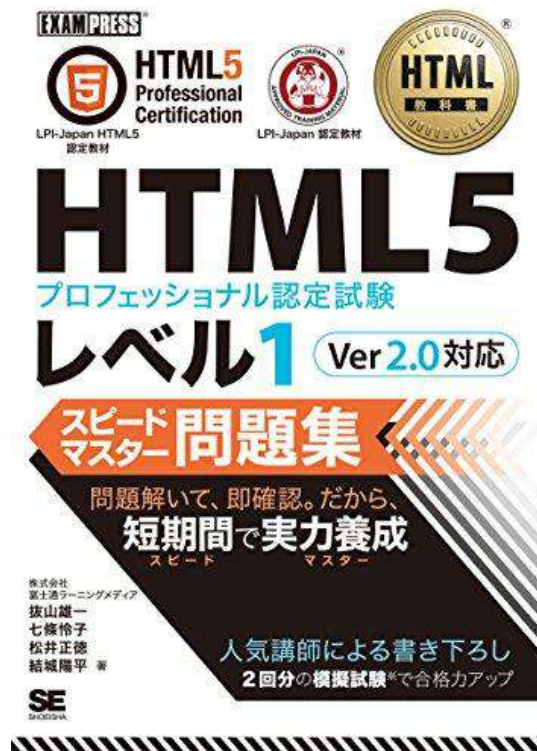
Can I use

<<<http://caniuse.com/>>>

ブラウザごとの対応状況を公開しているサイトだよ。



HTML教科書 HTML5プロフェッショナル認定試験 レベル1 スピードマスター問題集 Ver2.0対応



特別出張販売中です！！



通信・グラフィックス系APIの紹介



サンプルコード

サンプルコードは以下のリポジトリに格納しています。

クロスブラウザ対策などを行っていないため、環境によっては動作しません。ご了承ください。

<<goo.gl/E4Gf38>>

1.5.1 マルチメディア・グラフィックス系API概要

1.5.1 マルチメディア・グラフィックス系API概要

重要度	★★★★★ 5
出題種別	<ul style="list-style-type: none">知識問題記述問題
説明（望まれるスキル）	<ul style="list-style-type: none">ビデオやオーディオを適切にWebコンテンツとして活用するための基礎知識を有し、具体的に何ができるのかを理解している。静的・動的を問わず、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。

■ Canvasの特徴

JavaScriptを使ってブラウザ上に図を描画する機能です。**ビットマップ形式**で描画を行うためDOMツリーは構成しませんが、JavaScriptからCanvas上のピクセル情報を操作できます。アニメーション描画のメソッドは存在しませんが、**静止画の描画を繰り返すことでアニメーションを実現できます**。また、ブラウザに対応した画像ファイルをCanvasに読み込み、加工し、画像ファイルとして書き出すことも可能です。

■ SVGの特徴

Webで利用できる**ベクター形式**の画像データであり、実態はXMLに基づいたマークアップ言語です。SVGの画像は計算式で表現されるため、**画像を拡大しても画像の劣化が発生しにくい**という特徴があります。また、SVGで描画された図形はDOMツリーを構成するため、一度描画した図形をJavaScriptから操作することが可能です。

※ SVG生成ツールとして、vectr <<<https://vectr.com/>>>などがあります。



CanvasとSVGのソースコード (部分)

■ Canvas

1. `this.context = canvas.getContext('2d');`
2. `this.context.beginPath();`
- 3.
4. `this.context.lineTo(x, y);`
5. `this.context.stroke();`

■ SVG

1. `<?xml version="1.0" encoding="UTF-8" standalone="no"?>`
2. `<!DOCTYPE svg PUBLIC ...>`
3. `<svg version="1.1" ...>`
4. `<defs>`
5. ...

- Media Source Extensions(MSE)による動画配信
MSEとは、HLSやMPEG-DASHをサポートする技術であり、Video要素などをプラグインなしでストリーミング再生可能にします。MSEは、あらかじめ短い時間に区切られたメディアデータを扱うため、動画の合間に広告やその他のコンテンツを挿入したり、コンテンツの途中から再生を開始したりできます。また、ネットワーク帯域幅やCPU使用率に基づいてストリーミングのビットレートを変更することもできます。
 - Encrypted Media Extensions (EME)
なお、MSEには著作権管理の技術 (DRM) の仕組みがないため、不正にコンテンツをコピー・再配布される危険性があります。これらの危険性を低減するには、暗号化の機能を用いてコンテンツを保護できるEncrypted Media Extensions (EME) を使用します。
- ※ クライアントサイドの実装だけではなく、動画配信のサーバが必要になります。
クラウドサービスなどを利用すれば、比較的手軽に動画配信サーバを構築できます。



MPEG-DASHを使用したソースコード

■ HTML

1. `<main class="col-sm-12">`
2. `<video width="600px" autoplay controls
id="videoplayer"></video>`
3. `</main>`

■ JavaScript

1. `var url = "url";`
2. `var player = dashjs.MediaPlayer().create();`
3. `player.initialize(document.querySelector
("#videoplayer"), url);`

※ Dash.jsライブラリを使用しているため、W3C標準のJavaScriptのコードではありません。



1.5.4 通信系API概要

1.5.4 通信系API概要	
重要度	★★★★ 4
出題種別	<ul style="list-style-type: none">• 知識問題• 記述問題
説明（望まれるスキル）	<ul style="list-style-type: none">• 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。



通信系APIの比較

	Server-Sent Events	WebSocket	XMLHttpRequest	WebRTC
通信プロトコル	HTTP	WebSocket Protocol	HTTP	SDPなど
非同期通信	○	○	○	○
常時接続	△	○	△	○
双方向通信	×	○	△ (リクエストやレスポンス時にサーバ間とデータを送受信する)	○
特徴	Webサーバから一方的にデータを送信する。データ送信が不要ならばXMLHttpRequestによるロングポーリングよりも通信効率が高い。	双方向の常時接続が可能。複数クライアントとの通信も可能。そのため、チャット機能などの実装に向いている。	最も古くから実装されている通信系API。そのため、多くのブラウザに実装されており、互換性が高い。	オーディオ/ビデオなどを送受信できるP2P通信。オンライン会議システムなどの実装に向いている。

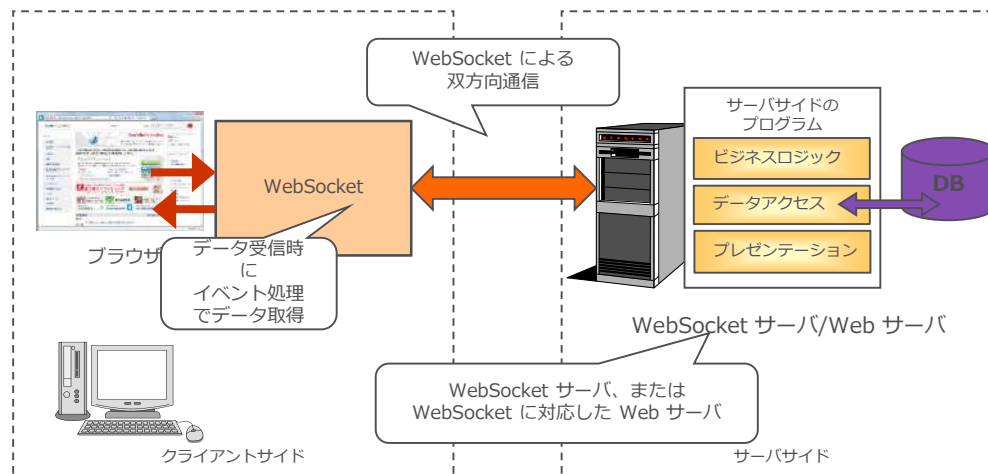
WebSocketの特徴

■ 双方向通信を実装できる

クライアントとWebSocketサーバ間で**接続を維持した双方向通信**を実装できます。XMLHttpRequestやServer-Sent Eventsなど、一回の通信ごとにリクエストとレスポンスを行うHTTPベースの通信と比較して、常時接続して双方向通信を行うWebSocketは、送受信処理の効率が高いです。

■ WebSocketサーバを使用する

通信方式として**HTTPではなくWebSocket Protocol**を使用します。そのため、サーバ側はWebSocket Protocolに対応した**WebSocketサーバ**を使用する必要があり、WebSocketに対応していない従来のWebサーバは使用できません。



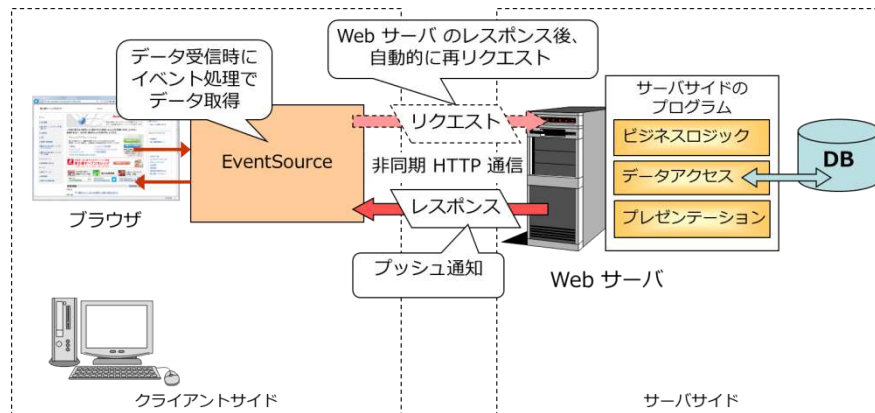


WebSocketのソースコード

```
1. // WebSocketの生成
2. var ws = new WebSocket("ws://localhost:8000/broadcast");
3.
4. // WebSocket接続イベント
5. ws.onopen = function () {
6.   console.log("接続が開きました");
7. };
8.
9. // WebSocketメッセージイベント
10. ws.onmessage = function (e) {
11.   var data = JSON.parse(e.data);
12. };
13.
14. // WebSocket切断イベント
15. ws.onclose = function () {
16.   console.log("切断しました");
17. };
18.
19. // メッセージ送信
20. ws.send(data);
```


Server-Sent Eventsの特徴

- **プッシュ通知に特化している**
JavaScriptを使用して**プッシュ通知を実装できます** (text/event-stream)。
XMLHttpRequestを使用と比べて、簡単にプッシュ通知を実装できます。ただし、**クライアントからWebサーバにデータを送信できません**。
- **効率的なデータ送信ができる**
Server-Sent Eventsのプッシュ通知はヘッダーに含まれる送信情報が少ないです。そのため、Long Pollingによるプッシュ通知よりも**効率的なデータ送信**ができます。
- **Webサーバを使用できる**
プッシュ通知はWebサーバを使用して実装できます。そのため、サーバサイドでは**従来のWebアプリケーションと同じ技術で実装**できます。





Server-Sent Eventsのソースコード

```
1. // EventSourceの生成
2. var source = new EventSource('/events');
3. // EventSourceのメッセージイベント
4. source.onmessage = function (e) {
5.     var p = document.createElement("p");
6.     p.innerText = e.data;
7.     document.getElementById("messageBody").appendChild(p);
8. };
```

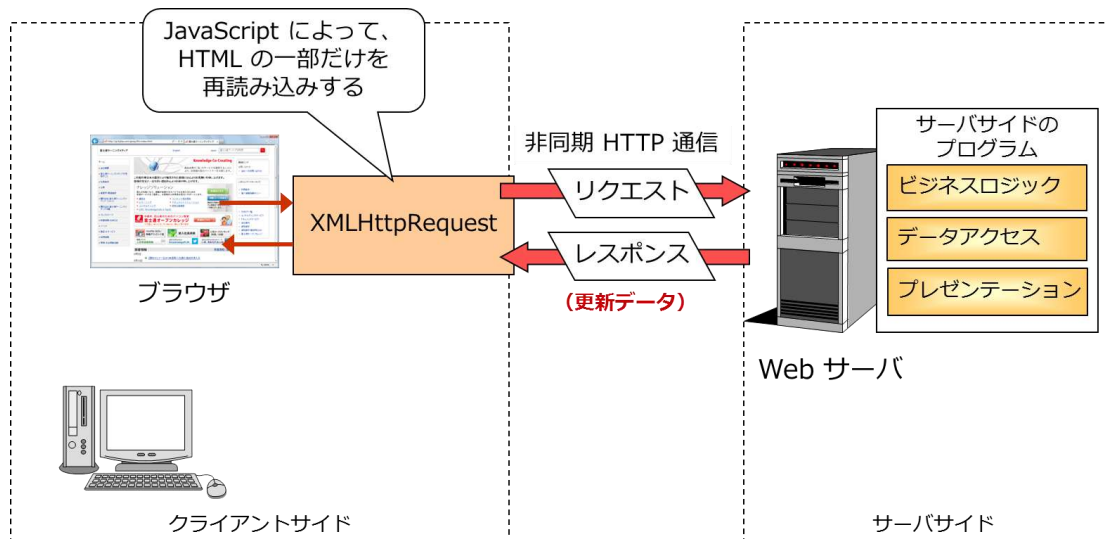
XMLHttpRequestの特徴

■ 非同期通信

Ajaxを実装できます。 Ajaxでは、非同期で通信し、リクエスト後、Webサーバからレスポンスでデータが渡されるまでの間、他の処理を実行できます。Webサーバからのレスポンス後に、コールバック関数を呼び出すことで、Web ページの一部を更新できます。

■ 通信系APIで最もブラウザ実装が進んでいる

XMLHttpRequestは、JavaScriptの通信系APIの中では、最も古いものの1つです。そのため、**最もブラウザ実装が進んでいます。**





XMLHttpRequestのソースコード

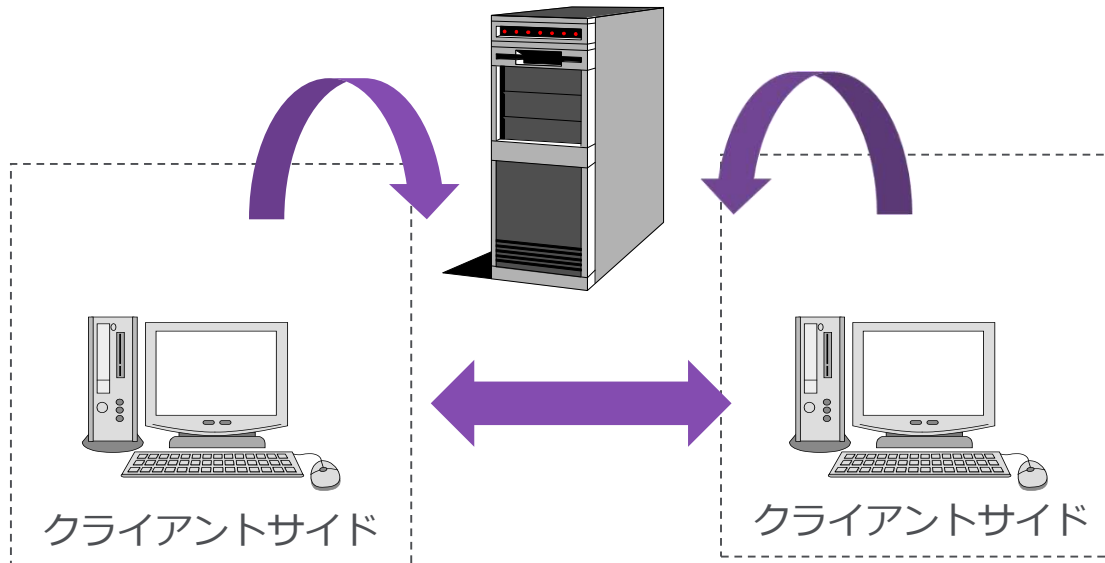
```
1. var client = new XMLHttpRequest();
2. client.onload = function () {
3.     if (this.readyState === 4) {
4.         if (this.status === 200) {
5.             document.getElementById("result").innerText =
                                   this.responseText;
6.         }
7.     }
8. };
9. // メッセージ送信
10. client.open("GET", "http://localhost:8000/factorial/"
                                   + number);
11. client.send();
```

■ P2P通信

WebSocketやXMLHttpRequestなどを用いた通信は、クライアント/サーバ間で通信が行われます。一方、WebRTCでは**クライアント同士 (P2P) での通信**を実現できます。なお、NAT内のクライアントと通信するには、ICEサーバが必要になります。

■ 動画やオーディオを用いた通信ができる

WebRTCでは動画やオーディオをもいた通信を実現できます。そのため、オンライン会議システムなどを作成できます。






WebRTCのソースコード

```
1. var webrtc = new SimpleWebRTC({
2.   localVideoEl: 'localVideo',
3.   remoteVideosEl: 'remotesVideos',
4.   autoRequestMedia: true,
5.   debug: true
6. });
7.
8. webrtc.on('readyToCall', function () {
9.   webrtc.joinRoom('myroom');
10.});
```

※ SimpleWebRTCライブラリを使用しているため、W3C標準のJavaScriptのコードではありません。

LPI-JAPAN HTML5 Professional Certification

Open the Future with **HTML5**.



FUJITSU

shaping tomorrow with you