

## 音声認識 API

### 概要

→パソコンの操作が難しい人，目の不自由な人向けに，青空文庫[1]を音声で操作し，声で検索した本の読み聞かせを行うウェブサイトの作成

### 機能

- 以下の単語を自サイト内で言葉を発すると，単語を読み取り，青空文庫内で機能を実行する。
    - ・検索（作品名，著者名で検索し，検索結果を音声で出力する）
    - ・戻る，次へ（ひとつ前のページに戻る，次のページに進む）
    - ・はい，いいえ（決定するかしないかを選択する）
    - ・リセット（一番初めのページに移動する）
    - ・開始，終了（読み聞かせの開始，停止を実行する）
    - ・早い，遅い（サイト内で発せられる音声のスピードを変更する）
- if(text === “単語”) function 名();で実現可能

- 履歴機能

#### 読み聞かせを開始するまでの手順

1. 自サイト内の google の検索バーで作品名または著者名を検索する。  
→無音になって3秒経過したら自動で音声認識終了し，検索
2. ・音声で出力された検索結果の中から，読み聞かせの作品を決定する。  
・検索結果の一番上の検索結果を自動でクリック  
→検索結果の出力を変更すれば実現できそう。
3. 読み聞かせの本のファイルを XHTML 版で開く。
4. 開始の合図で読み聞かせを開始する。

#### 課題

- ・セِيمオリジンポリシー[2]

他のオリジンにあるリソースへのアクセスを制限するセキュリティのしくみである。これは，自サイトの中に iframe で青空文庫を埋め込んでいるので，青空文庫を JavaScript で操作を実行するのが難しい可能性がある。そのため，何らかの方法で動かさなければならない。その方法の候補として，「スクレーピング」や青空文庫にある著作権切れの書籍のファイルをダウンロードし，自サイトのみで実現する方法が考えられる。

- ・スクレーピング[3]

ウェブサイトから必要な部分だけを抽出して変換すること。この技術を使うことで別のサイトの必要な情報だけを抜き取れるため、セームオリジンポリシーに触れずに、JavaScript を実行できると考える。

- ・サーバーサイドプログラミング[4]

サーバー上で動作するプログラムのことで、情報を効率的に保管し提供したり、コンテンツへのアクセスを制限したりすることが出来る。自サイト上でリクエストしたことを、青空文庫から情報が返ってくる仕組みを作れるのではないかと考える。

- ・音声認識開始の単語

音声認識を開始するための単語を決定する。有名な例を挙げると、「OK! Google」, 「Hey, Siri」, 「Alexa」などがある。覚えやすく、発音しやすいものがふさわしい。

## 変更点

iframe で作るの難しいと考えたため、新たな方法を試す必要がある。一番手間はかかるが簡単に作れるのは青空文庫からファイルをダウンロードし、自サイトに読み聞かせサイトを作ることである。ダウンロードするファイルの選定だが、青空文庫内で掲載されているアクセスランキングを利用する予定である。

## 10/11~10/17 の週の予定

システムの構築がメインとなる作業である。

青空文庫や google books 等の電子図書館を調査し、自サイトに取り入れるべき UI を見極める。

また、その成果をサイトに反映させる。

Custom Search API を使用して、google の検索バーの下に検索結果を表示させる。JSON データ形式で検索結果を取得する方式のため、検索結果を音声で選択することが可能になる。

## Custom Search API の利用方法

- ・body タグに下記のコードを記述することで Custom Search API の google 検索窓が表示される。

下記の図はコードの取得方法である。

```
<script async src="https://cse.google.com/cse.js?cx=3c473b213e0e4c131"></script>
```

<div class="gcse-search"></div>

## Programmable Search

新しい検索エンジン

▼ 検索エンジンの編集

青空

設定

デザイン

検索機能

統計情報とログ

▶ ヘルプ

ヘルプ フォーラムを表示  
(質問を入力)

フィードバックを送信

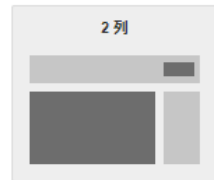
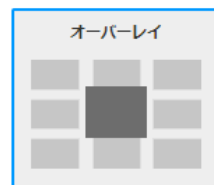
レイアウト

テーマ

カスタマイズ

サムネイル

検索エンジンのレイアウトを設定します。変更を公開するには、[保存] ボタンをクリックします。詳細



保存

保存してコードを取得

## Programmable Search

### 新しい検索エンジン

#### ▼ 検索エンジンの編集

青空

- 設定
- デザイン
- 検索機能
- 統計情報とログ

#### ▼ ヘルプ

- ヘルプセンター
- ヘルプフォーラム
- ブログ
- ドキュメント
- 利用規約
- ヘルプフォーラムを表示  
(質問を入力)
- フィードバックを送信

### コードを取得

HTML の例を表示

次のコードをコピーして、検索ボックスと検索結果の両方をレンダリングしたいサイトの <body> セクションの <div> 要素に貼り付けます。

```
<script async src="https://cse.google.com/cse.js?cx=3c473b213e0e4c131"></script>
<div class="gcse-search"></div>
```

カスタム検索エンジンに関する詳細なドキュメントを参照すると、検索 UI をさらにカスタマイズしたり、ページごとのカスタマイズを追加したりできるようになります。

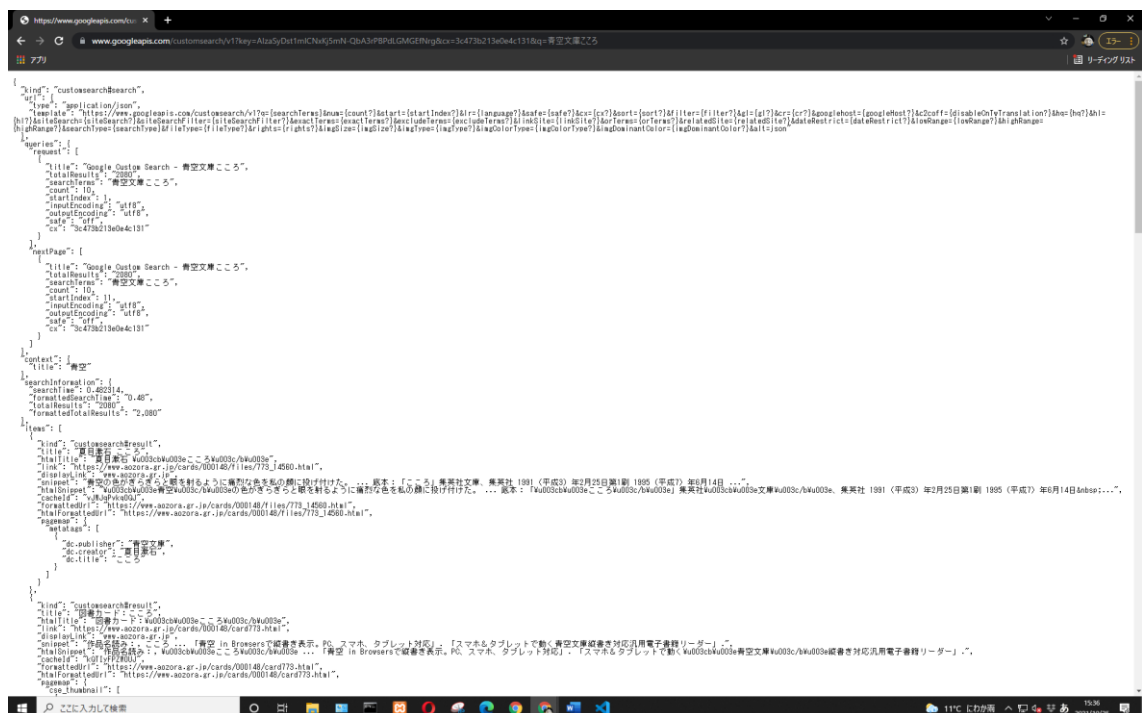
注: 互換性を最大限確保するには、ページに HTML5 DOCTYPE とビューポートを指定する必要があります。  
<!DOCTYPE html>  
<meta name="viewport" content="width=device-width, initial-scale=1">

- 検索結果をブラウザで取得する。

[https://www.googleapis.com/customsearch/v1?key=\[API キー\] &cx=\[検索エンジン ID\]&q=\[検索する内容\]](https://www.googleapis.com/customsearch/v1?key=[API キー] &cx=[検索エンジン ID]&q=[検索する内容])

例.

<https://www.googleapis.com/customsearch/v1?key=AIzaSyDst1mICNxKj5mN-QbA3rPBPdLGMGEfNrg&cx=3c473b213e0e4c131&q=青空文庫こちら>



参考文献

[1] 青空文庫

<https://www.aozora.gr.jp/>

[2] セイムオリジンポリシー - MDN

[https://developer.mozilla.org/ja/docs/Web/Security/Same-origin\\_policy](https://developer.mozilla.org/ja/docs/Web/Security/Same-origin_policy)

[3] スクレーピング - コトバンク

<https://kotobank.jp/word/web%E3%82%B9%E3%82%AF%E3%83%AC%E2%80%95%E3%83%94%E3%83%B3%E3%82%B0-1689955>

[4] サーバーサイドプログラミング - MDN

[https://developer.mozilla.org/ja/docs/Learn/Server-side/First\\_steps/Introduction](https://developer.mozilla.org/ja/docs/Learn/Server-side/First_steps/Introduction)