

第4章 WebSpeech API

Web応用

第14回 さまざまなAPI

第4章

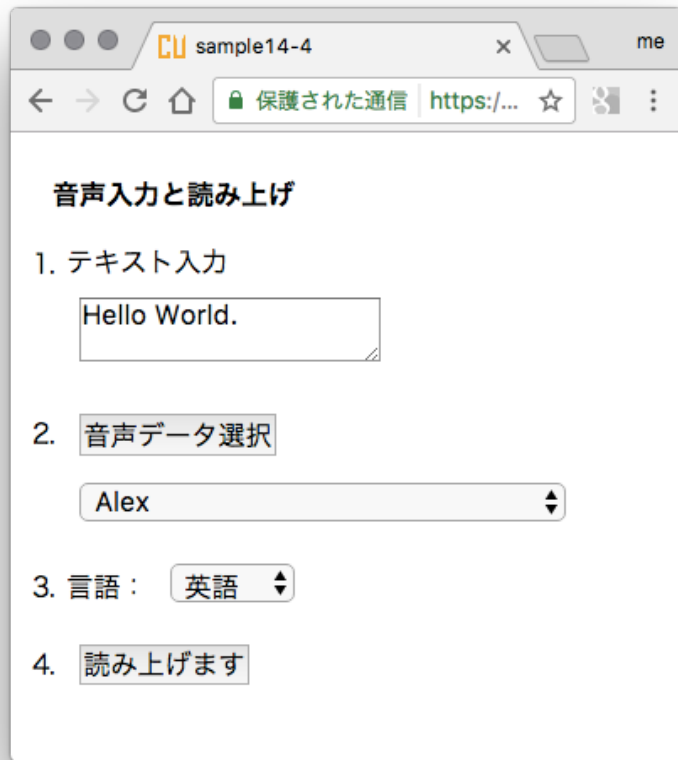
WebSpeech API

第4章 学習目標

テキストを読み上げたり、音声を認識するAPIについて理解できる。

1. ファイルの用意

WebSpeech APIでは、テキストを読み上げたり、音声を認識することができます。本章では、テキストの読み上げを演習しましょう。



ファイルを準備

それでは早速ファイルを準備しましょう。ファイル名は「sample14-4.html」です。

JavaScript内では、次の二つの関数を制作します。

- PC毎に読み上げできる音声が変わるため、それを取得するための関数「voice()」
- 音声で読み上げるプログラム「mySpeech()」

📄 サンプル

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>sample14-4</title>
6     <style>
7       *{font-size:16px;margin:0.5em;padding:0.3px;}
8     </style>
9   </head>
10  <body>
11    <h1>テキスト読み上げ</h1>
12    <ol>
13      <li>テキスト入力<br>
14        <textarea id="input1">Hello World.</textarea>
15      </li>
16      <li>
17        <button onclick="voice();">音声データ選択</button>
18        <select id="voice"></select>
19      </li>
20      <li>言語 :
21        <select id=lang>
22          <option value="ja-JP">日本語</option>
23          <option value="en-US" selected>英語</option>
24        </select>
25      </li>
26      <li><button onclick="speech1();">読み上げます</button></li>
27
28    <script>
29      //①バグのため
30      var voices = speechSynthesis.getVoices();
31
32      //②読み上げ可能な音声リストを取得
33      function voice(){
34
35      }
36
37      //③再生
38      function speech1(){
39
40      }
41    </script>
42  </body>
43 </html>

```

補足

- ①：このAPIにはバグが報告されていて、「speechSynthesis.getVoices()」の記述はそのバグに対処するためのものです。覚える必要はありません。

2. WebSpeech APIの設置

それでは音声のリストを取得する「voice()」と、読み上げる「mySpeech」を記述しましょう。

1. 音声リストの取得

皆さんのPCごとに利用できる音声はマチマチです。そのため、PCで利用できる音声のリストを取得しましょう。関数は「voice()」です。この関数は、

- 音声の配列を「voices」とします。voices[0]から順番に音声のリストのデータが格納されています。
- 具体的な音声名は、「voices[i].name」に格納されています。
- それらを取得するため、「speechSynthesis」オブジェクトの「.getVoices()」メソッドを実行します。

📄 サンプル

```
32 //②読み上げ可能な音声リストを取得
33 function voice(){
34     voices = speechSynthesis.getVoices(); //②-1
35     var voiceStr; //②-2
36     for(var i=0;i<voices.length;i++){ //②-3
37         if(voices[i]){
38             voiceStr += "<option value="+i+">"+voices[i].name+"</option>";
39         }
40     }
41     document.getElementById("voice").innerHTML = voiceStr; //②-4
42 }
```

補足

- ②-1：「speechSynthesis」オブジェクトの「.getVoices()」メソッドで音声リストを取得。
- ②-2：リストの一覧を表示するため、音声名などのテキストを保存する変数「voiceStr」を宣言。
- ②-3：0番目から音声のリストの項目の個数（voice.length）分だけ繰り返しながら、音声の名前「voices[i].name」を取得していく。その際、option要素を使うことでプルダウンで選択が可能。
- ②-4：音声の名前を含んだHTMLコードをid「voice」の要素の中に代入すれば、リストがプルダウンで使用可能。

2. 読み上げ

テキスト読み上げは関数「speech()」で行います。この関数は、

- 「SpeechSynthesisUtterance()」で音声のオブジェクト「speech1」を生成し、言語、音声の種類を設定します。
- 読み上げるオブジェクト「speechSynthesis」の「.speak(speech1)」メソッドで「speech1」を読み上げます。

📄 サンプル

```

46 //③再生
47 function speech1(){
48     var input1 = document.getElementById("input1").value;    //③-1
49     var voice = parseInt(document.getElementById("voice").value);    //③-2
50     voices = speechSynthesis.getVoices();    //③-3
51     var speech1 = new SpeechSynthesisUtterance(input1);    //③-4
52     speech1.lang = document.getElementById("lang").value;    //③-5
53     speech1.voice = voices[voice];    //③-6
54     speechSynthesis.speak(speech1);    //③-7
55 }

```

補足

- ③-1 : textarea要素(idはinput1)に入力された値を取得して、input1とする。
- ③-2 : 音声リストのプルダウンから選択された項目の値(value)を取得して、voiceとする。
- ③-3 : 「speechSynthesis」オブジェクトの「.getVoices()」メソッドで音声リストを取得する。
- ③-4 : 「new SpeechSynthesisUtterance(input1)」で、input1を元にした読み上げ音声のオブジェクト「speech1」を生成する。
- ③-5 : オブジェクト「speech1」の言語の値「.lang」を指定
- ③-6 : オブジェクト「speech1」の音声の値「.voice」を指定
- ③-7 : 「speechSynthesis」オブジェクトの「.speak(speech1)」メソッドで、読み上げを行う。

3. サーバにアップして動作確認

1. サーバにアップロード

この「WebSpeech API」をサーバにアップロードして試してみてください。

2. WebSpeech APIについて

なお、このAPIは、未サポートのブラウザもあり、まだまだ開発の段階です。興味があれば、次のURLを参照してください。

MDN「Web Speech API」

https://developer.mozilla.org/ja/docs/Web/API/Web_Speech_API

3. (参考)音声認識

音声認識は、PCの環境やマイクの有無により演習できないことがあるので、本講義では参考として紹介します。可能ならサーバにアップロードして試してみてください。



コードは次のとおりです。ファイル名は「sample14-4-rec.html」

ボタンをクリックすると関数「recognition1()」が起動します。この関数は、

- 「webkitSpeechRecognition()」から音声認識するオブジェクト「recognition」を生成し、言語を指定して、認識を開始します。
- 認識の結果は、「e.results.item(0).item(0).transcript」に格納され、それを出力します。

サンプル

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>sample14-4-rec</title>
    <style>
      *{margin:0;padding:0;}
      ol{list-style:none;}
      h1,li{margin:1em;font-size:16px;}
    </style>
  </head>
  <body>
    <h1>音声認識</h1>
    <ol>
      <li>言語 :
        <select id="lang">
          <option value="ja-JP">日本語</option>
          <option value="en-US" selected>英語</option>
        </select>
      </li>

      <li><button onclick="recognition1();">認識する</button></li>
      <li><p id="output1">認識した内容</p></li>
    </ol>

    <script>
      //音声認識
      function recognition1(){
        var recognition = new webkitSpeechRecognition(); //①
        recognition.lang = document.getElementById("lang").value; //②
        recognition.start(); //③

        recognition.addEventListener('result',function(e){
          var rec = e.results.item(0).item(0).transcript; //④
          document.getElementById("output1").innerHTML= rec; //⑤
        });
      }
    </script>
  </body>
</html>

```

解説

- ① : 「webkitSpeechRecognition()」は音声認識のコンストラクタ。new演算子は音声进行認識するオブジェクト「recognition」を生成。
- ② : オブジェクト「recognition」の「lang」の値を設定(「日本語」または「英語」)
- ③ : 「.start()」は認識をスタート。
- ④ : オブジェクト「recognition」の結果のデータは引数「e」に渡され、認識した文字列は「e.results.item(0).item(0).transcript」に格納されている。
- ⑤ : その文字列をp要素(idは「output1」)に出力。

参考

「テキスト読み上げ」と「音声認識」を組み合わせたサンプルを作りましたので紹介します。試したりソースをのぞいてみてください。

「テキスト読上と音声認識」

<https://web.p.cyber-u.ac.jp/~nakashima/web2/14/sample14-4-ref.html>

練習問題1

問題

【クイズ】 空所補充

テキストを読み上げるコードの空欄を埋めてください。

speechSynthesis.`speak` (speech1);

解答を提出する

練習問題1の解説

正解は

speechSynthesis.`speak`(speech1)

です。

第4章 まとめ

テキストを読み上げたり、音声を認識するAPIについて理解した。

第4章 終わり

Web応用

第14回 さまざまなAPI

第4章

WebSpeech API

終わり

© Cyber University Inc.