

2020 年度 卒業論文

コーヒー抽出に関する音声認識可能な Web レシピの開発

指導教員 須田 宇宙 准教授

千葉工業大学 情報ネットワーク学科

須田研究室

1632130 氏名 肥田雄也

提出日 2020 年 1 月 30 日

# 目次

第 1 章 緒言	1
第 2 章 コーヒーについて	2
2.1 コーヒーとは .....	2
2.2 コーヒーの起源と伝搬の歴史 .....	2
2.3 コーヒーの流行について .....	3
2.3.1 ファーストウェーブ .....	3
2.3.2 セカンドウェーブ .....	3
2.3.3 サードウェーブ .....	3
第 3 章 抽出方法と抽出器具について	4
3.1 抽出方式 .....	4
3.1.1 浸漬式 .....	4
3.1.2 透過式 .....	4
3.1.3 高圧抽出式 .....	4
3.1.4 真空濾過式 .....	4
3.2 抽出器具 .....	6
3.2.1 コーヒープレス .....	6
3.2.2 プアオーバー .....	7
3.2.3 エスプレッソ .....	8
3.2.4 ソロフィルター .....	9
3.2.5 サイフォン .....	10
3.2.6 クレバードリッパー .....	11
3.2.7 エアロプレス .....	12
第 4 章 コーヒーの学習について	13
4.1 コーヒー抽出の学習方法 .....	13
4.2 コーヒーの学習が可能な Web サイトについて .....	13
4.3 既存の音声認識可能な Web レシピについて .....	13
第 5 章 プログラミング言語と音声認識について	15
5.1 HTML・CSS とは .....	15
5.2 音声認識とは .....	15
5.2.1 音声認識の仕組み .....	15
5.2.2 WebSpeechAPI .....	15
第 6 章 本研究で開発する Web レシピの概要	17
6.1 実装機能 .....	17

6.1.1	抽出器具選択画面	18
6.1.2	レシピ閲覧画面	19
6.2	音声認識の実装について	20
第 7 章 結言		23
第 8 章 参考文献		25
付録 A 作成したプログラム		26

## 図目次

3.2-1	コーヒープレス . . . . .	6
3.2-2	プアオーバー . . . . .	7
3.2-3	エスプレッソマシン . . . . .	8
3.2-4	ソロフィルター . . . . .	9
3.2-5	サイフォン . . . . .	10
3.2-6	クレバードリッパー . . . . .	11
3.2-7	エアロプレス . . . . .	12
6.1-1	ページ構成 . . . . .	17
6.1-2	抽出器具選択画面 . . . . .	18
6.1-3	レシピ閲覧画面の構成 . . . . .	19
6.1-4	レシピ閲覧画面：抽出 . . . . .	20
6.2-1	音声認識の流れ . . . . .	22

## 表目次

5.2-1 音声認識の方法論の変遷(河原,2018) . . . . .	16
--------------------------------------	----

# 第1章 緒言

「悪魔のように黒く、地獄のように熱く、天使のように純粋で、愛のように甘美である。」これはフランスの外交官である、シャルル＝モーリス・ド・タレーラン＝ペリゴールが遺したコーヒーの名言である。コーヒーはただの飲料物でありながらも、長い年月をかけて愛され、世界の人々を魅了してきた。それは日本も例外ではない。江戸時代初期に長崎の出島に持ち込まれた際は、一部の人間しか飲用はできなかつたが、明治時代に文明開化が起きるとみるうちに一般層に普及していった。全日本コーヒー協会の統計資料「日本のコーヒー消費と株価」のグラフでは、2018年のコーヒー消費量がおよそ470千トンを記録していることがみてとれる[1]。

コーヒーの流行は、ファーストウェーブと呼ばれる大量消費時代から始まり、その後、風味に重きが置かれたセカンドウェーブが到来する。そして近年では、高品質なコーヒーをより良い淹れ方で味わうサードウェーブコーヒーが流行している。

美味しいコーヒーを淹れるためには抽出器具に合わせたテクニックが必要であり、そこから生まれる創意工夫がコーヒーの楽しみ方の一つである。誰しもが気軽にコーヒーを淹れられるようになれば、自然と新たなコーヒーの楽しみ方が生まれるようになる。

しかし抽出手順や、器具ごとの特徴の違いなどから学習の難易度が高くなり、コーヒーを淹れることの敷居も高くなっていることが問題点である。また、実際の抽出中は、両手が塞がっていたり手が濡れているなど、レシピブックを捲ることに対する障害も多い。クックパッドや macaroni のような、音声認識でレシピを閲覧できる Web サイトは複数存在するが、コーヒーの抽出に特化したサイトは無いことも問題点である。

そこで、様々な抽出器具のレシピが閲覧可能であり、音声認識によってページ移行が可能な Web レシピがあれば、問題点の改善に繋がると考えた。本研究では上記の Web レシピを作成することを目的とする。

## 第2章 コーヒーについて

### 2.1 コーヒーとは

コーヒーとは、コーヒーノキという樹木から採取される種子を焙煎し、お湯等で成分を抽出した飲料物である。主に北回帰線と南回帰線を挟むコーヒーベルトと呼ばれる地域で栽培されており、数百種類の品種が存在する。商業生産としては、高品質であるが栽培の難しいアラビカ種がおよそ 60 % 前後を占めている。アラビカ種は、スペシャルティコーヒーと呼ばれる高品質コーヒーの主要製品であり、多くのカフェで提供されている。残りの 40 % はロブスタ種が占めており、こちらは品質はアラビカ種に劣るが耐病性に優れ、大量生産向きであり缶コーヒーやインスタントコーヒー等に用いられる。

### 2.2 コーヒーの起源と伝搬の歴史

コーヒーノキ（アラビカ種）はエチオピアのアビシニア高原にて発見された。その後、アラビアに伝播しオランダの貿易商人達の手によってアフリカやアジアへと広がっていき、商業用生産が活発になる。アフリカやアジア各国にコーヒーが広がり、フランスに至ると、とある海軍兵士がコーヒーの苗木を当時フランス領であったマルティニク島に持ち出すことになる。こうしてラテンアメリカにもコーヒーは広がり、温暖でコーヒーの栽培に特に適した北回帰線と南回帰線を挟むコーヒーベルトにおいて栽培は進んでいった。現代でも、コーヒーベルト各国はコーヒーのシェアの多くを占めている。グアテマラ・ブラジル・コロンビア・スマトラなどは、コーヒーを普段飲まない人でも聞き馴染みのある生産地である。高品質なコーヒーを淹れる上で、アラビカ種を使用することはもはや前提といっても過言ではないシェアを誇るが、近年では、耐病性向上の観点からハイブリット種の開発も行われている。

## 2.3 コーヒーの流行について

ここでは、コーヒーの流行を時代別に説明する際に使われる、ウェーブという表現について説明する。コーヒーはその時代ごとに求められる姿を変え、ファーストウェーブ（第一の波）、セカンドウェーブ（第二の波）、サードウェーブ（第三の波）へとシフトしていく。このウェーブという表現は、明確な時間の括りを示すものではないが、流行りの移り変わりとしてコーヒーを語る上で多く用いられる。

### 2.3.1 ファーストウェーブ

ファーストウェーブ（第一の波）の期間は、1800年代後半～1960年頃と言われている。この流行はコーヒーの大量消費時代であり、真空保存パックの発明や、インスタントコーヒーが発売された時代でもある。真空保存パックの発明は、コーヒーの保存期間を劇的に延長し、地方だけでなく大都市でも新鮮なコーヒーが飲めるようになった。インスタントコーヒーの発明は、コーヒーを楽しむことをより手軽なこととした、第一次・第二次世界大戦においても多くの兵士が愛飲したと言われている。しかしその反面、この時代のコーヒーは風味がとても乏しく、粗悪品が多く出回る時代でもあった。

### 2.3.2 セカンドウェーブ

セカンドウェーブ（第二の波）の期間は、1960年頃～2000年頃と言われている。ファーストウェーブの際に、一般的かつ手軽な飲み物として普及が進んだコーヒーであったが、普及が進むとともに劣悪な品質が問題化した。そして、STARBUCKSなどのシアトル発祥のカフェを代表とする、味にフォーカスした流行が起こる、それがセカンドウェーブである。コーヒーそのものだけでなく、カフェラテなどのエスプレッソドリンクもシアトル系カフェの台頭とともに普及した。

### 2.3.3 サードウェーブ

サードウェーブ（第三の波）の期間は、2000年頃～現在も続くと言われている。セカンドウェーブの際に、品質についての意識が向上したことを皮切りに、高品質かつそのコーヒーの生まれや、農園、淹れ方、加工法などが重視される流行が訪れる事となる。それがサードウェーブコーヒーである。コーヒーの抽出技術を競う大会や、カップオブエクセレンスと呼ばれるその年のコーヒーで最高品質の物を決める評議会が開催されるなど、一杯の価値にフォーカスされている流行である。また、昔から日本のカフェで行われてきたハンドドリップは、日本でこそ当たり前の光景だったものの、海外では目新しいものであり、この流行とともに注目を集めていくこととなる。

## 第3章 抽出方法と抽出器具について

### 3.1 抽出方式

コーヒーの抽出器具は、それぞれ特有の抽出方式を持っているが、大きく区分したものが以下のものとなる。

#### 3.1.1 浸漬式

浸漬式は、名前の示す通りコーヒー豆をお湯や水に浸し、適切な時間漬けておくことで、風味を引き出す抽出方式である。他の方式に比べ、コーヒーの油分をしっかりと抽出できるため、豆本来の味わいを表現しやすいことがメリットである。コーヒーは抽出を過剰に行った場合、雑味やエグ味まで抽出されてしまうため、抽出にかかる時間を適切に管理することが、浸漬式では特に重要である。比較的長時間の抽出になるため、粗挽きの豆を使用するのが一般的である。特に水出しの場合、お湯に比べ抽出スピードが遅いため、丸一日時間を要する場合もある。

#### 3.1.2 透過式

透過式とは、ペーパーフィルターなどを介し、豆とお湯が分離しサーバーに落とす抽出方式である。浸漬式に比べ、粉末感がほとんど無く、あっさりめの味わいに仕上がる事がメリットである。抽出者により味わいが大きく変わることも特徴であり、自由な表現ができるが、テクニックを身につけるための時間が長いことがデメリットでもある。

#### 3.1.3 高圧抽出式

高圧抽出式は、お湯を浸したコーヒー豆に高い気圧をかけることにより、一気に味を抽出する方式である。高い圧力をかけるため、風味を濃縮した味わいになりやすいことが、他の抽出方法にないメリットである。その反面、高圧をかけるこの抽出方式は、器具自体が大きく機械式であることがほとんどのため、事前知識やメンテナンスに割く時間が多いうことがデメリットである。

#### 3.1.4 真空濾過式

真空濾過式はサイフォンに代表される抽出方式であり、意図的に準真空状態を作り出し抽出を行う方式である。この方式を採用している器具は少ないため、メリットやデメリットは後述するサイフォンという器具の詳細と同様となる。同じく真空状態を意図的に利用する器具で、CLOVERと呼ばれる浸漬式と真空バキュームを組

み合わせた器具も存在する。

## 3.2 抽出器具

### 3.2.1 コーヒープレス

コーヒープレスは、浸漬式に分類される抽出器具である。実際の写真を図 3.2-1 に示す。豆をお湯に漬け込み、味を染み出させたのちに、フィルターで濾すことで抽出を行う。仕上がりは、多少の粉末感を残すものの、濃厚でまろやかな味わいである。ステンレスフィルターを使用するため、コーヒーオイルをそのまま味わえること、コーヒー豆本来の味が楽しめることがメリットである。



図 3.2-1: コーヒープレス

### 3.2.2 プアオーバー

プアオーバーは、透過式に分類される器具であり、別名ハンドドリップとも呼ばれている。実際の写真を図 3.2-2 に示す。フィルターの上に挽いた豆を置き、お湯を注ぐことによって、下部のグラスサーバーにコーヒーを抽出していく。ペーパーフィルターを介しているため、出来上がりは、粉末感は殆どなくクリーンな味わいになりやすいメリットがある。また、お湯の注ぐタイミングや蒸らしの時間の掛け方などにより、様々な流派が存在し、淹れた人によって大きく風味が変化する事も特徴の一つである。



図 3.2-2: プアオーバー

### 3.2.3 エスプレッソ

エスプレッソは、高压抽出式に分類されるコーヒーの飲み方である。実際に抽出する器具はエスプレッソマシンと呼ばれており、水蒸気やピストンなどで圧力をかけ、短い時間で抽出を行う。実際の器具の写真を図 3.2-3 に示す。自動式・半自動式・ピストン式など様々なエスプレッソマシンが存在するが、使用者自身でフィルターに豆を押し込み、機械によって気圧をかける半自動式が最も一般的である。仕上がりは極めて濃厚であり、そのままの状態で飲む以外にも、ミルクを追加しカフェラテ・カプチーノとして提供されることが多い。



図 3.2-3: エスプレッソマシン

### 3.2.4 ソロフィルター

ソロフィルターは、カフェ等ではあまり提供されず、明確な抽出方式はないが、コーヒーを初めて淹れる人でも簡単に抽出でき、入門用に最適な器具の一つである。実際の器具の写真を図 3.2-4 に示す [2]。ステンレスフィルターの上に挽いたコーヒー豆を置いた後、複数の極細の穴があるパツツに規定量のお湯を注ぐだけで、適切な湯量が豆に注がれ続ける。容量は 1 杯分のコーヒー豆しか入らないため、大人数分のコーヒーを抽出するには向きだが、粉末感の殆どない高品質なコーヒーを手軽に味わえるため、コーヒーを学び始めの人や、朝の忙しい時間でも簡単に抽出することができる。



図 3.2-4: ソロフィルター

### 3.2.5 サイフォン

サイフォンは、真空濾過式に分類される抽出器具である。科学の実験器具のような形状をした抽出器具であり、数ある抽出器具の中でも特に見栄えの良い器具として、注目を集めている。実際の器具の写真を図 3.2-5 に示す [3]。お湯を熱したことによる蒸気圧を使用し、真空に近い状態を作り出すことで、コーヒーが器具の内部を上下し、フィルターを通して濾過される。日本の喫茶店でも多く取り入れられていたため、馴染みの深い抽出器具もある。他の器具に比べ、風味が豊かになりやすい。



図 3.2-5: サイフォン

### 3.2.6 クレバードリッパー

クレバードリッパーは、浸漬式と透過式を合わせた抽出器具である。実際の器具の写真を図 3.2-6 に示す [4]。浸漬式と同様にコーヒー豆をお湯に浸し、ペーパーフィルターを透過することで、豆とお湯を分離する。味わいは、浸漬式のマイルドさと、透過式のスッキリさを併せ持っている。一度に多くのコーヒーを抽出できるほか、プアオーバーのような複雑な工程を踏まないため、比較的簡単に抽出を行うことができる。



図 3.2-6: クレバードリッパー

### 3.2.7 エアロプレス

エアロプレスは、その名の通り空気圧を使用した抽出器具である。実際の器具の写真を図 3.2-7 に示す [5]。開発されたのは 2005 年であり、数多くの抽出器具の中でも特に直近に提案された器具である。注射器のような形状をしており、利用者自身が器具を押し込むことによって、空気圧を調整し抽出を行う。抽出時間が短いこと、味わいが濃厚であることが特徴である。2005 年という直近の開発でありながら、様々なカフェで提供されている他、エアロプレスに特化した、抽出技術を競う大会が開催されたりと、シェアを順調に伸ばしている。



図 3.2-7: エアロプレス

## 第 4 章 コーヒーの学習について

### 4.1 コーヒー抽出の学習方法

コーヒー抽出の学習方法は、他の学問と同様に様々な学習スタイルが存在する。

近年では、多くの個人経営カフェや、チェーン店においてもコーヒーセミナーという形で、実践を伴った学習ができる。このセミナーという形態のメリットは、実際の抽出を直近で見れることと、プロが積み上げてきた抽出メソッドを知ることができることにある。それらの情報は、Web 上に上がる事が稀なため、コーヒー学習をさらに深めたい人に需要が高い。

個人学習においては、抽出に関する事に主題を置いた書籍が多数出版されている他、インターネット上で抽出レシピやコツなどを簡単に閲覧することができる。特に Web ページを用いた学習方法は、どの学習方法よりも気軽に学習できることがメリットである。また、様々な抽出器具や抽出方法を閲覧することができるため、自分にあったコーヒーの楽しみ方を模索しやすいというメリットもある。

### 4.2 コーヒーの学習が可能な Web サイトについて

インターネット上の Web サイトでは、コーヒーの基本知識から、実際の抽出方法に至るまで多くの情報が閲覧できる。個人で運営されている Web サイトだけでなく、コーヒー業界の大手である STARBUCKS や、UCC、TULLY'S などの企業も自社ページで、器具ごとの抽出手順や実際の抽出動画を公開している。STARBUCKS は、自社のコーヒー豆の紹介をメインにしており、抽出に関しては、複数の器具の説明を図と文字で説明している [6]。TULLY'S は、他社と同様に図解で抽出方法を説明しているほか [7]、自社ページ以外にも YouTube で美味しい淹れ方の動画を掲載している [8]。UCC は、コーヒーの抽出に限らず、豆の選び方や焙煎の種類など、他の企業に比べ包括的で、辞典のような Web ページを掲載している [9]。抽出に関しては、抽出手順が図解されているほか、動画で確認をすることができる。また、他社に比べて理論的な説明が多くみられる。

### 4.3 既存の音声認識可能な Web レシピについて

既存の音声認識可能な Web レシピで特に有名なものとして、クックパッドと macaroni が挙げられる。クックパッドは、料理レシピの閲覧・投稿が可能であり、調理手順をステップごとに確認することができる。元々は音声認識は実装されていなかったが、Amazon が発売している、スマートスピーカー「Amazon Echo」のクックパッドスキルを使用することで、音声認識が使用できる。機能は主にレシピの検索や読み上げであり、画面ではなく音声を聞きながらの調理をすることになる。macaroni は、スマートフォンアプリで使用できる Web レシピである。こちらは、調

理動画を閲覧する際に音声認識が可能であり、「サイセイ」、「マキモドシ」などの発音で、動画の再生をコントロールできる。

双方ともに、料理レシピがメインコンテンツであり、コーヒーに関するレシピはほとんど掲載されていない。

## 第 5 章 プログラミング言語と音声認識について

### 5.1 HTML・CSS とは

HTML(Hyper Text Markup Language) とは、プログラミング言語の 1 つである。主に Web ページを作成する際に使用される言語である、単なる文字の集まりである文章に、見出し・ボタン・リストなどの役割を指定し、コンピュータに認識させることができる。現存するほとんどの Web ページが HTML によって構造を作られている。

CSS(Cascading Style Sheets) とは、HTML に記載された役割を持つ文章に対し、サイズ指定を行うことや、文字色の変更、配置を調整することができる言語である。HTML 単体の記述では、単調な文章表記やレイアウトになってしまふため、多くの Web サイトで CSS を用いたデザイン調整が行われている。使用方法は、HTML ファイルから別の CSS ファイルを読み込むというだけでなく、HTML ファイル中に直接 CSS を書くこともできるなど、用途に合わせた柔軟な記述ができるようになっている。

### 5.2 音声認識とは

音声認識とは、人間が発する声をコンピュータ上で解析を行い、テキストとして認識する技術である。

#### 5.2.1 音声認識の仕組み

音声認識の仕組みは利用するサービスによって異なるほか、その時代によっても音声認識の方法論が移り変わっている。河原（2018）の、音声認識の方法論の変遷にて記載されている表を、表 5.2-1 に示す [10]。この表 1 を見ると現在の音声認識の方法論は 4.5 世代にあたり、ニューラルネットワークの利用が主流になっていることがわかる。ニューラルネットワークはいわば機械学習の類であり、それらを応用することで音声認識を成し遂げている。実際に、Google 社の手がける音声認識サービス「Google Cloud Speech API」においてもニューラルネットワークモデルが利用されていることが述べられている [11]。

#### 5.2.2 WebSpeechAPI

WebSpeechAPI とは、JavaScript API の 1 つである。JavaScript と Web ブラウザのみで、音声の合成や音声認識を可能にする。音声の合成は入力した文字を機械側に発音させる機能であり、音声認識は音声をテキストとして入力する機能である、本 Web アプリケーションの音声認識はこの WebSpeechAPI を使用している。

表 5.2-1: 音声認識の方法論の変遷 (河原,2018)

第 1 世代	1950 ~ 1960 年代	ヒューリスティック
第 2 世代	1960 ~ 1980 年代	テンプレート (DP マッチング, オートマトン)
第 3 世代	1980 ~ 1990 年代	統計モデル (GMM-HMM, N-gram)
3.5 世代	1990 ~ 2000 年代	統計モデルの識別学習
第 4 世代	2010 年代	ニューラルネット (DNN-HMM, RNN)
4.5 世代	2015 年~	ニューラルネットによる End-to-End

## 第 6 章 本研究で開発する Web レシピの概要

### 6.1 実装機能

コーヒーの抽出は目的に合わせて抽出器具を使い分けるため、それぞれの特徴を踏まえた上で器具の選定を行うことが重要である。そのため、本研究で開発した Web レシピのページは、「抽出器具選択画面」、「レシピ閲覧画面」の二段階で構成している。抽出器具選択画面において、利用者はそれぞれの器具の特徴や風味の違いを理解し、レシピ閲覧画面で実際の抽出をサポートする。本 Web レシピのページ構成を図 6.1-1 に示す。

本研究で実装する機能は、「コーヒーの専門性の高い Web レシピの機能」と「音声認識」である。「コーヒーの専門性の高い Web レシピの機能」は、コーヒーの抽出器具ごとの情報や、その組み立て方および抽出手順である。上記に述べたとおり、抽出をする際に使用する器具の特徴をあらかじめ理解していることはとても重要である。抽出手順においても器具ごとに異なるため、そのステップごとに個別のページを用意し、順に解説を行う。「音声認識」は、抽出ステップ間の移動を行う際に使用する。実際の抽出中は両手が塞がることが多く、ページ移動のボタンを押せないことが予期されるため、ボタンの他に音声認識でもページを移動できるようにする。

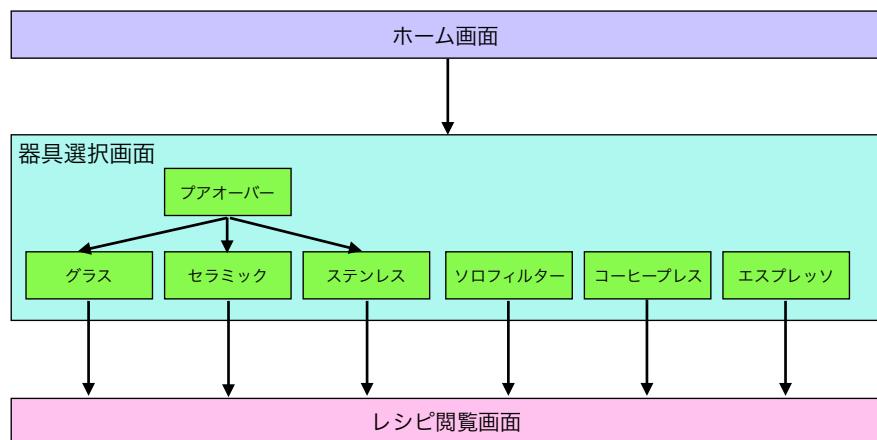


図 6.1-1: ページ構成

### 6.1.1 抽出器具選択画面

「抽出器具選択画面」では、レシピを閲覧する抽出器具を選択する。実際の抽出器具選択画面を図 6.1-2 に示す。

この画面では、抽出方式の異なる 3 つの代表的な抽出器具と、ソロフィルターを含めた 4 種類を選択できる。プアオーバー（透過式）、コーヒープレス（浸漬式）、エスプレッソ（高圧抽出式）は、日本で特に馴染みの深いコーヒーの抽出方法であり、様々なカフェで提供されていることから選択した。ソロフィルター（より手軽な入門向け）は、カフェ等ではありませんが、コーヒーを初めて淹れる人でも簡単に抽出でき、入門用の器具の一つとして選定した。

「レシピを見る」を選択することで、「レシピ閲覧画面」へと移動する。

「レシピ閲覧画面」では、実際に器具ごとのコーヒー抽出レシピを閲覧できる。



図 6.1-2: 抽出器具選択画面

### 6.1.2 レシピ閲覧画面

「レシピ閲覧画面」では、実際に器具ごとのコーヒー抽出レシピを閲覧できる。レシピ閲覧画面の構成を図 6.1-3 に、実際のレシピ閲覧画面を図 6.1-4 に示す。

ページは抽出ステップごとに分割している。1つのステップ毎に、個別の画像と説明文を表示し、記載されている指示に従うことで、抽出を行うことができる。ステップ間は、画面上部のボタンを押すことで移動が可能であるほか、音声認識も実装している。画面下部の「音声認識を開始する」ボタンを押すことにより、音声認識処理が開始し、「次へ」・「戻る」の発音で、ステップ間の移動が可能である。

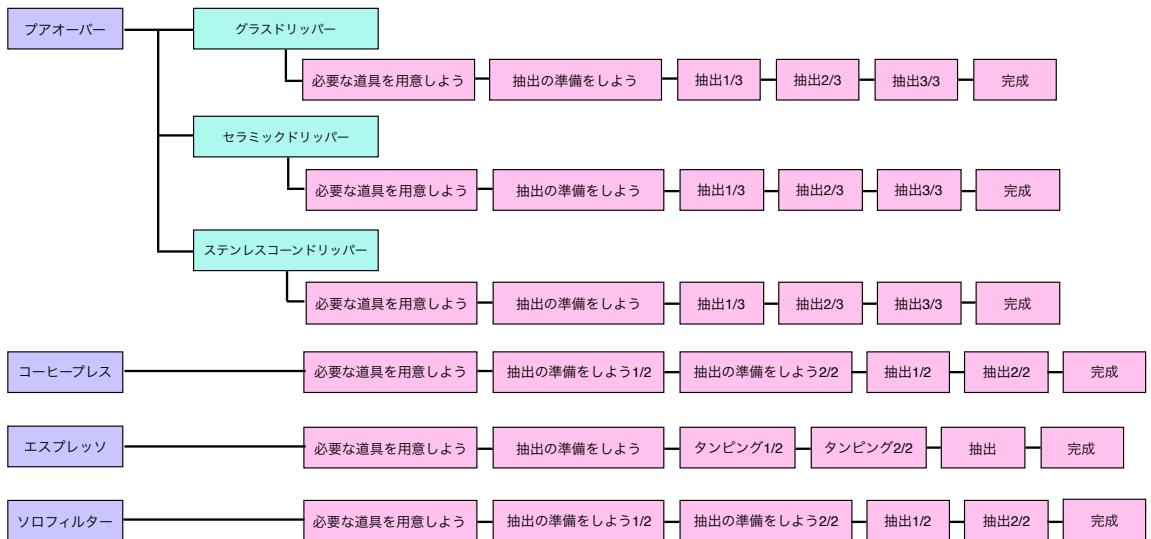


図 6.1-3: レシピ閲覧画面の構成

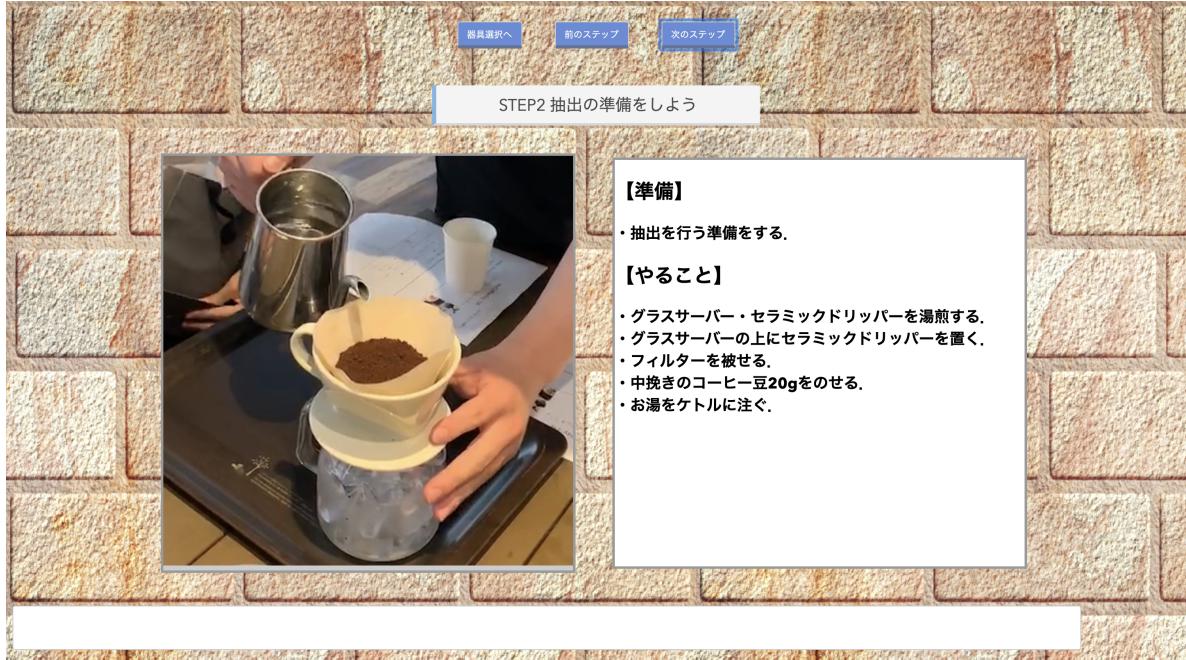


図 6.1-4: レシピ閲覧画面：抽出

## 6.2 音声認識の実装について

はじめに、音声認識機能の全体の流れを説明する。図 6.2-1 に示すように、音声認識によって得られた音声データは、WebSpeechAPI によって外部処理が行われ、テキストに変換される。変換されたテキストが特定の文字「次へ」、「戻る」であった場合、それぞれに合わせた関数を呼び出し、ページ移動を行う。

音声認識の実装には 2 つの JavaScript ファイル「seramic.js」と「page.js」を使用する。「seramic.js」は、主に音声認識を継続的に実行する JavaScript ファイルである。「page.js」は、ページ移動を実装する JavaScript ファイルである。

「seramic.js」は、宗定(2014)の記事に述べられているソースコードを元に作成した[12]。元のソースコードは音声認識を継続して行い、認識した音声をテキストとして表示するプログラムである。本開発ではボタンの押下、または特定の音声によって、ページ移動を行う機能を追加した。

音声認識が行われると、`recognition.onresult` が処理を始める。`function (e)` の `e` には、実際の音声認識データが含まれている。

```
recognition.onresult = function (e) {
```

認識された音声は、テキストとして変数 `interimText` に格納される。`e.results` は過去の認識データも含めた、全てのデータが含まれている。`[e.results.length-1]` が最新のデータになっており、`[0]` の中の `transcript` 内に、実際に認識したワードが格納されている。

```
interimText = e.results[e.results.length-1][0].transcript;
```

変数 interimText に格納されたテキストが「次へ」の場合、page.next(); が実行され、「戻る」の場合、page.prev(); が実行される。

```
if (interimText === 次へ "") page.next();
else if (interimText === 戻る "") page.prev();
```

また、図 6.1-4 の上部にある「前のステップ」「次のステップ」ボタンを押下することでも同様の関数が実行される。

```
document.querySelector('#next_button').addEventListener('click', () => { page.next(); });
document.querySelector('#prev_button').addEventListener('click', () => { page.prev(); });
```

続いてページ移動の説明をする。前述の通り、ページの移動は「page.js」が行う。number は 1 枚目のページを 0, 2 枚目のページを 1 とした時の、現在のページ番号である。this.page size は、ページの総数であり、number と数値の差を合わせるため、1 を引く。next() では、現在のページ数がページの総数より少なかった場合、prev() では、0 より現在のページ数が大きかった場合、処理が起こる。this.pages[number] には、現在のページの id が入っており、next() では次のページの id を、prev() では前のページの id を change() に渡している。

```
next() {
    let number = this.pages.indexOf( this.now );
    if( number < this.page_size-1 ) {
        let a = document.querySelector( this.now );
        this.change( this.pages[number+1] );
    }
}
prev() {
    let number = this.pages.indexOf( this.now );
    if( 0 < number ) {
        let a = document.querySelector( this.now );
        this.change( this.pages[number-1] );
    }
}
```

change() は画面の表示範囲を設定することで、ページ移動を行なっている。下記のコードの 2-5 行目において、全てのページを非表示にし、6 行目にて、指定の id のページのみを表示させている。最後の 7 行目では、現在のページ id を記録している。

```
change( page ) {
    let pages1 = document.querySelectorAll('.page');
    for( let p of pages1 ) {
        p.style.display = "none";
    }
    document.querySelector( page ).style.display = "block";
    this.now = page;
}
```

前述の通り，ページの移動は，同 HTML ファイル内の表示範囲を変えることで実装した. そうすることで，繰り返しブラウザからマイク使用の許可を求められることが無いほか，スムーズな画面切り替えを実現している. なお，この音声認識は Google Chrome のみで対応している.

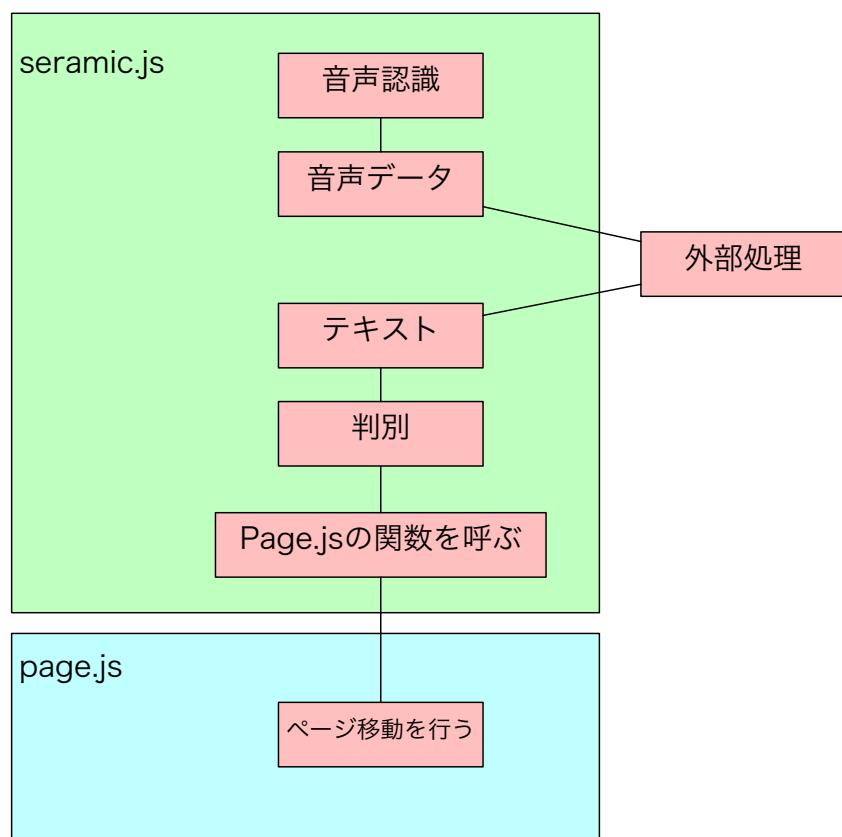


図 6.2-1: 音声認識の流れ

## 第7章 結言

コーヒーは、1章にて述べた通り、長い年月をかけて愛され、世界の人々を魅了してきた。また、その生産に当たっても、農家・バイヤー・焙煎士・加工者など多くの人々の手に渡りながら世界を巡り、最終的に消費者の元に訪れる。そのような長い過程を経たとしても、最終的には抽出次第で味は如何様にも変化する。難しいと感じる人も多いだろうが、そこがコーヒーの魅力でもあり、人々を惹きつける要因の1つだろう。

しかし、抽出手順や器具ごとの特徴の違いなどから学習の難易度が高くなり、コーヒーを淹れることの敷居も高くなっている実情は、その楽しみを阻害するだけでなく、新たなコーヒーの楽しみ方を模索する機会を失ってしまっている問題点がある。また実際の抽出中は両手が塞がってしまうのに対し、現存のWebレシピでは音声認識を実装していない、もしくはコーヒーの専門性が低いという問題点があった。

そこで本研究では、様々な抽出器具のレシピが閲覧可能であり、音声認識によってページ移行が可能なWebレシピを開発し、問題点の解決に取り組んだ。

今回の開発で、気軽にコーヒー抽出を行うことのサポートとして第一歩を踏み出せたと考えている。今後このような、抽出アシストツールがより普及し、コーヒーを抽出できるようになることを期待している。

## 謝辞

本研究の遂行及び本論文の作成にあたり，須田研究室の仲間に多くの手助けを頂きました，深く感謝の意を表します。そして，本論文の作成にあたり，多大なる御指導及び御助言を頂きました須田宇宙准教授に深く感謝の意を表します。

## 第 8 章 参考文献

### 参考文献

- [1] 全日本コーヒー協会, “日本のコーヒー消費と株価”, [http://coffee.ajca.or.jp/wp-content/uploads/2019/04/data00c\\_2019\\_04.pdf](http://coffee.ajca.or.jp/wp-content/uploads/2019/04/data00c_2019_04.pdf)
- [2] スターバックス コーヒー ジャパン, “スターバックス ソロフィルター”, <https://www.starbucks.co.jp/brewing/dripper/4524785211094/>
- [3] HARIO, “テ ク ニ カ 3 人 用”, <https://www.hario.com/seihin/productdetail.php?product=TCA-3>
- [4] LILO COFFEE ROASTERS, ‘E.K.INT’L CLEVER DRIPPER・クレバードリッパー L”, <https://lilocoffee.thebase.in/items/158429103>
- [5] 小川珈琲, “エアロプレスコーヒーメーカー”, <https://www.oc-ogawa.co.jp/products/tools/aeropress/>
- [6] スターバックス コーヒージャパン, “おいしいコーヒーをいれるコツ”, <https://www.starbucks.co.jp/howto/coffee/skill.html>
- [7] タリーズコーヒージャパン, “タリーズ流 コーヒーの美味しい淹れ方”, [https://www.tullys.co.jp/quality/coffee\\_tips/coffee\\_tips06.html](https://www.tullys.co.jp/quality/coffee_tips/coffee_tips06.html)
- [8] Youtube , “タリーズ流 ハンドドリップの淹れ方！ ホットコーヒ一篇”, <https://www.youtube.com/watch?v=io9JiBfEySg>
- [9] UCC, “おいしいコーヒーの淹れ方”, <https://www.ucc.co.jp/enjoy/brew/>
- [10] 河原達也, “音声認識技術の変遷と最先端”, <http://sap.ist.i.kyoto-u.ac.jp/members/kawahara/paper/ASJ18-7.pdf>
- [11] google , “Cloud Speech-to-Text”, <https://cloud.google.com/speech-to-text/?hl=ja>
- [12] 宗定洋平, “[HTML5] Web Speech API に入門”, <https://www.yoheim.net/blog.php?q=20140701>

## 付録A 作成したプログラム

### index.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>コーヒーレシピブック</title>
6          <link rel="stylesheet" href="./css/index.css">
7          <script type="text/javascript" src="./footerFixed.js"></script>
8      </head>
9
10     <body>
11         <div class="container">
12             <div class="item">
13                 <h1>コーヒーレシピブック</h1>
14             </div>
15         </div>
16
17         <div class="container">
18             <div class="item2">
19                 <a href="./kigusentaku.html" class="btn-animation-02">
20                     <span>Click to START!<span></a>
21                 </div>
22             </div>
23
24
25
26     </body>
27 </html>
```

## kigusentaku.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>コーヒーレシピブック</title>
6          <link rel="stylesheet" href=".css/seramic.css">
7          <script type="text/javascript" src="onsei.js"></script>
8      </head>
9
10     <body>
11         <header>
12             <div class="item13">
13                 <h1>閲覧したい器具を選択してください！</h1>
14             </div>
15         </header>
16         <div class="main">
17
18             <button onclick="location.href='./index.html'" class="btn-square">
19                 戻る</button>
20             <div class="container">
21                 <div class="item4">
22                     
23                     <h2>プアオーバー</h2>
24                     <a href=".pourover.html" class="btn-square">この器
25                         具のレシピを見る</a>
26                     <ul style="text-align: left">
27                         <li>さっぱりな味わい</li>
28                     </ul>
29                 </div>
30                 <div class="item4">
31                     
32                     <h2>コーヒープレス</h2>
```

```
31          <a href="../press.html" class="btn-square">この器具  
のレシピを見る</a>  
32          <ul style="text-align: left">  
33              <li>濃厚な味わい</li>  
34          </ul>  
35      </div>  
36      <div class="item4">  
37            
38          <h2>エスプレッソ</h2>  
39          <a href="../espresso.html" class="btn-square">この器具  
のレシピを見る</a>  
40          <ul style="text-align: left">  
41              <li>ミルクを入れてラテに</li>  
42          </ul>  
43      </div>  
44      <div class="item4">  
45            
46          <h2>ソロフィルター</h2>  
47          <a href="../solo.html" class="btn-square">この器具の  
レシピを見る</a>  
48          <ul style="text-align: left">  
49              <li>1人分だけならコレ！</li>  
50          </ul>  
51      </div>  
52      </div>  
53  </div>  
54  
55  </body>  
56  </html>
```

## seramic.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <script src=".js/nylon.js"></script>
6      <script src=".js/page.js"></script>
7      <title>コーヒーレシピブック</title>
8      <link rel="stylesheet" href=".css/seramic.css">
9      <script src=".js/seramic.js"></script>
10     </head>
11
12     <body>
13         <header>
14             <div class="item12">
15                 <button onclick="location.href='./pourover.html'">
16                     器具選択へ</button>
17             </div>
18             <div class="item12">
19                 <button id="prev_button" class="btn-square">前のステップ</button>
20             </div>
21             <div class="item12">
22                 <button id="next_button" class="btn-square">次のステップ</button>
23             </div>
24         </header>
25
26         <main>
27             <!-- 表紙 -->
28             <div id="classes" class="page">
29                 <div class="container">
30                     <div class="item13">
```

```
31          <h1>STEP1 必要な道具を用意しよう</h1>
32      </div>
33  </div>
34  <div class="container">
35      <article class="item">
36          
37          <h3>グラスサーバー</h3>
38          <a href="https://www.amazon.co.jp/s?k=グラスサーバー
&__mk_ja_JP=カタカナ&ref=nb_sb_ss_i_4_8"
39              class="btn-square">この器具を Amazon で探す</a>
40          <ul style="text-align: left">
41              <li>300ml 以上</li>
42          </ul>
43      </article>
44      <article class="item">
45          
46          <h3>ドリッパー</h3>
47          <a href="https://www.amazon.co.jp/s?k=セラミックドリ
ツバー&__mk_ja_JP=カタカナ&ref=nb_sb_noss_1"
48              class="btn-square">この器具を Amazon で探す</a>
49          <ul style="text-align: left">
50              <li>セラミック製</li>
51          </ul>
52      </article>
53      <article class="item">
54          
55          <h3>コーヒー豆</h3>
56          <a href="https://www.amazon.co.jp/s?k=コ－ヒ－
豆&__mk_ja_JP=カタカナ&ref=nb_sb_noss_1"
57              class="btn-square">この器具を Amazon で探す</a>
58          <ul style="text-align: left">
59              <li>中挽き : 20g</li>
60          </ul>
61      </article>
62      <article class="item">
```

```
63          
64          <h3>ケトル</h3>
65          <a href="https://www.amazon.co.jp/s?k=ケトル&__mk_ja_JP=
   カタカナ&ref=nb_sb_noss_1k"
66              class="btn-square">この器具の Amazon で探す</a>
67          <ul style="text-align: left">
68              <li>細口の物を</li>
69          </ul>
70      </article>
71  </div>
72 </div>
73
74 </div>
75
76 <!-- 2 ページめ -->
77 <div id="new2" class="page">
78     <div class="container">
79         <div class="item13">
80             <h1>STEP2 抽出の準備をしよう</h1>
81         </div>
82     </div>
83     <div class="container">
84         <div class="item15">
85             
86         </div>
87         <div class="item16">
88             <h2>【準備】</h2>
89             <h3>・抽出を行う準備をする。<br>
90             </h3>
91
92             <h2>【やること】</h2>
93             <h3>
94                 ・グラスサーバー・セラミックドリッパーを湯煎する。<br>
95                 ・グラスサーバーの上にセラミックドリッパーを置く。<br>
96
```

```

97          · フィルターを被せる. <br>
98          · 中挽きのコーヒー豆 20g をのせる. <br>
99          · お湯をケトルに注ぐ. <br>
100         </h3>
101     </div>
102   </div>
103 </div>
104
105 <!-- 3 ページめ -->
106 <div id="new3" class="page">
107   <div class="container">
108     <div class="item13">
109       <h1>STEP3 抽出 1/3</h1>
110     </div>
111   </div>
112   <div class="container">
113     <div class="item15">
114       
115     </div>
116     <div class="item16">
117       <h2>【蒸らし（1投目）】</h2>
118       <h3>· コーヒーの味を引き出すため、蒸らしを行う。<br>
119     </h3>
120
121       <h2>【やること】</h2>
122       <h3>
123         · 豆全体が浸るように、お湯を注ぐ。<br>
124         · 全体が浸ったら注ぐの止める。<br>
125         · 20~30 秒待つ。<br>
126       </h3>
127     </div>
128   </div>
129 </div>
130
131 <!-- 4 ページめ -->

```

```
132 <div id="new4" class="page">
133   <div class="container">
134     <div class="item13">
135       <h1>STEP4 抽出 2/3</h1>
136     </div>
137   </div>
138   <div class="container">
139     <div class="item15">
140       
141     </div>
142     <div class="item16">
143       <h2>【味を引き出す（2投目）】</h2>
144       <h3>・コーヒーの味を抽出する。<br>
145       </h3>
146
147       <h2>【やること】</h2>
148       <h3>
149         ・500円玉の外側を添うように、お湯を回しかけ続ける。<br>
150         ・出来高の7割まで続ける。<br>
151       </h3>
152     </div>
153   </div>
154 </div>
155
156 <!-- 5ページめ -->
157 <div id="new5" class="page">
158   <div class="container">
159     <div class="item13">
160       <h1>STEP5 抽出 3/3</h1>
161     </div>
162   </div>
163   <div class="container">
164     <div class="item15">
165       
166     </div>
```

```
167 <div class="item16">
168     <h2>【味を整える（3投目）】</h2>
169     <h3>・コーヒーの味を整える。<br>
170     </h3>
171
172     <h2>【やること】</h2>
173     <h3>
174         · 中心にお湯を注ぎ続け、水位を維持する。<br>
175         · 抽出高の直前で注ぐのを止める。<br>
176     </h3>
177     </div>
178 </div>
179 </div>
180
181
182     <!-- 6ページめ -->
183 <div id="new6" class="page">
184     <div class="container">
185         <div class="item13">
186             <h1>STEP6 完成</h1>
187         </div>
188     </div>
189     <div class="container">
190         <div class="item15">
191             
192         </div>
193         <div class="item16">
194             <h2>【仕上げ】</h2>
195             <h3>・抽出を終了し、仕上げをする。<br>
196             </h3>
197
198             <h2>【やること】</h2>
199             <h3>
200                 · 抽出高まで達したところで、サーバーを取り外す。<br>
201                 · 味を均一にするため、サーバーを回し攪拌する。<br>
```

```
202          </h3>
203      </div>
204      </div>
205  </div>
206
207  </main>
208  <div id="area2"><span class="final" id="final_span"></span>
209      <span class="interim" id="interim_span"></span></div>
210  <input id="btn2" type="button" value="音声認識を継続的に行
う" />
211  <select id="select2">
212      <option value="ja-JP">日本語</option>
213      <option value="en-US">English</option>
214  </select>
215
216  </div>
217  </div>
218
219  </body>
220
221  </html>
```

## index.css

```
1  body {  
2      font-family: "Avenir Next";  
3      background-image: url("../image/84595.png");  
4  }  
5  
6  .btn-square{  
7      background-color: #c5956b;  
8      border: 1px dashed #FFF;  
9      box-shadow: 0px 0px 0px 5px #c5956b;  
10     font-size: 2.0em;  
11  
12 }  
13 /* ここから下がボタンの CSS */  
14 .btn-animation-02 {  
15     display: inline-block;  
16     width: 200px;  
17     text-align: center;  
18     background-color: #c5956b;  
19     border: 2px solid #c5956b;  
20     font-size: 16px;  
21     color: #c5956b;  
22     text-decoration: none;  
23     font-weight: bold;  
24     padding: 10px 24px;  
25     border-radius: 4px;  
26     position: relative;  
27 }  
28  
29 .btn-animation-02 span {  
30     position: relative;  
31     z-index: 1;  
32 }
```

```
33
34 .btn-animation-02::before,
35 .btn-animation-02::after {
36   content: "";
37   display: block;
38   background-color: #FFF;
39   width: 50%;
40   height: 100%;
41   position: absolute;
42   top: 0;
43   transition: .2s;
44 }
45 .btn-animation-02::before {
46   left: 0;
47 }
48 .btn-animation-02::after {
49   right: 0;
50 }
51
52 .btn-animation-02:hover::before,
53 .btn-animation-02:hover::after {
54   width: 0;
55   background-color: #FFF;
56 }
57
58 .btn-animation-02:hover {
59   color: #FFF;
60 }
61
62
63
64
65
66 .container {
67   display: flex;
```

```
68     margin-top: 30px;
69     justify-content: center;
70     flex-wrap:wrap
71     background-color: rgba(255,0,0,0);
72 }
73
74 .item {
75     color:#313131;
76     padding-top: 10px;
77     padding-bottom: 10px;
78     margin-top: 150px;
79     margin-bottom: 10px;
80     padding-left: 20px;
81     padding-right: 20px;
82     border-radius: 5px;
83     letter-spacing: 3px;
84     font-weight:400;
85     float: left;
86     background-color: rgba(255,0,0,0);
87 }
88 .item h1 {
89     position: relative;
90     background: #c5956b;
91     box-shadow: 0px 0px 0px 5px #c5956b;
92     border: dashed 2px white;
93     padding: 0.2em 0.5em;
94     color: #454545;
95     font-size: 100px;
96 }
97
98 .item h1:after {
99     position: absolute;
100    content: '';
101   left: -7px;
102  top: -7px;
```

```
103 border-width: 0 0 15px 15px;  
104 border-style: solid;  
105 border-color: #fff #fff #a8d4ff;  
106 box-shadow: 1px 1px 1px rgba(0, 0, 0, 0.15);  
107 }
```

## seramic.css

```
1  body {
2      font-family: "Avenir Next";
3      background-image: url("../image/84595.png");
4  }
5
6  li {
7      list-style: none;
8  }
9
10 .main {
11     padding: 10px 30px;
12 }
13
14 .btn-square {
15     display: inline-block;
16     padding: 0.5em 1em;
17     text-decoration: none;
18     background: #668ad8; /*ボタン色*/
19     color: #FFF;
20     border-bottom: solid 4px #627295;
21     border-radius: 3px;
22 }
23 .btn-square:active {
24     /*ボタンを押したとき*/
25     -webkit-transform: translateY(4px);
26     transform: translateY(4px); /*下に動く*/
27     border-bottom: none; /*線を消す*/
28 }
29 .btn-square2 {
30     display: inline-block;
31     padding: 0.5em 1em;
32     text-decoration: none;
```

```
33     background: #ff6347; /*ボタン色*/
34     color: #FFF;
35     border-bottom: solid 4px #ff4500;
36     border-radius: 3px;
37 }
38 .btn-square2:active {
39     /*ボタンを押したとき*/
40     -webkit-transform: translateY(4px);
41     transform: translateY(4px); /*下に動く*/
42     border-bottom: none; /*線を消す*/
43 }
44
45 /* input と textarea の CSS を指定してください */
46 input, textarea{
47     width:400px;
48     margin-top:10px;
49     margin-bottom:30px;
50     padding:20px;
51     font-size:18px;
52     border:1px solid #dee7ec;
53 }
54
55 /* contact-submit の CSS を指定してください */
56 .contact-submit{
57     background-color:#dee7ec;
58     color:#889eab
59 }
60
61 .midashi h1{
62     text-align: center;
63     font-size: 25px;
64     margin: 0px;
65     color:#313131;
66 }
67
```

```
68 .container {  
69     display: flex;  
70     background-color: #00000000;  
71     margin-top: 0px;  
72     justify-content: center;  
73     flex-wrap: wrap;  
74 }  
75 .container2 {  
76     display: flex;  
77     background-color: #00000000;  
78     height: 550px;  
79     margin-top: 40px;  
80     justify-content: center;  
81     flex-wrap:wrap  
82 }  
83  
84 header {  
85     display: flex;  
86     background-color: #00000000;  
87     height: 70px;  
88     margin-bottom: 15px;  
89     justify-content: center;  
90     flex-wrap:wrap  
91 }  
92  
93 .item1{  
94     margin: 50px;  
95     flex-basis: 500px;  
96 }  
97 .item2{  
98     margin: 50px;  
99     background-color: #ffffff;  
100    border: 3px solid #bbbbbb;  
101 }  
102 .item4{
```

```
103     margin: 40px;
104     background-color: #ffffff;
105     border: 3px solid #bbbbbb;
106     text-align:center
107 }
108 .item5{
109     margin: 40px;
110     background-color: #ffffff;
111     border: 3px solid #bbbbbb;
112     text-align:center
113 }
114 .item6{
115     margin: 40px;
116     background-color: #ffffff;
117     border: 3px solid #bbbbbb;
118     text-align:center
119 }
120 .item7{
121     margin: 40px;
122     background-color: #ffffff;
123     border: 3px solid #bbbbbb;
124     text-align:center
125 }
126 .item{
127     width: 250px;
128     height: 400px;
129     margin: 20px;
130     background-color: #ffffff;
131     border: 3px solid #bbbbbb;
132     text-align:center
133 }
134 .item9{
135     margin: 20px;
136     background-color: #ffffff;
137     border: 3px solid #bbbbbb;
```

```
138     text-align:center
139 }
140 .item10{
141     margin: 20px;
142     background-color: #ffffff;
143     border: 3px solid #bbbbbb;
144     text-align:center
145 }
146 .item11{
147     margin: 20px;
148     background-color: #ffffff;
149     border: 3px solid #bbbbbb;
150     text-align:center
151 }
152 .item12{
153     margin: 20px;
154     background-color: #00000000;
155 }
156 .item13{
157     margin: 0px;
158     width:500px;
159     text-align:center;
160     flex-basis: 400px;
161     flex-wrap: nowrap;
162 }
163
164 .item13 h1 {
165     font-size:20px;
166     font-weight:400;
167     padding: 0.4em 0.5em; /*文字の上下 左右の余白*/
168     color: #494949; /*文字色*/
169     background: #f4f4f4; /*背景色*/
170     border-left: solid 5px #7db4e6; /*左線*/
171     border-bottom: solid 3px #d7d7d7; /*下線*/
172 }
```

```
173 .item14{
174     margin: 20px;
175     background-color: #00000000;
176 }
177 .item15{
178     margin: 20px;
179     background-color: #cccccc;
180     border: 3px solid #999999;
181     text-align:center;
182     flex-basis: 500px;
183 }
184
185 .item16{
186     margin: 25px;
187     background-color: #ffffff;
188     border: 3px solid #999999;
189     text-align:left;
190     flex-basis: 500px;
191 }
192 #area2 {
193     width: 90%;
194     height: 2em;
195     padding: 10px;
196     border: 1px solid #ccc;
197     border-radius: 3px;
198     margin-bottom: 10px;
199     margin-top: 20px;
200     background-color:#ffffff;
201 }
202 #btn2 {
203     background: -webkit-linear-gradient(top,#008dfd 30%,#0370ea 100%);
204     color: white;
205     text-shadow: 1px 1px 1px #076bd2;
206     border-radius: 3px;
207     border: 1px solid #076bd2;
```

```
208     padding: 8px 25px;  
209     font-weight: 700;  
210     font-size: 15px;  
211 }  
212 #btn2.select {  
213     background: -webkit-linear-gradient(top, #f2739e 30%, #f23f7b 100%);  
214     text-shadow: 1px 1px 1px #f1115c;  
215     border: 1px solid #f1115c;  
216 }  
217 #select2 {  
218     width: 200px;  
219 }
```

## page.js

```
1  class Page {
2      constructor( pages ) {
3          this.pages = pages;
4          console.log(" this.pages " + this.pages );
5          this.now = pages[0];
6          console.log(" this.now " + this.now );
7          this.page_size = pages.length;
8          console.log(" this.page_size " + this.page_size );
9      }
10     change( page ) {
11         let pages1 = document.querySelectorAll('.page');
12         console.log(" pages1 " + pages1 );
13         for( let p of pages1 ) {
14             p.style.display = "none";
15         }
16         document.querySelector( page ).style.display = "block";
17         this.now = page;
18         console.log(" this.now " + this.now );
19     }
20     next() {
21         let number = this.pages.indexOf( this.now );
22         console.log(" number " + number );
23         if( number < this.page_size-1 ) {
24             let a = document.querySelector( this.now );
25             console.log(" this.pages[number+1] " + this.pages[number+1] );
26             this.change( this.pages[number+1] );
27         }
28     }
29     prev() {
30         let number = this.pages.indexOf( this.now );
31         if( 0 < number ) {
32             let a = document.querySelector( this.now );
```

```
33         this.change( this.pages[number-1]);  
34     }  
35 }  
36 }
```

# seramic.js

```
1 window.addEventListener('load', () => {
2     page = new Page(
3         [ "#classes", "#new2", "#new3", "#new4", "#new5", "#new6"]);
4     page.change('#classes');
5
6     let a = document.querySelector('#next_button');
7     console.log(a);
8
9     document.querySelector('#next_button').addEventListener('click', () => { page.next(); });
10    document.querySelector('#prev_button').addEventListener('click', () => { page.prev(); });
11
12    function () {
13        var recognition;
14        var nowRecognition = false;
15        var $finalSpan = document.querySelector('#final_span');
16        var $interimSpan = document.querySelector('#interim_span');
17
18        function start() {
19            recognition = new webkitSpeechRecognition();
20            recognition.lang = document.querySelector('#select2').value;
21            recognition.continuous = true;
22            recognition.interimResults = false;
23            //音声認識をすると onresult が動く
24            //e が音声認識したデータが入っている
25            recognition.onresult = function (e) {
26                var finalText = '';
27                var interimText = '';
28                //e.results が音声認識の結果（全体）
29                //#[e.results.length-1] が最新のデータ
30                //#[0].transcript が Ø を日本語にしたもの
31                //#[0] の中の transcript に認識したワードが入ってい
```

```
32         interimText =
33             e.results[e.results.length-1][0].transcript;
34         $interimSpan.textContent = interimText;
35         //console.log(" seramic.js:33 " + interimText );
36         if (interimText === "次へ") page.next();
37         else if (interimText === "戻る") page.prev();
38         $finalSpan.textContent = finalText;
39     };
40     recognition.onstart = function () {
41         console.debug('[onstart]');
42     };
43     recognition.onaudiostart = function () {
44         console.debug('[onaudiostart]');
45     }
46     recognition.onsoundstart = function () {
47         console.debug('[onsoundstart]');
48     }
49     recognition.onspeechstart = function () {
50         console.debug('[onspeechstart]');
51     }
52     recognition.onspeechend = function () {
53         console.debug('[onspeechend]');
54     }
55     recognition.onsoundend = function () {
56         console.debug('[onsoundend]');
57     }
58     recognition.onaudioend = function () {
59         console.debug('[audioend]');
60     }
61     recognition.onnomatch = function () {
62         console.debug('[onnomatch]');
63     }
64     recognition.onerror = function () {
65         console.debug('[onerror]');
66     }
```

```
67         recognition.onstart = function () {
68             console.debug('[onstart]');
69         }
70         recognition.onend = function () {
71             console.debug('[onend]');
72         }
73         recognition.start();
74         nowRecognition = true;
75     };
76     function stop() {
77         recognition.stop();
78         nowRecognition = false;
79     }
80
81
82     document.querySelector('#btn2').onclick = function () {
83
84         // unsupported.
85         if (!'webkitSpeechRecognition' in window) {
86             alert('Web Speech API には未対応です。');
87             return;
88         }
89
90         if (nowRecognition) {
91             stop();
92             this.value = '音声認識を継続的に行う';
93             this.className = '';
94         } else {
95             start();
96             console.log("seramic.js:94 start");
97             this.value = '音声認識を止める';
98             this.className = 'select';
99         }
100    }
101 }
```

102 } );