



**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6月**

**NTTテクノクロス株式会社**

**鈴木雅貴**



# 自己紹介

- 鈴木雅貴（すずきまさたか）
  - NTTテクノクロス株式会社
  - レベル1認定取得
- レベル1試験対策本執筆
- 弊社技術ブログでVR記事連載



# 本日は話すこと

- JavaScript概要
- 各種APIを試してみよう
  - マルチメディアグラフィックスAPI (Video、Canvas)
  - デバイスアクセスAPI (Geolocaition)
  - オフラインストレージAPI (Web Storage)

# JavaScript概要

何をするものかを理解しよう



# JavaScriptとは

- Webページの構成物を操作することができるプログラミング言語
  - 主にWebブラウザ上で動作する
- ECMAScriptと呼ばれる仕様の実装の総称
- ES5やES6といった表記はECMAScriptの略 + バージョン番号



# よくあるJavaScriptの利用シーン

- フォーム入力内容チェック
- 特定要素のクリックで新しく要素を表示
- 再読込なしでページの情報を更新



# ページ内でのJavaScript使用方法 (1)

- script要素の内容として記述する

```
<script>  
// JavaScriptのコードを記述  
</script>
```



## ページ内でのJavaScript使用方法 (2)

- script要素のsrc属性で外部スクリプトファイルを指定して読み込む

```
<!-- JavaScriptのコードを記述したscript.jsを読み込む -->  
<script src="js/script.js"></script>
```





- CodePenを使用

- Webブラウザ上でHTML/CSS/JavaScriptを書いて確かめることができる
- 右上の[Create]→[New Pen]で新規作成
- あとは書くだけで結果が表示されていく
- 勉強の際のちょっとした確認に便利

# video

video要素を操作



- video要素で動画を再生可能
- 普通に使った場合はブラウザ組み込みのプレイヤーで操作する
- JavaScriptを使えば組み込みプレイヤー以外から制御可能

- <https://codepen.io/suzukima/pen/qmzRby>
- [再生]ボタンで再生、[停止]ボタンで停止

```
<video width="100%" controls src="動画ファイルURL"></video>
<div class="controller">
  <button type="button" class="btn btn-play">再生</button>
  <button type="button" class="btn btn-pause">停止</button>
</div>
```

- "var 変数名"で変数宣言
- 変数として宣言したものは、あとから利用可能

```
// 変数videoの宣言
var video = document.querySelector("video");

// 略

playBtn.addEventListener("click", function () {
// 変数videoの利用
  video.play();
});
```

- 制御する要素を取得
- "document.querySelector(セレクタ)"でセレクタでの要素取得が可能

```
// 変数videoに取得したvideo要素をいれておく  
var video = document.querySelector("video");
```

- "取得した要素.addEventListener()"で、取得した要素に指定したイベントが発生した際に実行する内容を指定することができる

```
// 再生ボタンがclick(クリック)されたら、function()を実行
playBtn.addEventListener("click", function () {
  // 実行する中身はvideoの再生
  video.play();
});
```





# 代表的イベント

イベント名	概要
click	クリックされたとき
mouseenter	マウスカーソルが要素に重なったとき
mouseleave	マウスカーソルが要素から離れたとき
focus	要素がフォーカスされたとき
keydown	キーが押されたとき
keyup	キーが離されたとき
change	input等の値が変更されたとき
DOMContentLoaded	HTMLの読み込みと解析が終わったとき
load	画像等ページが完全に読み込み終わったとき

## • 取得したvideo要素を使いメソッド実行

```
// 再生ボタンクリック時の処理
playBtn.addEventListener("click", function () {
    // 取得したvideo要素のplay()メソッドで再生
    video.play();
});

// 停止ボタンクリック時の処理
pauseBtn.addEventListener("click", function () {
    // 取得したvideo要素のpause()メソッドで停止
    video.pause();
});
```



- 以下が参考になります

- [HTML5 の audio 要素と video 要素の使用 - HTML | MDN](#)
- [HTMLMediaElement - Web API インターフェイス | MDN](#)

# Geolocation

## 位置情報の利用



# Geolocation API

- Webブラウザから位置情報を利用可能
- 近くの店舗を表示したり、現在の位置を情報として提示するようなサービスを実現することができる

- <https://codepen.io/suzukima/pen/vmqmyd>

```
<button onclick="geoDisp()">位置情報表示</button>  
<div id="out"></div>
```

- "onclick=関数"で、クリックされたときに実行する関数を指定
- この場合はgeoDisp()に書かれた処理が実行される



## 要素の取得 (2)

- "document.getElementById(ID名)"で、指定したID名の要素を取得

```
// id="out"の要素を取得  
var output = document.getElementById("out");
```



- "取得した要素.innerHTML=表示内容"で、取得した要素の表示を、指定した表示内容のものに書き換えることができる

```
// 位置情報が表示されるまでの表示  
output.innerHTML = "<p>位置情報を取得しています…</p>";
```



## 位置情報の取得 (1)

- "navigator.geolocation.getCurrentPosition(success, error)"で、位置情報を取得し、成功したらsuccessの処理、失敗したらerrorの処理を実行させることができる

```
// 位置情報を取得  
navigator.geolocation.getCurrentPosition(success, error);
```

- 成功時に実施する処理の中で、緯度や経度などの情報を取得して利用可能

```
function success(position) {  
  // 緯度  
  var latitude = position.coords.latitude;  
  // 経度  
  var longitude = position.coords.longitude;  
  // 以下略
```

- "new Image()"でimg要素を作る
- 作った要素のsrc属性を指定して画像を読み込むようにする
- 作った要素は"取得要素.appendChild(作った要素)"でHTMLに追加

```
var img = new Image();  
img.src =  
"https://maps.googleapis.com/maps/api/staticmap?center=" +  
latitude + "," + longitude +  
"&zoom=13&size=300x300&sensor=false";  
output.appendChild(img);
```



# Geolocation APIについて

- 以下が参考になります
  - [Geolocation の利用 - Web API インターフェイス | MDN](#)

# Web Storage

ブラウザにデータを保存し利用



- ブラウザにデータを保存可能
- データはキーと値のペアで保存される
  - 例えばキーが"名前"で、値が"鈴木雅貴"
- かんたんな設定の保存などに使われる

- <https://codepen.io/suzukima/pen/QvXvZW>
- 指定した色が背景色になる
- 再読込しても指定した色はそのまま



```
<input type="color" class="colorpallet">
```

- "document.getElementsByTagName(要素名)"で指定した要素を取得
- 複数取得するので、その1つ目という意味で後ろに[0]をつける

```
// body要素  
var bodyElm = document.getElementsByTagName('body')[0];
```



# ストレージの宣言

- 2種類あるストレージのうち、どちらを使うかを指定して宣言
  - localStorageはブラウザを閉じて開いてもデータが残っている
  - sessionStorageはブラウザを開いている間に限りデータが残っている（再読込を含む）

```
// 変数storageはlocalStorageに  
var storage = localStorage;
```



# ストレージにデータを格納

- "ストレージ.setItem(キー, 値)"で、ストレージに値を保存することができる
- 今回はキーをbgcolorとしている

```
// ストレージのbgcolorに値を保存  
storage.setItem('bgcolor', this.value);
```

- "if(条件) { 特定の処理 }"で、条件どおりのときに特定の処理が実施される

```
// ストレージにbgcolorがあれば  
if (storage.getItem('bgcolor')) {  
    // ストレージからbgcolorを取得して、背景色を変更する処理  
}
```

- "ストレージ.getItem(キー)"で、キーに対応する値を取得することができる

```
// ストレージからbgcolorを取得  
var color = storage.getItem('bgcolor');
```



# スタイルの変更

- "取得した要素.style.プロパティ名"に値を指定することで、指定したプロパティの値を変更することができる

```
// body要素のbackgroundColor(background-color)を変更  
bodyElm.style.backgroundColor = color;
```



# フォームの値を変更

- "指定した要素.value"に値を指定することで、フォームの値を外から変更可能

```
// 選択している色を変数colorに変更  
colorPallet.value = color;
```





# Web Storageについて

- 以下が参考になります
  - [Web Storage API を使用する - Web API インターフェイス | MDN](#)

# 勉強方法



# できれば触ってみましょう

- level1はプログラミングが対象外ですが、読んだだけではイメージがわきづらい
- CodePenのような手軽に試せる環境があるので、それを利用して少しでも試してみるのがおすすめ
  - そのほうが技術者としての力もつく
- [Mozilla Developer Network](#) は日本語も多く情報源として優秀

# LPI-JAPAN HTML5 Professional Certification

Open the Future with **HTML5**.