

2019年度 卒業論文

論理的文章のアウトラインの作成を  
支援するツールの開発

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

1632144 三浦 恋

提出日 2020年1月30日

# 目 次

<b>1 緒言</b>	<b>1</b>
<b>2 論文について</b>	<b>2</b>
2.1 論文とは . . . . .	2
2.2 論文の書き方 . . . . .	2
2.2.1 テーマを決める . . . . .	2
2.2.2 下調べを行う . . . . .	3
2.2.3 アウトラインを作成する . . . . .	4
2.2.4 執筆する . . . . .	5
<b>3 アカデミックライティングについて</b>	<b>6</b>
3.1 アカデミックライティングとは . . . . .	6
3.2 アカデミックライティングの特徴 . . . . .	6
3.3 なぜアカデミックライティングが必要がなのか . . . . .	7
3.4 作文・感想文との違い . . . . .	7
<b>4 論文などの作成をするソフトウェア紹介</b>	<b>8</b>
4.1 アウトラインプロセッサ . . . . .	8
4.2 TeX . . . . .	9
4.3 Microsoft Word . . . . .	11
<b>5 マインドマップ</b>	<b>12</b>
5.1 マインドマップについて . . . . .	12
<b>6 開発言語と関連技術</b>	<b>13</b>
6.1 HTML . . . . .	13
6.1.1 HTMLについて . . . . .	13
6.1.2 HTMLの特徴 . . . . .	13

6.1.3	HTML5について . . . . .	14
6.2	CSS . . . . .	15
6.2.1	CSSについて . . . . .	15
6.2.2	CSS3について . . . . .	15
6.2.3	メディアクエリについて . . . . .	15
6.3	JavaScript . . . . .	16
6.3.1	JavaScriptとは . . . . .	16
6.3.2	ネイティブアプリについて . . . . .	17
6.3.3	Web アプリについて . . . . .	17
6.4	PWAについて . . . . .	18
6.5	ローカルストレージについて . . . . .	19
6.5.1	ローカルストレージとは . . . . .	19
6.5.2	ローカルストレージ利用例 . . . . .	19
<b>7</b>	<b>本研究で開発したツールの概要</b>	<b>20</b>
7.1	実装理由 . . . . .	20
7.2	実装した機能について . . . . .	20
7.2.1	主張と根拠の明確化 . . . . .	21
7.2.2	課題に対する疑問とその答えの記入 . . . . .	21
7.2.3	論理的な構成の整理 . . . . .	21
7.2.4	参考文献の管理 . . . . .	21
7.3	本ツールの利用方法 . . . . .	24
7.3.1	ホーム画面 . . . . .	24
7.3.2	課題名の記入欄 . . . . .	25
7.3.3	主張の記入欄 . . . . .	25
7.3.4	根拠の記入欄 . . . . .	26
7.3.5	問い合わせの記入欄 . . . . .	27
7.3.6	参考文献の管理 . . . . .	28

7.4 画面構成について . . . . .	29
<b>8 結言</b>	<b>30</b>
謝辞	31
参考文献	31
<b>付録</b>	<b>33</b>

## 図 目 次

1	アウトラインの作成例	4
2	Dynalist のアウトラインの例	8
3	T <sub>E</sub> X フォーマットによる文章の例	10
4	Microsoft Word	11
5	Simple Mind lite の使用例	12
6	PWA の例	18
7	localStorage の表示例	20
8	アカデミックアウトライントール	23
9	起動時の画面	24
10	課題名の記入欄	25
11	主張の記入欄	25
12	根拠の表示前	26
13	根拠の表示後	26
14	問い合わせの表示前	27
15	問い合わせの表示後	27
16	参考文献の表示前	28
17	参考文献の表示後	28
18	PC での画面表示	29
19	スマートフォンでの画面表示と iPad での画面表示	29

# 1 緒言

大学では論文やレポート課題などを書く機会があり、大学生に対して、論理的な思考力や論理的文章作成能力の要求が高まっている。しかし、論文やレポートを書く際にアウトラインなどの事前準備をせずに文章の作成を行ってしまう学生が多く、論理的な文章の特徴の1つである、主張や根拠が明確であり一貫した文章にならないことが問題点として挙げられる。そのため、レポートの書き方の指導や修正を行うライティングセンターの設置などが進められているが継続的な利用が必要とされている。

一般に論文や小説などの長文の作成を支援するためのツールとして、アウトラインプロセッサが使用されることが多い。これは、文章を階層的に管理することに主眼が置かれており、学生にとって主張や根拠などが明確な一貫した文章を書く力を養うためのツールではないことが問題点として挙げられる。

そこで本制作では、主張や根拠などが明確な一貫した論文やレポートを書くため、文章作成の準備段階であるアウトラインの作成を支援する、アカデミックアウトラインツールを開発することを目的としている。

## 2 論文について

### 2.1 論文とは

論文とはエッセイや小説のように自由な文章表現ではなく、一定の形式に備えた文章表現である。またテーマをもとに問題を立て、問題に対し様々な手法で分析、考察し、問題解決につながる新たな知見や検証を行い、その結果を報告するものが論文である [1].

### 2.2 論文の書き方

論文を書く流れとして主に4つ作業工程を繰り返し行うことで、論文を書くことができる。また、4つの工程を以下に示す。

- テーマを決める
- 下調べを行う
- アウトラインの作成する
- 執筆する

#### 2.2.1 テーマを決める

どのようなテーマの論文を書くのかを決めるため、素朴な疑問や資料を読んだ際の疑問を大切にし、論文のテーマを決めていく。またテーマが既に決まっている場合はキーワードをもとに、図や表などを使って思考を整理し、論点を見いだし下調べに入る。

### 2.2.2 下調べを行う

論文のテーマが決まった場合テーマに関する知識を得るために、似たテーマの論文を調べ知見を広げる必要がある。文献等を調べる際は図書館での検索やデータベースによる検索などを行うなどをして、資料を収集する。また収集した資料を読み込み、疑問点などが出てきた際には2.2.1に戻りテーマについて思考の整理などを行う。論点が定まり、十分な情報が集まるまでテーマ決めと下調べを繰り返し行う。

### 2.2.3 アウトラインを作成する

下調べが終わり、論文のテーマが決定した場合、次に、作成する文章の骨組みである、アウトラインの作成を行う。紙に書き出すことやアウトラインプロセッサなどのソフトウェアを用いて、章や節の書く内容を箇条書きに近いかたちで書いていき、全体の文章構造を決めていく。またアウトラインの作成例を図1に示す。

#### 第1章 初めに

- 第1節 根拠（大学生には論文を書く力は必要）
  - 第1項 2022年の論理国語
  - 第2項 ライティングセンターの設置
  - 第3項 初年次教育
- 第2節 問題点
  - 第1項 ライティングセンターは自主的な利用者のみが効果がある  
主張一貫していない  
一文が長い  
言葉使いの統一
  - 第2項 初年次教育の説明では不十分  
初年次教育はレポートの書き方を教えることが目的の授業ではないため
  - 第3項 アウトラインの作成をしていない
- 第3節 目的
  - 第1項 アウトラインプロセッサが存在する
  - 第2項 アウトラインの作成をしていない
  - 第3項 アカデミックアウトライントールの開発をする

#### 第2章 アカデミックライティングについて

- 第1節 特徴など

#### 第3章 ツールの概要

- 第1項 AOTの機能について
  - どんなものが必要であるか（根拠）
  - 一般的なアウトライントールの紹介（比較）
  - 機能について（事実）
    - 主張の表示
    - 根拠の表示
    - バラグラフ構造（問と答え）
    - ラベル文章のラベル（番号の入力）
    - 参考文献の表示（まとめ）

#### 第4章 おわりに

図 1: アウトラインの作成例

#### 2.2.4 執筆する

フォーマットやアウトラインをもとに、執筆を行う。アウトラインや整理した資料、行った実験や検証の結果をもとにアウトラインを更に細かく作成していく、アウトラインから文章を作成をしていく。そこで必要な情報があった際には調べ、アウトラインを修正し、文章作成を行う。また、文章の書き出しから完成まで、途中何度も書いた文章を確認、添削を行う必要がある。

### 3 アカデミックライティングについて

#### 3.1 アカデミックライティングとは

大学では答えのない問題を扱うことが多く、問題に対して自分の考え方を主張することが必要とされている。そこで、大学で作成が求められる論文やレポート等には、複数の特徴のある文章作成が求められる。このような文章を書く技術、書く行為はアカデミックライティングと呼ばれている[2]。

#### 3.2 アカデミックライティングの特徴

アカデミックライティングには重要な特徴として、以下の(1)~(5)が挙げられる。

- (1) 主張と根拠が明示されている
- (2) 問いと答えの構造と論理的な説明での構成されている
- (3) 引用の倫理のルールに従っている
- (4) パラグラフ構造になっている
- (5) 学術的文章に特有の一定の形式に従っている

またアカデミックライティングは、専門的な内容を論じたり、まだ答えが発見されていないことについて論じることがあるため、複雑な概念や専門用語を用いて文章の作成が行われる。しかし内容が読者に伝わらなければ文章の意味がなく、そのためにアカデミックライティングはわかりやすい文章で書く必要があることも特徴として挙げられる[3]。

### 3.3 なぜアカデミックライティングが必要がなのか

大学では答えのない問題を扱うことが多く、授業ではレポート課題が出されることがある。そこで、新たな発見やすすでに分かっているが解釈が分かれたり、位置付けのはっきりしない事柄が多くある。そのため答えのない問題について、学生がどの程度授業の内容を理解し、また自分なりの問い合わせや答えを見つけることに努力を行ったかを確認するために課している。

### 3.4 作文・感想文との違い

同じ文章でもアカデミックライティングは作文や感想文とは大きく異なる。作文や感想文は自分の経験や思いを書くものであり、言葉遣いも話し言葉のような文体でも許される。しかし、アカデミックライティングは文献や調査結果などの根拠をもとに学術的なルールに従った報告書である。そのため、内容として論理的な一貫した説明でなければならぬ。

## 4 論文などの作成をするソフトウェア紹介

### 4.1 アウトラインプロセッサ

アウトラインプロセッサとは、一般的には小説などの長文を書く際に利用されている全体の構成を確認しながら文章の作成を支援するソフトウェアである。特徴として、見出しをつけ階層的に管理できる事によりブロック単位で位置の入れ替えや章から節への移動などもすることができる。また文章を書くことが目的として作成されたソフトウェアであるため動作が軽く、シンプルなものが多い。しかし長文を書くことに長けているが図や表などを挿入されることは考えられないことが多いが多くの図や表を扱う場合には向いていないことなどが特徴として挙げられる。実際のアウトラインプロセッサの1つである Dynalist の動作例を図2に示す。

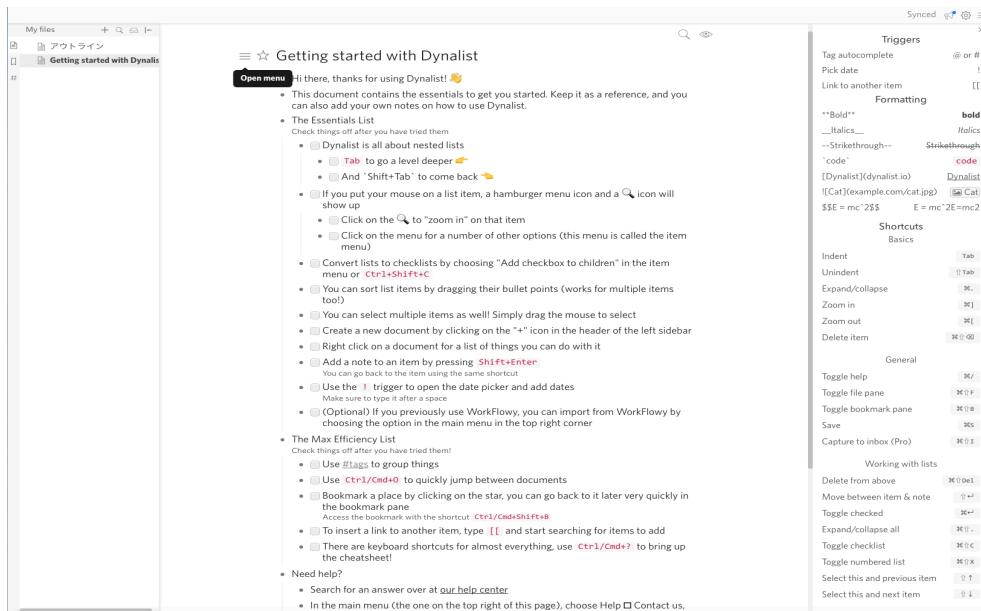


図 2: Dynalist のアウトラインの例

## 4.2 TeX

$\text{\TeX}$ とは、アメリカの著名な数学者にして計算機科学者である Donald E. Knuth が作成した組版フリーソフトウェアである。 $\text{\TeX}$ 本体は、文字を配置する、基本的な組版作業に対応する命令を処理するものであり、命令だけを用いて文書を作成するのは効率的ではない。そこで多くの場合マクロセットと呼ばれる命令セットを用いて文書を作成している。

マクロとは、組版された文書の作成を容易にするために、複数の基本的な命令を組み合わせて作成された新たな命令である。マクロセットにはさまざまなものがあるが、もっとも有名でよく用いられているものが、アメリカの計算機科学者である Leslie Lamport が作成した  $\text{\LaTeX}$  である。 $\text{\TeX}$  で文書を作成するという場合、実際はこの  $\text{\LaTeX}$  の命令を用いて作成することがほとんどである。

また、日本語で書かれた文章  $\text{\TeX}$  で組版するため、(株)アスキーにおいて ASCII 日本語  $\text{\TeX}$  が、日本電信電話公社において NTT JTeX がそれぞれ開発されたことにより、日本においても  $\text{\TeX}$  が普及し現在に至っている [4]。特徴としてフォーマットをほとんど気にせず作成することができる数式のコマンドが豊富で複雑な数式であっても綺麗に記述することができる。また参考文献や図、表などの記述や管理、文章の調整なども手作業ですることなくコマンドを利用することで、一括で行うことが可能である。そのため数学を多用する分野などで広く用いられている。しかし  $\text{\TeX}$  を使用する場合  $\text{\TeX}$  の文法を学ぶ必要があることや直感的な操作をすることができないため、慣れるまでは難しく感じてしまうことがあるなどが特徴として挙げられる。 $\text{\TeX}$  フォーマットによる文章の例を図 3 に示す。

The screenshot shows a LaTeX editor window with the file 'texitorionD4.tex' open. The code is written in Japanese and discusses academic writing tools. It includes sections on chapter 2 and 3, and a description of features like main arguments, references, and a tool developed for the project.

```

27 \% 2章
28 \section{アカデミックライティングについて}
29 \% 既存のツールで不足する点
30 大学では答えのない問題を扱い、問題に対して自分の考えを主張することが必要とされている。そこで、論文やレポート等には下記(1)～(5)が求められる。このような文章を書く技術、書く行為はアカデミックライティングと呼ばれている\cite{ren01}。
31
32 \% アカデミックライティングツールの理想
33 \begin{description}
34 \item[(1)] 主張と根拠が明示されている
35 \item[(2)] 問いと答えの構造と論理的な説明での構成されている
36 \item[(3)] 引用の倫理のルールに従っている
37 \item[(4)] パラグラフ構造になっている
38 \item[(5)] 学術的文章に特有の一定の形式に従っている
39 \end{description}
40
41 \% 3章
42 \section{開発したツールについて}
43 \% アカデミックライティングとコンセプトについてを書く
44 本制作では論理的な構成の文章を書く際の準備段階のために、主張と根拠の確認や参考文献の管理を行うことができるツールが必要だと考えた。また、通学時間などの隙間時間で、意見や構成の整理を行えることを目指した。
45
46 \% 2章から導かれた、求められる機能
47 主張が一貫した論理的文章のアウトラインを作成するために、2章で述べた(1)～(3)の考え方の整理の補助、文章構成の整理、スマートフォンでの利用を考えたツールを開発した。(4)、(5)においては文章に起こした際の特徴であるため、本制作では対象外とした。また実際に書く文章は別のアプリケーションで記述することにする。
48
49 本ツールの具体的な機能を①～④で記載し、動作例と機能①～④の対応したものを図\ref{fig:g}に示す。
50 \begin{description}
51 \item[①] 主張と根拠の明確化
52 見返した際に主張からはずれた意見が出ることを防ぐため、主張と根拠を登録する機能
53
54 \item[②] 课题に対する疑問とその答えの記入
55
56

```

図 3: TeX フォーマットによる文章の例

## 4.3 Microsoft Word

Word とは Microsoft 社が提供する文章作成ソフトウェアである。Microsoft Office の中でも、主要なソフトウェアの 1 つに挙げられる。また、文章作成だけでなく、図形描画やグラフ、アウトラインの作成など、豊富で様々な機能のを持つ。特徴として直感的な操作が可能であり、画面に表示された通りに印刷されることや複雑な図や表を作成することが可能である。またテキストの検索をすることができるため、修正や置換を行えるや同じ Microsoft 社の Excel や PowerPoint で作成したデータを取り扱うことができる。しかし図や表などを挿入する位置の調整やページの調整を手作業で行わなければならないなどが特徴として挙げられる。実際の動作例を図 4 に示す。

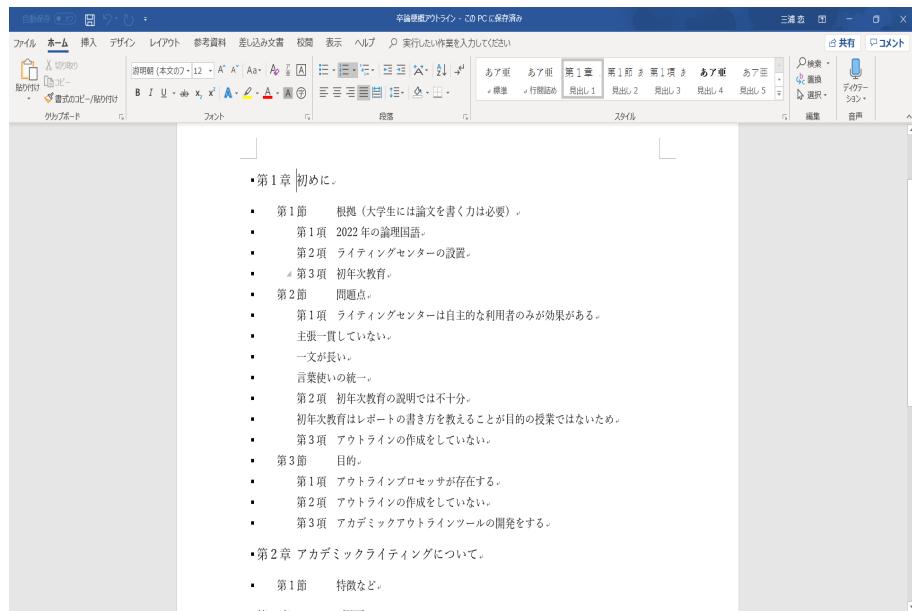


図 4: Microsoft Word

## 5 マインドマップ

### 5.1 マインドマップについて

マインドマップとは、頭の中で自然に行っている思考のプロセスを反映したノート法である。イギリス人教育者であるトニー・ブザン (Tony Buzan) が1970年代に出演していたTV番組を始めとして様々な著作で「マインドマップ」という言葉が広め始めた[5]。また、自由な思考、アイデアや情報の流れを中心となる概念から分岐させる形で描画した図である。描画することで、アイデアの整理、効果的なメモの作成、記憶の定着強化などを実現することが可能になる。実際のマインドマップの例として Simple Mind lite の動作例を図5に示す。

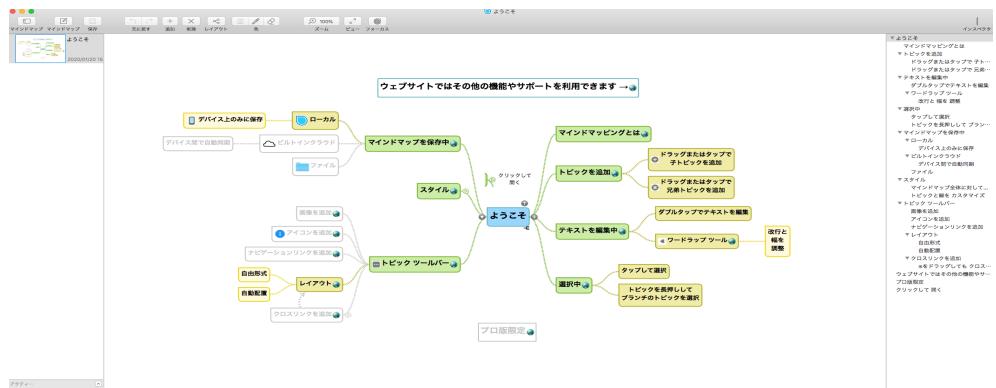


図 5: Simple Mind lite の使用例

## 6 開発言語と関連技術

開発言語として、自宅や学校のみならず隙間時間に利用することを視野に入れたため、PC とスマートフォンの両方からのアクセスによる利用を考えた。そこで Web 上で動作するツールが望ましいと考え、HTML5,CSS3, JavaScript を使用し開発を行った。

### 6.1 HTML

#### 6.1.1 HTMLについて

HTML とは、Hyper Text Markup Language の略称であり、ウェブ上のドキュメントを記述するためのマークアップ言語である。Web ページを作成する際の基本的プログラミング言語であり、C 言語の様なプログラミングとは異なり、文章中に記述することで様々な機能を設定することができる。HTML でマークアップされたドキュメントは異なるドキュメントへのハイパーリンクを設定できるハイパーテキストであり、リスト、表等の高度な表現力も持ち合わせている。また、JavaScript や CSS などを直接書き込まなくても、別のファイルから呼び出すことが可能である [8]。

#### 6.1.2 HTML の特徴

HTML の特徴として、ハイパーテキストを利用した相互間文章参照のフレームワークが挙げられる。文章の特定要素に URI(Uniform Resource Identifier) を用いた他文章へのリンクを記載することで、ユーザエージェントはそれを解釈し、指定された他文章を表示させることができる。マークアップは、プレーンテキストの文章を要素で括り意味づけすることで行うが、その際に引用する画像の埋め込みや文章タイトルの指定等を定める要素を記載することで、ユーザエージェントがそれらを解釈し見合った表示を行う [8]。

### 6.1.3 HTML5について

HTML5とは、以前標準となっていたHTML4やXHTML1.xの後継にあたる仕様である。しかし、HTML4やXHTML1.xが扱う範囲とは大きく異なり、範囲が多岐にわたることが特徴といえる。ここでは、マークアップの仕様だけでなく、周辺API(Application Programming Interface)まで含めてHTML5と定義を行い、説明する。

HTML4やXHTML1.xではマークアップの仕様が主だったが、HTML5からマークアップだけでなく、DOM (Document Object Model) の仕様やAPIの仕様が数多く盛り込まれるようになった。これまで使用してきたAPIは、標準化団体であるW3C(World Wide Web Consortium)が規定したAPIだけでなく、デファクトスタンダード化したものやブラウザベンダー独自のものが数多く使われてきた。このようなAPIはブラウザによって挙動が異なることや、仕様書がないなどの問題があったが、HTML5で改めて仕様として規定し直されている。

これまでプラグインを用いなければ実現できなかった機能を、CanvasやVideoなどHTMLの要素によって、プラグインなしで実現することができる。また、File APIやWeb Workerの様なプラグインがあっても実現できなかった機能が開発された。これらのことから、HTML5はWebアプリケーションに必要とされる多くの要素を集結させた仕様であるといえる。

しかし、HTML4などで使用していた要素、属性などがいくつか廃止され、他の要素、属性を用いて使用する形になるなど、新しいものが追加されただけでなく、以前からあったものも多く変更がされているため注意が必要である[8]。

## 6.2 CSS

### 6.2.1 CSSについて

CSS とは Cascading Style Sheet の略称であり, HTMLなどのウェブページのスタイルを指定するための言語である。ウェブページを装飾する際のスタイルシート言語の1つであり, CSS が一般的に利用されている。ウェブページに書かれた各要素の装飾を指定することができ, スクリーンに表示される色やサイズ, レイアウトなどを CSS で指定することが可能である。

### 6.2.2 CSS3について

CSS3 とは単一の規格ではなく, 「CSS Color Module Level3」など機能単位で策定される方針に変更され, それらを総称して CSS3 と呼ばれている。

### 6.2.3 メディアクエリについて

スマートフォンやタブレットの登場により, PC以外の端末で Web サイトを見る機会が増えているため, 端末に合わせて表示する必要がある。そこで, 端末の種類や画面の解像度などに応じてサイトやアプリの表示を変更することが可能な, メディアクエリが利用されている [10]。

利用方法を以下に示す。また HTML 側のビューポートの設定が以下のように記述されていることを前提とする。

ソースコード 1: ほげほげ

---

```
1 <meta name="viewport" content="width=device-width">
```

---

HTML ファイルでふあいるを読み込みメディアクエリを利用する方法は, デバイスによって読み込むファイル自体を振り分け, HTML

側で以下のように記述する. 例として今回は横幅が 480px までのデバイスのみ, index.css というファイルを読み込む

---

ソースコード 2: ファイルを読み込む

---

```
1 <link rel="stylesheet" type="text/css" media="(max-width: 480px)"  
      href="index.css">
```

---

次に, CSS 内で記述を行う.CSS のファイル内の先頭に以下記述を行い {} 内にスタイルの記述を行う.

---

ソースコード 3: CSS の記述例

---

```
1 @media print, screen and (max-width: 480px){  
2   css の記述  
3 }
```

---

## 6.3 JavaScript

### 6.3.1 JavaScript とは

Netscape Communications 社が開発した, プロトタイプベースのオブジェクト指向スクリプト言語である. Web ブラウザにおいて, 従来は印刷物のような静的な表現しかできなかった Web ページに動きや対話性を付加することを目的に開発され, 主要な Web ブラウザに搭載された. しかし, 各社の実装には微妙な違いがあり, Web ブラウザによって使えない機能や, 同じプログラムでも挙動が異なるなどの問題があった. そのため, 国際標準化団体である Ecma インタナショナルによって中核的な仕様が ECMAScript として標準化された.

Web ブラウザと統合している JavaScript の処理系では DOM 操作が規定されており, 動的な HTML の構築を可能としている. この機能は Ajax の中核技術としても使われており, 現在の Web アプリケーション開発に不可欠なものとなっている.

以前はインタプリタ方式で実行されることが一般的であったため, 実行速度はさほど速くなかった. 現在では JIT コンパイルなどを利用

した各種の最適化がなされており、各 Web ブラウザベンダーともに高速化を図ってしのぎを削っている。処理系の高速化や WebWorkers や WebGL などの新しい API の追加により、Web ブラウザ上で高度な計算や 3DCG の描画など高度な処理が可能となっている [10]。

### 6.3.2 ネイティブアプリについて

ネイティブアプリとはプラットホームにおいて、直接動作するアプリケーションのことである。ネイティブアプリには OS によって様々であり、主に Windows アプリ、Mac アプリ、Android アプリ、iPhone(iOS) アプリの 4 種類が存在する。またメリットとして Web アプリケーションよりも動作が早く、ネットワーク環境が必要なくとも動作する点が挙げられる。デメリットとしてアプリケーションのインストールが必要であることが挙げられる。

### 6.3.3 Web アプリについて

Web アプリとは Google Chrome や Firefox などの Web ブラウザ上で動作するアプリケーションのことである。メリットとして Web アプリはインストールの必要がなく、インターネット上で管理を行っているため、別の端末でアクセスしても動作が可能である点が挙げられる。デメリットとして、インターネットに接続されていなければ利用ができないことやネイティブアプリと比べると細かな操作をすることができない点が挙げられる。

## 6.4 PWAについて

PWA(Progressive Web Apps)とはWebサイトなどをスマートフォン向けアプリケーションのように動作を行うことが可能な技術である。具体的にはインストールが必要なく、ホーム画面へのアイコン追加やプッシュ通知などが可能であり、ユーザーとの接触機会を増やすことができる。また読み込み速度や表示の高速化、オフラインでの閲覧も可能であるなど様々なメリットが得られる。またアプリとの違いとして、アプリストアを経由してダウンロードやインストールする手間がなく、アプリの導入までの手順を短縮ができる。またプラットフォームごとに開発する必要もなく1つのPWAを構築するだけで、デバイスを問わずに一貫した内容を表示できるなど開発の自由度が高いことが特徴として言える[7]。PWAを実装した本ツールのホーム画面を図6の左に、起動後の画面を右に示す。



図 6: PWA の例

## 6.5 ローカルストレージについて

### 6.5.1 ローカルストレージとは

ローカルストレージ(localStorage) とは Web ブラウザにデータを保存する仕組みであり、 HTML5 では WebStorage というローカル環境である Web ブラウザ側でデータを保存することが可能な仕組みの中の 1 つである。ローカルストレージは 1 つのオリジンに対して、数 MB (2~10MB と各ブラウザによって異なる) までデータを保存でき、ページを閉じても保存したデータは保持される。

### 6.5.2 ローカルストレージ利用例

ローカルストレージの利用方法として JavaScript を用いた記述例を以下に示す。保存するデータは key と value の組み合わせによって保存される。また、1 つの key に対して複数保存を行う場合は、JSON 形式の文字列に変換する必要があるため、`JSON.stringify()` メソッドを用いて記述を行う。Web 上での表示例を図 7 に示す。

ソースコード 4: 利用例

---

```
1 //localStorage にデータを保存する
2 let data =localStorage.setItem('キー', 'ああああ');
3
4 //複数保存する例
5 var Jdata ={
6     data1: document.querySelector('#keysave').value,
7     data2: document.querySelector('#save').value
8 };
9 localStorage.setItem("deku" , JSON.stringify(Jdata));
10
11 //確認用
12 console.log(data);
13 console.log(Jdata);
14
15 localStorage からデータを削除する
16 //localStorage.removeItem("キー");
```

---

The screenshot shows the Chrome DevTools Application tab interface. On the left, there's a sidebar with sections for Application (Manifest, Service Workers, Clear storage), Storage (Local Storage, Session Storage, IndexedDB, Web SQL, Cookies), Cache (Cache Storage, Application Cache), and Background Services (Background Fetch, Background Sync). The Local Storage section is expanded, showing a list for the domain 'http://127.0.0.1:5501'. One item is selected: 'キー' (Key) 'deku' and 'Value' 'ああああ' (aaaa). A preview pane on the right shows the JSON object: {"data1": "キー", "data2": "保存するデータ"}. At the bottom, it says 'Line 1, Column 1'.

図 7: localStorage の表示例

## 7 本研究で開発したツールの概要

### 7.1 実装理由

本制作では論理的文章を書く際の準備段階である、アウトラインの作成において、主張と根拠の確認や参考文献の管理を行うことで、アウトラインの作成や主張の一貫した文章の作成を支援することができるツールが必要であると考えた。また、通学時間などの隙間時間で意見や構成の整理を行うことで、アウトラインの作成時間を短くし、論理的文章を書く時間の確保ができることを目指した。

### 7.2 実装した機能について

本制作で開発したツールの機能は以下の4つである。ツール画面と機能①～④の対応したものを図8に示す。

- ① 主張と根拠の明確化
- ② 課題に対する疑問とその答えの記入
- ③ 論理的な構成の整理
- ④ 参考文献の管理

#### 7.2.1 主張と根拠の明確化

主張と根拠の明確化の機能は、アウトライン作成時に主張や根拠を表示させることで、アウトラインの作成や文章作成の際に確認を行い、主張からはずれた意見が出ることを防ぐことができると考えこのような機能にした。主張のテキストボックス内では課題に対した自分の主張を記入し、根拠のテキストボックスでは、下調べを行った際根拠となるものを記入する。

#### 7.2.2 課題に対する疑問とその答えの記入

この機能では文章をアカデミックライティングの特徴である「問い合わせ」と「答え」の形式で記述を行うことで、文章に必要な情報などを明確化していくことができると考えこのような機能にした。また記入する内容としては、課題のテーマに対しての疑問とそれに対する自分の考えである答えをそれぞれ「問い合わせ」と「答え」の部分に記入していく。

#### 7.2.3 論理的な構成の整理

この機能では、一般的なアウトラインプロセッサと同様に論理的な文章を書く上で各内容の順番や情報を整理するため順番を入れ替える機能、章や段落の情報を表示する機能にした。具体的にはボタンを押すことによって上下の内容が入れ替わる機能になっている。

#### 7.2.4 参考文献の管理

文章を作成する際下調べなどで行った引用した文献や本などを確認、整理する機能にした。ここでは閲覧したサイトや検索した文献のWebページなどのURLを管理することや何章で扱う参考文献であるかを明確に表示することで参考文献の管理をしやすくするためこのような機能にした。

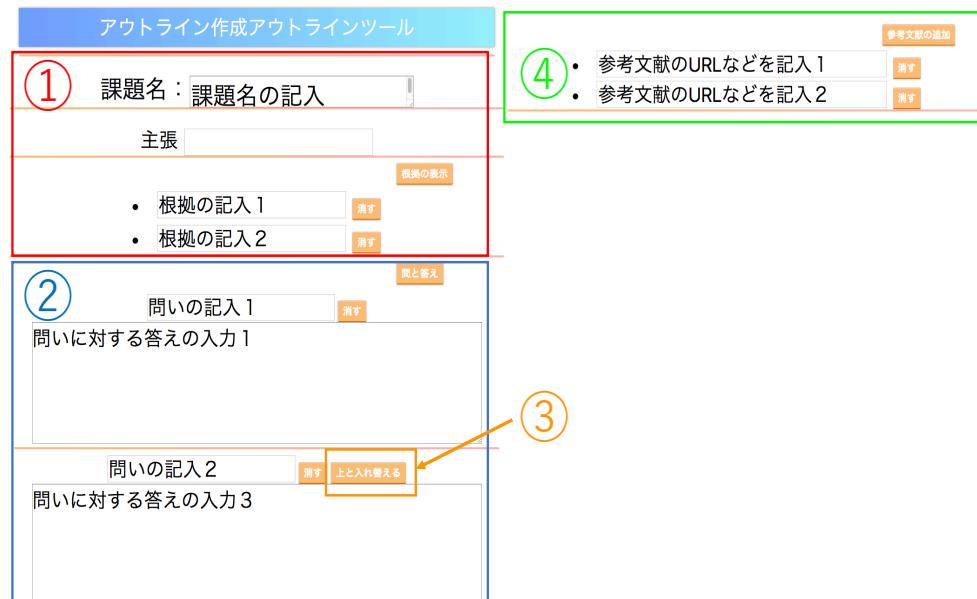


図 8: アカデミックアウトラインツール

### 7.3 本ツールの利用方法

本ツールの利用目的は、論文やレポートなどのアカデミックライティングのアウトラインの作成を支援することである。そのため実際に文章を書くのは別のアプリケーションで行う。本章では7.2に記述されている、①～④の4つの機能の利用方法を以下に示す。

#### 7.3.1 ホーム画面

図9は、本ツールの起動時の画面である。

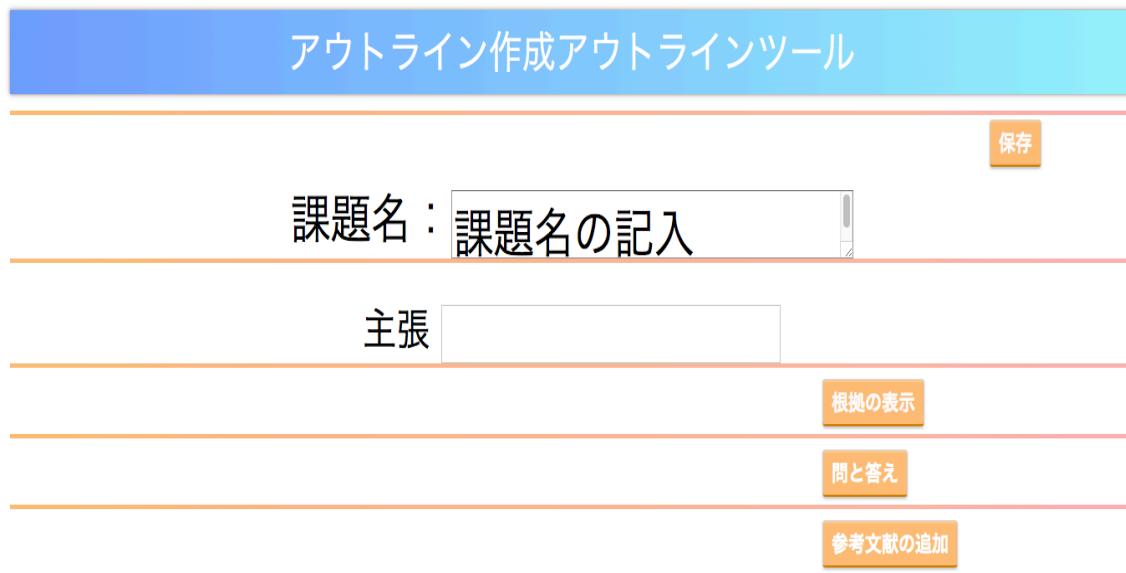


図 9: 起動時の画面

### 7.3.2 課題名の記入欄

図 10 は課題名の記入欄であり、論文のテーマやレポートの課題名などを記入する。

---

課題名 :  保存

---

図 10: 課題名の記入欄

### 7.3.3 主張の記入欄

図 11 は主張の記入欄であり、テーマや課題に対しての自分が考える主張を記入する。

---

主張

---

図 11: 主張の記入欄

#### 7.3.4 根拠の記入欄

根拠の表示のボタンを押すことで根拠を記入するテキストボックスが表示される。下調べを行った際の根拠となるものを記入していく。またアウトラインを作成し、整理していく中で不要になったものがあった際には消すボタンでその根拠を消すことができる。表示前を図 12 に表示後を図 13 に示す。



図 12: 根拠の表示前

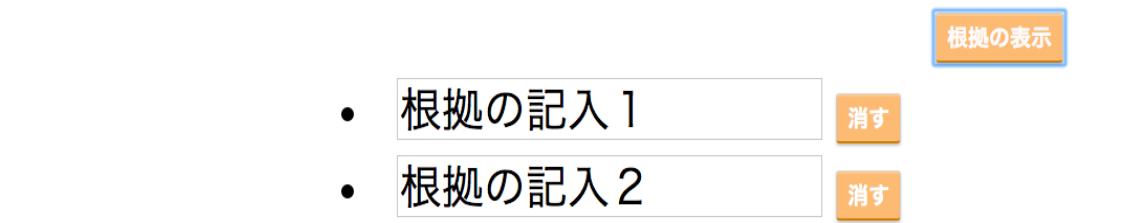


図 13: 根拠の表示後

### 7.3.5 問いと答えの記入欄

問い合わせボタンを押すことで、問い合わせの記入と答えの記入欄が表示される。テーマや課題に対しての疑問や何を論じるのかを問い合わせの部分に記入をする。またそれに対応する答えを記入する。書いた文章の整理を行うため、上と入れ替えるボタンを押すことで上下のテキストボックス内の記述内容を入れ替えることができ、論理的なアウトラインの作成を支援することができる。表示前を図14に表示後を図15に示す。



図 14: 問いと答えの表示前

A screenshot of the same user interface after input. It shows two stacked input fields. The top field is labeled '問い合わせ 1' (Question 1) and contains the text '問い合わせに対する答えの入力 1' (Input for the answer to the question). To its right are two buttons: '消す' (Delete) and '上と入れ替える' (Replace with top). The bottom field is labeled '問い合わせ 2' (Question 2) and contains the text '問い合わせに対する答えの入力 2' (Input for the answer to the question). To its right are also two buttons: '消す' (Delete) and '上と入れ替える' (Replace with top). Both fields have a light gray background and are enclosed in a thin gray border.

図 15: 問いと答えの表示後

### 7.3.6 参考文献の管理

参考文献のボタンを押すことで引用したサイトのURLなどを記入する欄が表示される。下調べの際に見た本やサイトを記入し管理をしておく。管理する際に何章で引用する文献なのか記入しておく。また参考文献の記入欄を追加する際は、参考文献のボタンをもう一度押すことで、記入欄が増える。表示前を図16に表示後を図17に示す。



図 16: 参考文献の表示前

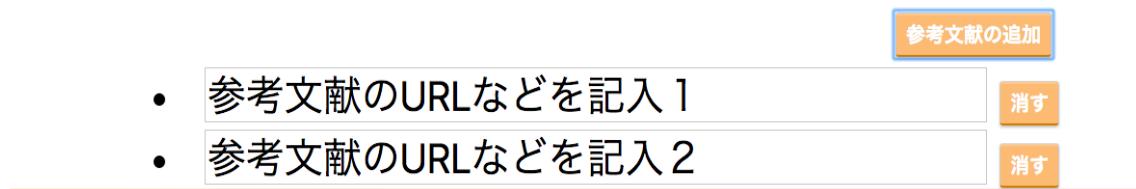


図 17: 参考文献の表示後

## 7.4 画面構成について

本ツールはPCでの作業だけでなく、移動時間などの隙間時間での利用も考えたため、スマートフォンでの利用もすることができる。そのため機能は変わらないがPC版とスマートフォン版とipad版の3種類に対応できるよう画面サイズに合わせた画面構成を行った。以下の図がそれぞれPCでの画面表示を図18、スマートフォンでの画面表示とipadでの画面表示を図19に示す。

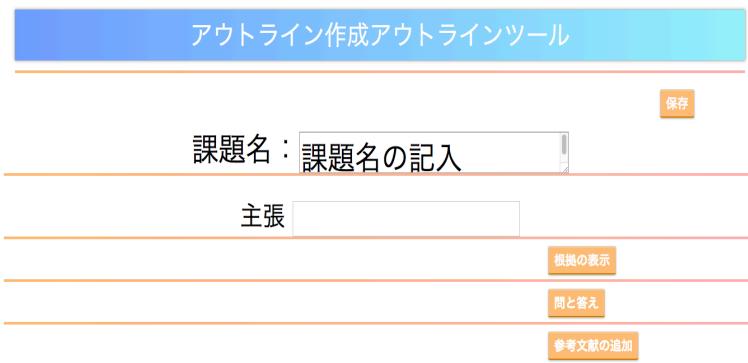


図 18: PC での画面表示

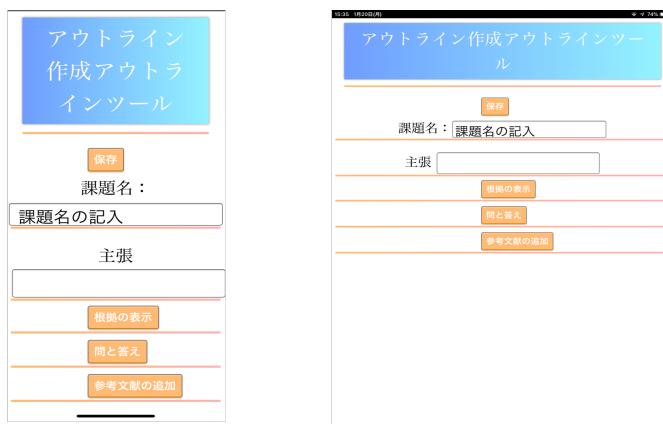


図 19: スマートフォンでの画面表示とiPadでの画面表示

## 8 結言

大学で論文やレポートを書かなければならぬ大学生に対して、論理的な思考力や論理的文章作成能力の要求が高まっている。しかし、論文やレポートを書く際にアウトラインなどの事前準備をせずに文章の作成を行ってしまう学生が多く、論理的な文章にならぬことが問題点として挙げられる。そのため、レポートの書き方の指導や修正を行うライティングセンターの設置などが進められているが継続的な利用が必要とされている。

しかし、一般に論文や小説などの長文を作成を支援するためのツールとして、アウトラインプロセッサが使用されることが多い。これは、文章を階層的に管理することに主眼が置かれており、学生にとって主張や根拠などが明確な一貫した文章を書く力を養うためのツールではないことが問題点となっている。

そこで本制作では、主張や根拠などが明確な一貫した論文やレポートを書くため、準備段階であるアカデミックアウトラインツールを開発することを目的としている。

本制作を行い論理的文章のアウトラインの作成を支援するツールの開発を行った。

## 謝辞

本制作の遂行および本論文の作成にあたり、多くの御助言とご指導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。

## 参考文献

- [1] 山崎 憲一, 萬代 雅希:”論文とは”, 電子情報通信学会 通信ソサイエティマガジン, 2016 年 9 卷 4 号 216-221. [https://www.jstage.jst.go.jp/article/bplus/9/4/9\\_216/\\_pdf](https://www.jstage.jst.go.jp/article/bplus/9/4/9_216/_pdf)
- [2] 堀 一成, 坂尻 彰宏:”阪大生のためのアカデミックライティング”, <https://ir.library.osaka-u.ac.jp/repo/ouka/all/27153/Academic%20Writing%20Introduction.pdf>, 2019/8/23 参照
- [3] 早稲田ウィークリー:”アカデミック・ライティング力を磨こう”, <https://www.waseda.jp/inst/weekly/feature/2014/06/23/20860/>
- [4] 山本 浩: ”TeX を使った論文作成方法”, 2000 年 103 卷 984 号 770-773. [https://www.jstage.jst.go.jp/article/jsmemag/103/984/103\\_KJ00001459868/\\_article/-char/ja/](https://www.jstage.jst.go.jp/article/jsmemag/103/984/103_KJ00001459868/_article/-char/ja/)
- [5] Lucidchart:”5 分でわかる、マインドマップの書き方と意味”,[https://www.lucidchart.com/pages/ja/mind-map#section\\_0](https://www.lucidchart.com/pages/ja/mind-map#section_0)
- [6] マインドマップの学校:”マインドマップの書き方・描き方「6 つの法則」”,<https://www.mindmap-school.jp/mindmap/mindmap-law/>

- [7] ディーエムソリューションズ株式会社:”PWA とは?メリットと実装例について”<https://digital-marketing.jp/seo/what-is-progressive-web-apps/#i-6>
- [8] 大嶋 泰史:”HTML 5 に対応した仮想座標グラフィックライブラリの構築”, 千葉工業大学卒業論文, 2012
- [9] 田口 優希: ”MeSH 拡張のためのサーバ感通信アーキテクチャ”, 千葉工業大学卒業論文, 2012
- [10] WEMO.tech:”CSS でメディアクエリ (Media Queries) の基本的な書き方、記述の意味を理解し、「何となく使う」を卒業する。”<https://wemo.tech/839>

## 付録A 作成したソースコード

### index.html

```
1  <!DOCTYPE html>
2  <html lang="ja">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1"
7      <script src="ex2.js"></script>
8      <script src="nylon.js"></script>
9      <script src="page.js"></script>
10     <script src="service_worker.js"></script>
11     <link rel="manifest" href="/manifest.json">
12     <link rel="stylesheet" type="text/css" media="(min-width: 640px)">
13     <link rel="stylesheet" type="text/css" media="(max-width: 640px)">
14     <link rel="stylesheet" type="text/css" media="(max-width: 800px)">
15     <link rel="stylesheet" type="text/css" href="face01.css">
16     <link rel="stylesheet" type="text/css" href="face01_mb.css">
17
18
19     <title>アウトラインツール</title>
20
21     <script>
22         if ('serviceWorker' in navigator) {
23             navigator.serviceWorker.register('/service_worker.js')
24                 .then(function() {
25                     console.log('Service worker registered!');
26                 });
27         }
28     </script>
29
30     </head>
31
32     <body>
```

```
33      <header class="hed">
34          <div class="face01hed">
35              アウトライン作成アウトラインツール
36          </div>
37      </header>
38  <div class="savediv"><button class="savebtn">保存</button></div>
39      <div class="hed2">
40          <div class="oya01">
41              <div class="chi02"><label for="kadai">課題名:<textarea class="chi02" name="kadai">課題名の記入</textarea></label>
42          </div>
43      </div>
44  </div>
45
46      <!--主張の部分-->
47      <p></p>
48      <div class="shu">
49          <form>
50              <label>主張</label>
51              <input class="chou" type="text" name="namae" id="shuchou">
52          </form>
53      </div>
54
55
56      <!--根拠の部分-->
57      <div class="kon_st_div">
58          <button id="b_kon" class="kon_st">根拠の表示</button>
59      </div>
60
61      <div id="kon_hontai" class="kon_hon"></div>
62
63      <template id="temp0">
64
65          <li class="kk">
66              <input type="text" name="konkon" class="textK" >
67              <button class="delete2">消す</button>
```

```
68      </li>
69  </template>
70
71  <!--問と答えの部分-->
72  <div class="panda_div">
73      <button id="b_qanda" class="btnpanda">問と答え</button>
74  </div>
75
76  <div id="qa_hontai" class="qa_hontai"></div>
77
78  <template id="temp1_noup">
79      <div class="qa">
80          <input type="text" name="dummy" class="textQ"> <button cla
 消す</button>
81          <br>
82          <textarea name="dummy2" cols="50" rows="5" class="textA">
    もげもげ</textarea>
83
84
85      </div>
86  </template>
87  <template id="temp1">
88      <div class="qa">
89          <input type="text" name="dummy" class="textQ"> <button cla
 消す</button> <button
90              class="change">上と入れ替える</button>
91
92          <br>
93          <textarea name="dummy2" cols="50" rows="5" class="textA">
    もげもげ</textarea>
94      </div>
95  </template>
96
97  <!--参考文献-->
98  <div class="sanst_div">
99      <button id="san_start" class="sanst">参考文献の追加</button>
```

```
100     </div>
101
102     <div id="san_hontai" class="san_hon"></div>
103
104     <template id="temp3">
105
106         <li class="ss">
107             <input type="text" name="sansan" class="textS">
108             <button class="delete3">消す</button>
109         </li>
110     </template>
111
112
113 </body>
114
115 </html>
```

## face01.css

```
1 @media print, screen and (min-width: 640px){  
2 /*ヘッダーのcss*/  
3 .face01hed{  
4     font-size: 40px;  
5     text-align: center;  
6     padding-bottom: 5px;  
7     padding: 0.2em 0.5em;  
8     background: -webkit-linear-gradient(to right, rgb(111, 157, 255),  
9     background: linear-gradient(to right, rgb(111, 157, 255), #94f3ff)  
10    color: white;  
11    font-weight: lighter;  
12    box-shadow: 0 0 4px rgba(0, 0, 0, 0.56);  
13    margin-bottom: 15px;  
14 }  
15  
16 }
```

## index.css

```
1  @media print, screen and (min-width: 640px){  
2      .savediv{  
3          width: 300px;  
4          margin: auto;  
5          position: relative;  
6          padding-bottom: 50px;  
7      }  
8      .savebtn{  
9          font-size: 20px;  
10         position: absolute;  
11         left: 650px;  
12  
13         display: inline-block;  
14         padding: 0.25em 0.5em;  
15         text-decoration: none;  
16         color: rgb(255, 254, 254);  
17         background: rgb(255, 186, 115);/*背景色*/  
18         border-bottom: solid 2px #ce7d05; /*少し濃  
い目の色に*/  
19         border-radius: 4px; /*角の丸み*/  
20         box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2  
21         font-weight: bold;  
22     }  
23     .savebtn:active{  
24         border-bottom: solid 2px #fd9535;  
25         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);  
26     }  
27     .oya01{  
28  
29         display: flex;  
30         -webkit-flex-direction: row;  
31         flex-direction: row;  
32         justify-content: center;  
33     }
```

```
34 .hed{  
35     position: relative;  
36     padding: 0.25em 0;  
37 }  
38  
39 .hed:after{  
40     content: "";  
41     display: block;  
42     height: 4px;  
43     background: -webkit-linear-gradient(to right, rgb(255,  
44     background: linear-gradient(to right, rgb(255, 186, 11  
45 )}  
46 .hed2{  
47     position: relative;  
48     padding: 0.25em 0;  
49 }  
50  
51 .hed2:after{  
52     content: "";  
53     display: block;  
54     height: 4px;  
55     background: -webkit-linear-gradient(to right, rgb(255,  
56     background: linear-gradient(to right, rgb(255, 186, 11  
57 )}  
58 .lab{  
59     position: relative;  
60     padding: 0.25em 0;  
61 }  
62 .lab:after{  
63     content: "";  
64     display: block;  
65     height: 4px;  
66     background: -webkit-linear-gradient(to right, rgb(255,  
67     background: linear-gradient(to right, rgb(255, 186, 11  
68 )}  
69
```

```
70 .tex{  
71     font-size: 48px;  
72     vertical-align: middle;  
73 }  
74  
75 .chi02{  
76     /*border: solid #800015;*/  
77     flex-grow: 5;  
78     text-align: center;  
79     font-size: 48px;  
80 }  
81 .shu{  
82     font-size: 40px;  
83     text-align: center;  
84     position: relative;  
85     padding: 0.25em 0;  
86 }  
87 .shu::after{  
88     content: "";  
89     display: block;  
90     height: 4px;  
91     background: -webkit-linear-gradient(to right, rgb(255,  
92         background: linear-gradient(to right, rgb(255, 186, 11  
93     })  
94     .shu input{  
95         vertical-align : middle;  
96     }  
97     .chou{  
98         font-size: 40px;  
99         vertical-align: middle;  
100    }  
101    .ul{  
102        text-align: center;  
103        font-size: 24px;  
104    }  
105    .kon{
```

```
106         font-size: 40px;
107         vertical-align: middle;
108         position: relative;
109
110     }
111     .kon:after{
112         content: "";
113         display: block;
114         height: 4px;
115         background: -webkit-linear-gradient(to right, rgb(255,
116         background: linear-gradient(to right, rgb(255, 186, 11
117     }
118     .konkon{
119         font-size: 40px;
120         vertical-align: middle;
121         padding: 0.25em 0;
122     }
123     .kt{
124         font-size: 40px;
125         vertical-align: middle;
126     }
127     .toi{
128         position: relative;
129         padding: 0.25em 0;
130     }
131     .toi:after{
132         content: "";
133         display: block;
134         height: 4px;
135         background: -webkit-linear-gradient(to right, rgb(255,
136         background: linear-gradient(to right, rgb(255, 186, 11
137     }
138     .toittoi{
139         font-size: 40px;
140         vertical-align: middle;
141     }
```

```
142 /*.ans{
143         border: solid #a19894;
144     }*/
145 .ansans{
146         font-size: 40px;
147         vertical-align: middle;
148 }
149 .number{
150         font-size: 40px;
151         vertical-align: middle;
152 }
153 .toians{
154         font-size: 40px;
155
156 }
157 .sankou{
158
159         font-size: 40px;
160
161 }
162 .uemoto{
163         font-size: 40px;
164 }
165 .btn{
166         font-size: 40px;
167 }
168 /*こんきよ*/
169 .kon_st_div{
170         width: 300px;
171         margin: auto;
172         position: relative;
173         padding-bottom: 50px;
174 }
175 .kon_st{
176         font-size: 20px;
177         position: absolute;
```

```
178         left: 450px;
179
180         display: inline-block;
181         padding: 0.25em 0.5em;
182         text-decoration: none;
183         color: rgb(255, 254, 254);
184         background: rgb(255, 186, 115);/*背景色*/
185         border-bottom: solid 2px #ce7d05; /*少し濃
い目の色に*/
186         border-radius: 4px; /*角の丸み*/
187         box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2
188         font-weight: bold;
189     }
190     .kon_st:active{
191         border-bottom: solid 2px #fd9535;
192         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
193     }
194     .kon_hon{
195         font-size: 20px;
196         text-align:center;
197     }
198     .kon_hon::after{
199         content: "";
200         display: block;
201         height: 4px;
202         background: -webkit-linear-gradient(to right, rgb(255,
203         background: linear-gradient(to right, rgb(255, 186, 11
204     }
205     .kk{
206
207         font-size: 40px;
208         padding: 3.0px 0;
209     }
210     .textK{
211         font-size: 40px;
212     }
```

```
213 .delete2{  
214     font-size: 20px;  
215  
216     display: inline-block;  
217     padding: 0.25em 0.5em;  
218     text-decoration: none;  
219     color: #FFF;  
220     background: rgb(255, 186, 115);/*背景色*/  
221     border-bottom: solid 2px #d27d00; /*少し濃  
    い目の色に*/  
222     border-radius: 4px; /*角の丸み*/  
223     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2  
224         font-weight: bold;  
225 }  
226 .delete2:active{  
227     border-bottom: solid 2px #fd9535;  
228     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);  
229 }  
230 /**/  
231 /*toioans*/  
232 .panda_div{  
233     width: 300px;  
234     margin: auto;  
235     margin-top: 10px;  
236     position: relative;  
237     padding-bottom: 50px;  
238 }  
239 .btntpanda{  
240     font-size: 20px;  
241     position: absolute;  
242     left: 450px;  
243  
244     display: inline-block;  
245     padding: 0.25em 0.5em;  
246     text-decoration: none;  
247     color: #FFF;
```

```
248         background: rgb(255, 186, 115);/*背景色*/
249         border-bottom: solid 2px #d27d00; /*少し濃
い目の色に*/
250         border-radius: 4px; /*角の丸み*/
251         box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2
252         font-weight: bold;
253     }
254     .btntpanda:active{
255         border-bottom: solid 2px #fd9535;
256         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
257     }
258     .qa{
259         vertical-align: middle;
260         position: relative;
261         text-align:center;
262         margin: 10px;
263     }
264     .qa:after{
265         content: "";
266         display: block;
267         height: 4px;
268         background: -webkit-linear-gradient(to right, rgb(255,
269         background: linear-gradient(to right, rgb(255, 186, 11
270     }
271     .textQ{
272         font-size: 40px;
273         padding: 1.0px 0;
274         margin: 2px 0px;
275     }
276     .textA{
277         font-size: 40px;
278         padding: 1.0px 0;
279     }
280     .toisuta{
281         font-size: 20px;
282         margin: 5px 0px;
```

```
283 }
284 .delete{
285     font-size: 20px;
286
287     display: inline-block;
288     padding: 0.25em 0.5em;
289     text-decoration: none;
290     color: #FFF;
291     background: rgb(255, 186, 115);/*背景色*/
292     border-bottom: solid 2px #d27d00; /*少し濃
い目の色に*/
293     border-radius: 4px; /*角の丸み*/
294     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2
295     font-weight: bold;
296 }
297 .delete:active{
298     border-bottom: solid 2px #fd9535;
299     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
300 }
301 .change{
302     font-size: 20px;
303
304     display: inline-block;
305     padding: 0.25em 0.5em;
306     text-decoration: none;
307     color: #FFF;
308     background: rgb(255, 186, 115);/*背景色*/
309     border-bottom: solid 2px #d27d00; /*少し濃
い目の色に*/
310     border-radius: 4px; /*角の丸み*/
311     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2
312     font-weight: bold;
313 }
314 .change:active{
315     border-bottom: solid 2px #fd9535;
316     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
```

```
317 }
318 .qa_hontai{
319     font-size: 20px;
320 }
321 .qa_hontai::after{
322     content: "";
323     display: block;
324     height: 4px;
325     background: -webkit-linear-gradient(to right, rgb(255,
326         background: linear-gradient(to right, rgb(255, 186, 115));
327 }
328 /**sannkoubunnkenn*/
329 .sanst_div{
330     width: 300px;
331     margin: auto;
332     margin-top: 10px;
333     position: relative;
334     padding-bottom: 50px;
335 }
336 .sanst{
337     font-size: 20px;
338     position: absolute;
339     left: 450px;
340
341     display: inline-block;
342     padding: 0.25em 0.5em;
343     text-decoration: none;
344     color: #FFF;
345     background: rgb(255, 186, 115);/*背景色*/
346     border-bottom: solid 2px #d27d00; /*少し濃い目の色に*/
347     border-radius: 4px; /*角の丸み*/
348     box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 5px 0 #d27d00;
349     font-weight: bold;
350 }
351 .sanst:active{
```

```
352         border-bottom: solid 2px #fd9535;
353         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
354     }
355     .san_hon{
356         font-size: 20px;
357         text-align:center;
358     }
359     .san_hon::after{
360         content: "";
361         display: block;
362         height: 4px;
363         background: -webkit-linear-gradient(to right, rgb(255,
364             background: linear-gradient(to right, rgb(255, 186, 115),
365         }
366     .ss{
367
368         font-size: 40px;
369     }
370     .textS{
371         font-size: 40px;
372         width: 60%;
373     }
374     .delete3{
375         font-size: 20px;
376
377         display: inline-block;
378         padding: 0.25em 0.5em;
379         text-decoration: none;
380         color: #FFF;
381         background: rgb(255, 186, 115);/*背景色*/
382         border-bottom: solid 2px #d27d00; /*少し濃い目の色に*/
383         border-radius: 4px; /*角の丸み*/
384         box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px
385         font-weight: bold;
386     }
```

```
387     .delete3:active{  
388         border-bottom: solid 2px #fd9535;  
389         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);  
390     }  
391 }
```

## indexmb.css

```
1  @media print, screen and (max-width: 479px){  
2    img{  
3      max-width: 100%;  
4      height: auto;  
5      width: auto;  
6    }  
7    .savediv{  
8      width: 300px;  
9      margin: auto;  
10     position: relative;  
11     padding-bottom: 50px;  
12   }  
13   .savebtn{  
14     font-size: 20px;  
15     position: absolute;  
16     left: 100px;  
17     display: inline-block;  
18     padding: 0.25em 0.5em;  
19     text-decoration: none;  
20     color: rgb(255, 254, 254);  
21     background: rgb(255, 186, 115);/*背景色*/  
22     border-bottom: solid 2px #ce7d05; /*少し濃い目の色に  
   */  
23     border-radius: 4px; /*角の丸み*/  
24     box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 2px rgba(255, 186, 115, 0.1);  
25     font-weight: bold;  
26   }  
27   .savebtn:active{  
28     border-bottom: solid 2px #fd9535;  
29     box-shadow: 0 0 2px rgba(0, 0, 0, 0.3);  
30   }  
31   .oya01{  
32     display: flex;
```

```
34         -webkit-flex-direction: row;
35         flex-direction: row;
36         justify-content: center;
37     }
38     .hed{
39         position: relative;
40         padding: 0.25em 0;
41     }
42
43     .hed:after{
44         content: "";
45         display: block;
46         height: 4px;
47         background: -webkit-linear-gradient(to right, rgb(255, 186, 115),
48         background: linear-gradient(to right, rgb(255, 186, 115), #ffb6c1);
49     }
50     .hed2{
51         position: relative;
52         padding: 0.25em 0;
53     }
54
55     .hed2:after{
56         content: "";
57         display: block;
58         height: 4px;
59         background: -webkit-linear-gradient(to right, rgb(255, 186, 115),
60         background: linear-gradient(to right, rgb(255, 186, 115), #ffb6c1);
61     }
62     .lab{
63         position: relative;
64         padding: 0.25em 0;
65     }
66     .lab:after{
67         content: "";
68         display: block;
69         height: 4px;
```

```
70         background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffb703);
71         background: linear-gradient(to right, rgb(255, 186, 115), #ffb703);
72     }
73
74     .tex{
75         font-size: 20px;
76         vertical-align: middle;
77     }
78
79     .chi02{
80         flex-grow: 5;
81         text-align: center;
82         font-size: 20px;
83     }
84     .shu{
85         font-size: 20px;
86         text-align: center;
87         position: relative;
88         padding: 0.25em 0;
89     }
90     .shu::after{
91         content: "";
92         display: block;
93         height: 4px;
94         background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffb703);
95         background: linear-gradient(to right, rgb(255, 186, 115), #ffb703);
96     }
97     .shu input{
98         vertical-align : middle;
99     }
100    .chou{
101        font-size: 20px;
102        vertical-align: middle;
103    }
104    .ul{
105        text-align: center;
```

```
106         font-size: 24px;
107     }
108     .kon{
109         font-size: 40px;
110         vertical-align: middle;
111         position: relative;
112     }
113 }
114 .kon:after{
115     content: "";
116     display: block;
117     height: 4px;
118     background: -webkit-linear-gradient(to right, rgb(255, 186, 115),
119     background: linear-gradient(to right, rgb(255, 186, 115), #ffb703);
120 }
121 .konkon{
122     font-size: 40px;
123     vertical-align: middle;
124     padding: 0.25em 0;
125 }
126 .kt{
127     font-size: 40px;
128     vertical-align: middle;
129 }
130 .toi{
131     position: relative;
132     padding: 0.25em 0;
133 }
134 .toi:after{
135     content: "";
136     display: block;
137     height: 4px;
138     background: -webkit-linear-gradient(to right, rgb(255, 186, 115),
139     background: linear-gradient(to right, rgb(255, 186, 115), #ffb703);
140 }
141
```

```
142 /*こんきよ*/
143 .kon_st_div{
144     width: 300px;
145     margin: auto;
146     position: relative;
147     padding-bottom: 50px;
148 }
149 .kon_st{
150     font-size: 20px;
151     position: absolute;
152     left: 100px;
153
154     display: inline-block;
155     padding: 0.25em 0.5em;
156     text-decoration: none;
157     color: rgb(255, 254, 254);
158     background: rgb(255, 186, 115);/*背景色*/
159     border-bottom: solid 2px #ce7d05; /*少し濃い目の色に
*/
160     border-radius: 4px; /*角の丸み*/
161     box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 2px rgba(0, 0, 0, 0.1);
162     font-weight: bold;
163 }
164 .kon_st:active{
165     border-bottom: solid 2px #fd9535;
166     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
167 }
168 .kon_hon{
169     font-size: 20px;
170     text-align: center;
171 }
172 .kon_hon::after{
173     content: "";
174     display: block;
175     height: 4px;
176     background: -webkit-linear-gradient(to right, rgb(255, 186, 115,
```

```
177         background: linear-gradient(to right, rgb(255, 186, 115), #ffb
178     }
179     .kk{
180
181         font-size: 10px;
182         padding: 3.0px 0;
183     }
184     .textK{
185         font-size: 20px;
186     }
187     .delete2{
188         font-size: 20px;
189
190         display: inline-block;
191         padding: 0.25em 0.5em;
192         text-decoration: none;
193         color: #FFF;
194         background: rgb(255, 186, 115);/*背景色*/
195         border-bottom: solid 2px #d27d00; /*少し濃い目の色に
 */
196         border-radius: 4px; /*角の丸み*/
197         box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba
198         font-weight: bold;
199     }
200     .delete2:active{
201         border-bottom: solid 2px #fd9535;
202         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
203     }
204     /**
205     /*toioans*/
206     .panda_div{
207         width: 100px;
208         margin: auto;
209         margin-top: 10px;
210         position: relative;
211         padding-bottom: 50px;
```

```
212 }
213 .btntpanda{
214     font-size: 20px;
215     position: absolute;
216     left: 100px;
217
218     display: inline-block;
219     padding: 0.25em 0.5em;
220     text-decoration: none;
221     color: #FFF;
222     background: rgb(255, 186, 115);/*背景色*/
223     border-bottom: solid 2px #d27d00; /*少し濃い目の色に
 */
224     border-radius: 4px; /*角の丸み*/
225     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(255,255,255,0.1);
226     font-weight: bold;
227 }
228 .btntpanda:active{
229     border-bottom: solid 2px #fd9535;
230     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
231 }
232 .qa{
233     vertical-align: middle;
234     position: relative;
235     text-align:center;
236     margin: 10px;
237 }
238 .qa:after{
239     content: "";
240     display: block;
241     height: 4px;
242     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffbd00);
243     background: linear-gradient(to right, rgb(255, 186, 115), #ffbd00);
244 }
245 .textQ{
246     font-size: 20px;
```

```
247         padding: 1.0px 0;
248         margin: 2px 0px;
249
250     }
251     input.textA{
252         font-size: 20px;
253         padding: 1.0px 0;
254         width: 30%;
255     }
256     .toisuta{
257         font-size: 20px;
258         margin: 5px 0px;
259     }
260     .delete{
261         font-size: 20px;
262
263         display: inline-block;
264         padding: 0.25em 0.5em;
265         text-decoration: none;
266         color: #FFF;
267         background: rgb(255, 186, 115);/*背景色*/
268         border-bottom: solid 2px #d27d00; /*少し濃い目の色に
269 */
270         border-radius: 4px; /*角の丸み*/
271         box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(0,0,0,0.1);
272         font-weight: bold;
273     }
274     .delete:active{
275         border-bottom: solid 2px #fd9535;
276         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
277     }
278     .change{
279         font-size: 20px;
280
281         display: inline-block;
282         padding: 0.25em 0.5em;
```

```
282     text-decoration: none;
283     color: #FFF;
284     background: rgb(255, 186, 115);/*背景色*/
285     border-bottom: solid 2px #d27d00; /*少し濃い目の色に
286     */
287     border-radius: 4px; /*角の丸み*/
288     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(255,255,255,0.1);
289     font-weight: bold;
290   }
291   .change:active{
292     border-bottom: solid 2px #fd9535;
293     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
294   }
295   .qa_hontai{
296     font-size: 20px;
297   }
298   .qa_hontai::after{
299     content: "";
300     display: block;
301     height: 4px;
302     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffbd99);
303     background: linear-gradient(to right, rgb(255, 186, 115), #ffbd99);
304   }
305   /**sannkoubunnkenn*/
306   .sanst_div{
307     width: 300px;
308     margin: auto;
309     margin-top: 10px;
310     position: relative;
311     padding-bottom: 50px;
312   }
313   .sanst{
314     font-size: 20px;
315     position: absolute;
316     left: 100px;
```

```
317     display: inline-block;
318     padding: 0.25em 0.5em;
319     text-decoration: none;
320     color: #FFF;
321     background: rgb(255, 186, 115);/*背景色*/
322     border-bottom: solid 2px #d27d00; /*少し濃い目の色に
323 */
324     border-radius: 4px; /*角の丸み*/
325     box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 2px rgba(255, 186, 115, 0.2);
326     font-weight: bold;
327 }
328 .sanst:active{
329     border-bottom: solid 2px #fd9535;
330     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
331 }
332 .san_hon{
333     font-size: 20px;
334     text-align:center;
335 }
336 .san_hon::after{
337     content: "";
338     display: block;
339     height: 4px;
340     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffbd99);
341     background: linear-gradient(to right, rgb(255, 186, 115), #ffbd99);
342 }
343 .ss{
344     font-size: 20px;
345 }
346 .textS{
347     font-size: 20px;
348     width: 60%;
349 }
350 .delete3{
351     font-size: 20px;
```

```
352
353     display: inline-block;
354     padding: 0.25em 0.5em;
355     text-decoration: none;
356     color: #FFF;
357     background: rgb(255, 186, 115);/*背景色*/
358     border-bottom: solid 2px #d27d00; /*少し濃い目の色に
359     */
360     border-radius: 4px; /*角の丸み*/
361     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(255,255,255,0.1);
362     font-weight: bold;
363 }
364 .delete3:active{
365     border-bottom: solid 2px #fd9535;
366     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
367 }
```

## indexmbipad.css

```
1  @media print, screen and (max-width: 1024px){  
2    img{  
3      max-width: 100%;  
4      height: auto;  
5      width: auto;  
6    }  
7    .savediv{  
8      width: 300px;  
9      margin: auto;  
10     position: relative;  
11     padding-bottom: 50px;  
12   }  
13   .savebtn{  
14     font-size: 20px;  
15     position: absolute;  
16     left: 100px;  
17  
18     display: inline-block;  
19     padding: 0.25em 0.5em;  
20     text-decoration: none;  
21     color: rgb(255, 254, 254);  
22     background: rgb(255, 186, 115);/*背景色*/  
23     border-bottom: solid 2px #ce7d05; /*少し濃い目の色に  
 */  
24     border-radius: 4px; /*角の丸み*/  
25     box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 2px rgba(255, 186, 115, 0.2);  
26     font-weight: bold;  
27   }  
28   .oya01{  
29  
30     display: flex;  
31     -webkit-flex-direction: row;  
32     flex-direction: row;  
33     justify-content: center;
```

```
34 }
35 .hed{
36     position: relative;
37     padding: 0.25em 0;
38 }
39
40 .hed:after{
41     content: "";
42     display: block;
43     height: 4px;
44     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffb300);
45     background: linear-gradient(to right, rgb(255, 186, 115), #ffb300);
46 }
47 .hed2{
48     position: relative;
49     padding: 0.25em 0;
50 }
51
52 .hed2:after{
53     content: "";
54     display: block;
55     height: 4px;
56     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffb300);
57     background: linear-gradient(to right, rgb(255, 186, 115), #ffb300);
58 }
59 .lab{
60     position: relative;
61     padding: 0.25em 0;
62 }
63 .lab:after{
64     content: "";
65     display: block;
66     height: 4px;
67     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffb300);
68     background: linear-gradient(to right, rgb(255, 186, 115), #ffb300);
69 }
```

```
70
71 .tex{
72     font-size: 30px;
73     vertical-align: middle;
74 }
75
76 .chi02{
77     /*border: solid #800015;*/
78     flex-grow: 5;
79     text-align: center;
80     font-size: 30px;
81 }
82 .shu{
83     font-size: 30px;
84     text-align: center;
85     position: relative;
86     padding: 0.25em 0;
87 }
88 .shu::after{
89     content: "";
90     display: block;
91     height: 4px;
92     background: -webkit-linear-gradient(to right, rgb(255, 186, 115),
93     background: linear-gradient(to right, rgb(255, 186, 115), #ffb6c1);
94 }
95 .shu input{
96     vertical-align : middle;
97 }
98 .chou{
99     font-size: 30px;
100    vertical-align: middle;
101 }
102 .ul{
103     text-align: center;
104     font-size: 24px;
105 }
```

```
106 .kon{  
107     font-size: 40px;  
108     vertical-align: middle;  
109     position: relative;  
110 }  
111 .kon:after{  
112     content: "";  
113     display: block;  
114     height: 4px;  
115     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffb300);  
116     background: linear-gradient(to right, rgb(255, 186, 115), #ffb300);  
117 }  
118 .konkon{  
119     font-size: 40px;  
120     vertical-align: middle;  
121     padding: 0.25em 0;  
122 }  
123 .kt{  
124     font-size: 40px;  
125     vertical-align: middle;  
126 }  
127 .toi{  
128     position: relative;  
129     padding: 0.25em 0;  
130 }  
131 .toi:after{  
132     content: "";  
133     display: block;  
134     height: 4px;  
135     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffb300);  
136     background: linear-gradient(to right, rgb(255, 186, 115), #ffb300);  
137 }  
138 .toittoi{  
139     font-size: 40px;  
140     vertical-align: middle;
```

```
142 }
143 .ansans{
144     font-size: 40px;
145     vertical-align: middle;
146 }
147 .number{
148     font-size: 40px;
149     vertical-align: middle;
150 }
151 .toians{
152     font-size: 40px;
153 }
154 }
155 .sankou{
156
157     font-size: 40px;
158 }
159 }
160 .uemoto{
161     font-size: 40px;
162 }
163 .btn{
164     font-size: 40px;
165 }
166 /*こんきよ*/
167 .kon_st_div{
168     width: 300px;
169     margin: auto;
170     position: relative;
171     padding-bottom: 50px;
172 }
173 .kon_st{
174     font-size: 20px;
175     position: absolute;
176     left: 100px;
177 }
```

```
178     display: inline-block;
179     padding: 0.25em 0.5em;
180     text-decoration: none;
181     color: rgb(255, 254, 254);
182     background:  rgb(255, 186, 115);/*背景色*/
183     border-bottom: solid 2px #ce7d05; /*少し濃い目の色に
184 */
185     border-radius: 4px; /*角の丸み*/
186     box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 2px rgba(255, 186, 115, 0.1);
187     font-weight: bold;
188 }
189 .kon_st:active{
190     border-bottom: solid 2px #fd9535;
191     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
192 }
193 .kon_hon{
194     font-size: 30px;
195     text-align:center;
196 }
197 .kon_hon::after{
198     content: "";
199     display: block;
200     height: 4px;
201     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffbf70);
202     background: linear-gradient(to right, rgb(255, 186, 115), #ffbf70);
203 }
204 .kk{
205     font-size: 10px;
206     padding: 3.0px 0;
207 }
208 .textK{
209     font-size: 30px;
210     width: 200px;
211 }
212 .delete2{
```

```
213         font-size: 20px;
214
215         display: inline-block;
216         padding: 0.25em 0.5em;
217         text-decoration: none;
218         color: #FFF;
219         background: rgb(255, 186, 115);/*背景色*/
220         border-bottom: solid 2px #d27d00; /*少し濃い目の色に
 */
221         border-radius: 4px; /*角の丸み*/
222         box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 2px rgba(255, 186, 115, 0.1);
223         font-weight: bold;
224     }
225     .delete2:active{
226         border-bottom: solid 2px #fd9535;
227         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
228     }
229     /**
230     *toioans*/
231     .panda_div{
232         width: 300px;
233         margin: auto;
234         margin-top: 10px;
235         position: relative;
236         padding-bottom: 50px;
237     }
238     .btnpanda{
239         font-size: 20px;
240         position: absolute;
241         left: 100px;
242
243         display: inline-block;
244         padding: 0.25em 0.5em;
245         text-decoration: none;
246         color: #FFF;
247         background: rgb(255, 186, 115);/*背景色*/
```

```
248         border-bottom: solid 2px #d27d00; /*少し濃い目の色に
249 */
250         border-radius: 4px; /*角の丸み*/
251         box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(255,255,255,0.1);
252         font-weight: bold;
253     }
254     .btnpanda:active{
255         border-bottom: solid 2px #fd9535;
256         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
257     }
258     .qa{
259         vertical-align: middle;
260         position: relative;
261         text-align:center;
262         margin: 10px;
263     }
264     .qa:after{
265         content: "";
266         display: block;
267         height: 4px;
268         background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffbf70);
269         background: linear-gradient(to right, rgb(255, 186, 115), #ffbf70);
270     }
271     .textQ{
272         font-size: 30px;
273         padding: 1.0px 0;
274         margin: 2px 0px;
275     }
276
277     .textA{
278         font-size: 30px;
279         padding: 1.0px 0;
280         width: 250px;
281     }
282
```

```
283 .toisuta{
284     font-size: 30px;
285     margin: 5px 0px;
286 }
287 .delete{
288     font-size: 20px;
289
290     display: inline-block;
291     padding: 0.25em 0.5em;
292     text-decoration: none;
293     color: #FFF;
294     background: rgb(255, 186, 115);/*背景色*/
295     border-bottom: solid 2px #d27d00; /*少し濃い目の色に
 */
296     border-radius: 4px; /*角の丸み*/
297     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(255,255,255,0.1);
298     font-weight: bold;
299 }
300 .delete:active{
301     border-bottom: solid 2px #fd9535;
302     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
303 }
304 .change{
305     font-size: 20px;
306
307     display: inline-block;
308     padding: 0.25em 0.5em;
309     text-decoration: none;
310     color: #FFF;
311     background: rgb(255, 186, 115);/*背景色*/
312     border-bottom: solid 2px #d27d00; /*少し濃い目の色に
 */
313     border-radius: 4px; /*角の丸み*/
314     box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(255,255,255,0.1);
315     font-weight: bold;
316 }
```

```
317 .change:active{
318     border-bottom: solid 2px #fd9535;
319     box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
320 }
321 .qa_hontai{
322     font-size: 20px;
323 }
324 .qa_hontai::after{
325     content: "";
326     display: block;
327     height: 4px;
328     background: -webkit-linear-gradient(to right, rgb(255, 186, 115), #ffbb78);
329     background: linear-gradient(to right, rgb(255, 186, 115), #ffbb78);
330 }
331 /**sankoubunnkenn*/
332 .sanst_div{
333     width: 300px;
334     margin: auto;
335     margin-top: 10px;
336     position: relative;
337     padding-bottom: 50px;
338 }
339 .sanst{
340     font-size: 20px;
341     position: absolute;
342     left: 100px;
343
344     display: inline-block;
345     padding: 0.25em 0.5em;
346     text-decoration: none;
347     color: #FFF;
348     background: rgb(255, 186, 115);/*背景色*/
349     border-bottom: solid 2px #d27d00; /*少し濃い目の色に
*/
350     border-radius: 4px; /*角の丸み*/
351     box-shadow: inset 0 2px 0 rgba(255, 255, 255, 0.2), 0 2px 2px rgba(255, 255, 255, 0.1);
352 }
```

```
352         font-weight: bold;
353     }
354     .sanst:active{
355         border-bottom: solid 2px #fd9535;
356         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
357     }
358     .san_hon{
359         font-size: 20px;
360         text-align:center;
361     }
362     .san_hon::after{
363         content: "";
364         display: block;
365         height: 4px;
366         background: -webkit-linear-gradient(to right, rgb(255, 186, 115),
367         background: linear-gradient(to right, rgb(255, 186, 115), #ffb703);
368     }
369     .ss{
370
371         font-size: 20px;
372     }
373     .textS{
374         font-size: 30px;
375         width: 60%;
376     }
377     .delete3{
378         font-size: 20px;
379
380         display: inline-block;
381         padding: 0.25em 0.5em;
382         text-decoration: none;
383         color: #FFF;
384         background: rgb(255, 186, 115);/*背景色*/
385         border-bottom: solid 2px #d27d00; /*少し濃い目の色に
 */
386         border-radius: 4px; /*角の丸み*/
```

```
387         box-shadow: inset 0 2px 0 rgba(255,255,255,0.2), 0 2px 2px rgba(255,255,255,0.1);
388         font-weight: bold;
389     }
390     .delete3:active{
391         border-bottom: solid 2px #fd9535;
392         box-shadow: 0 0 2px rgba(0, 0, 0, 0.30);
393     }
394 }
```

ex2.js

```

35      * @date 2019-12-13
36      * @param {any} element
37      * @param {number} id1 入れ替えるID
38      * @param {number} id2 入れ替えるID
39      * @returns {any}
40      */
41 replace_dom(element, id1, id2) {
42     console.log(73, id1, id2);
43     let new_elm = element.childNodes[id1].cloneNode();
44     let old_elm = element.replaceChild(new_elm, element.childNodes[id1]);
45     element.replaceChild(old_elm, element.childNodes[id1]);
46 }
47
48 /**
49  * 描述
50  * @date 2019-12-13
51  * @param {any} element Q&A情報を追加するDOM要素      ex: document.qu
52  * @param {number} num レポート番号      ex: 0
53  * @returns {any}
54 */
55
56 set_qanda() {
57     var report = this.report.report;
58     var num = this.number;
59     let temp_noup = document.querySelector('#temp1_noup').content;
60     let temp_up = document.querySelector('#temp1').content;
61     for (let number in report[num]["qanda"]) {
62         var temp;
63         if (number == 0) temp = temp_noup;
64         else temp = temp_up;
65         let item = report[num].qanda[number];
66         let content = temp.cloneNode(true);
67         // 問と答えを入れておく
68         content.querySelector('.textQ').value = item.q;
69         content.querySelector('.textA').value = item.a;
70         content.querySelector('.qa').setAttribute('id', 'qa' + num);

```

```

71     content.querySelector('.textQ').setAttribute('id','qid' +n
72     content.querySelector('.textA').setAttribute('id','aid' +n
73     console.log('id','aid' +number);
74
75
76 //移動ボタン
77 if (number != 0) {
78     content.querySelector('.change').addEventListener('cli
79         this,this.save_qa();
80         let chid = ev1.srcElement.parentNode.getAttribute(
81         let number = chid.slice(2);
82         console.log(96, report[num].qanda[number]);//
確認用
83         this.report.replace(report[num].qanda, number, num
84         while (this.parent.firstChild) this.parent.removeC
85         new nylon().emit('page', { page: '#new_report' })
86         console.log(report[num].qanda[number]);//
確認用
87         });
88     } else {
89         console.log(85, content.querySelector('.change'));
90
91     }
// 削除ボタン
92     content.querySelector('.delete').addEventListener('click',
93         let id = ev.srcElement.parentNode.getAttribute('id');
94         let number = id.slice(2);
95         report[num].qanda.splice(number, 1);
96         ev.srcElement.parentNode.parentNode.removeChild(ev.sr
97         console.log(number);
98     })
99     this.parent.appendChild(content);
100 }
101 }
102 save_qa() {
103     var report = this.report.report;

```

```

105     var num = this.number;
106     let rep = [];
107     let qa_element = document.querySelector('#qa_hontai');
108     for (let data of qa_element.querySelectorAll('.qa')) {
109         let q_text = data.querySelector('input').value;
110         let a_text = data.querySelector('textarea').value;
111         rep.push({ q: q_text, a: a_text });
112     }
113     this.report.report[num].qanda = rep;
114     console.log(131, this.report.report[num].qanda);
115 }
116 }
117
118 class Report {
119     constructor() {
120         this.report = [];
121         //グローバル変数は ex2.html にある
122         // テストデータの仮代入
123         this.report[0] = {};
124         this.report[0].qanda = [];
125         this.report[0].konkyo = [];
126         this.report[0].sankou = [];

127         // テストデータの細かい部分を仮代入
128         let qa = this.report[0].qanda;
129         let k = this.report[0].konkyo;
130         let r = this.report[0].sankou;
131         qa.push({ q: "問い合わせの記入 1 ", a: "問い合わせに対する答えの
    入力 1 " });
132         qa.push({ q: "問い合わせの記入 2 ", a: "問い合わせに対する答えの
    入力 2 " });
133         qa.push({ q: '問い合わせの記入 3 ', a: '問い合わせに対する答えの
    入力 3' });
134         k.push({ k: '根拠の記入 1 ' });
135         k.push({ k: '根拠の記入 2 ' });
136         r.push({ r: '参考文献の URLなどを記入 1 ' });
137

```

```

138         r.push({ r: '参考文献のURLなどを記入2' });
139     }
140
141     save() {
142         document.querySelector('.save').addEventListener('click', () =>
143             var mydata = document.querySelector('.shuchou');
144             console.log(これは)
145             localStorage.setItem('mydata', mydata);
146         })
147     }
148
149     set_data() {
150         document.querySelector('.save').addEventListener('click', () =>
151             localStorage.setItem('reminder', JSON.stringify(qa));
152         })
153
154     }
155 //根拠の表示
156     set_konkyo( element, num ) {
157         document.querySelector('#b_kon').addEventListener('click', () =>
158             let temp = document.querySelector('#temp0').content;
159
160             for( let number in this.report[num].konkyo ) {
161                 let item = this.report[num].konkyo[number];
162                 let content = temp.cloneNode(true);
163
164                 //根拠を入れておく
165                 content.querySelector('.textK').value = item.k;
166                 content.querySelector('.kk').setAttribute('id', 'kk' + number);
167                 content.querySelector('.textK').setAttribute('id', 'tk' + number);
168                 console.log( 'id', 'kk' + number );
169                 console.log('id', 'tk'+ number);
170
171             //削除ボタン
172             content.querySelector('.delete2').addEventListener('click', () =>
173                 this.report[num].konkyo.splice(number, 1);
174             )
175         )
176     }

```

```

174         let id2 = ev2.srcElement.parentNode.getAttribute('
175         let number2 = id2.slice(2);
176         this.report[num].konkyo.splice(number2,1);
177         ev2.srcElement.parentNode.parentNode.removeChild(e
178     })
179     element.appendChild( content );
180 }
181
182
183 );
184 }
185 //参考文献の表示
186 set_sankou(element, num) {
187     document.querySelector('#san_start').addEventListener('click',
188     let temp = document.querySelector('#temp3').content;
189
190     for (let number in this.report[num].sankou) {
191         let item = this.report[num].sankou[number];
192         let content = temp.cloneNode(true);
193
194         //根拠を入れておく
195         content.querySelector('.textS').value = item.r;
196         content.querySelector('.ss').setAttribute('id', 'ss' +
197         content.querySelector('.textS').setAttribute('id', 'si
198         console.log('id', 'ss' + number);
199         console.log('id', 'sid' + number);
200
201
202     //削除ボタン
203     content.querySelector('.delete3').addEventListener('cl
204         console.log(ev3.srcElement.parentNode);
205         let id3 = ev3.srcElement.parentNode.getAttribute('
206         let number3 = id3.slice(2);
207         this.report[num].sankou.splice(number3, 1);
208         ev3.srcElement.parentNode.parentNode.removeChild(e
209     })

```

```

210             element.appendChild(content);
211         }
212
213     });
214 }
215 }
216 //入れ替え処理(下
217 replace(qanda, id1, id2) {
218     let change = qanda[id1];
219     qanda[id1] = qanda[id2];
220     qanda[id2] = change;
221 }
222
223 }
224 class PageUI {
225     constructor() {
226         let nl = new nylon();
227         document.querySelector('#b_qanda').addEventListener('click', (
228             nl.emit("page", { page: '#new_report' });
229         ));
230     }
231 }
232
233 function save() {
234     var Jdata = {
235         shuchou: document.querySelector('.tex').value,
236         ckonkyo: document.querySelector('.chou').value,
237         konkyo1: document.querySelector('#tk0').value,
238         konkyo2: document.querySelector('#tk1').value,
239         q1: document.querySelector('#qid0').value,
240         a1: document.querySelector('#aid0').value,
241         q1: document.querySelector('#qid1').value,
242         a1: document.querySelector('#aid1').value,
243         sankou1: document.querySelector('#sid0').value,
244         sankou2: document.querySelector('#sid1').value
245

```

```
246     };
247     localStorage.setItem("savedata", JSON.stringify(Jdata));
248     console.log("保存しました");
249 }
```

## nylon.js

```
1
2  /**
3   * クライアント側nylonの本体
4   * nylom関係の機能を使う場合は必ず読み込んでください
5   */
6  class nylom {
7
8      /**
9       * constructor
10      * グローバル変数nylon, nylom.mapなどを初期化します
11      */
12      constructor() {
13          /**
14           * @type {hash}
15           */
16          this.nylom = window.nylom;
17
18          /**
19           * @type {window}
20           */
21          this.parent = window.parent;
22
23          if( window.nylom == null ) {
24              window.nylom = {};
25          }
26          if( window.nylom.map == null ) {
27              window.nylom.map = {};
28          }
29
30          /**
31           * @type {array}
32           */
33          this.map = window.nylom.map;
34      }
}
```

```

35
36    /**
37     * 受け取ったイベント情報をそのまま外部に渡すための関
38     * 数
39     * nylon.iFrame で使用します
40     * @param {function} fn - イベントを処理するコールバ
41     * ック関数
42     */
43     setPassThrough( fn ) {
44         if( window.nylon.passthrough == null ) {
45             window.nylon.passthrough = fn;
46         } else {
47             console.log( "Error by duplexy registration" )
48         }
49     /**
50      * キーワードと、キーワードに対する処理を登録する関数
51      * @param { string } keyword - キーワード
52      * @param { function } fn - コールバック関数
53      */
54     on( keyword, fn ) {
55         console.log( this );
56         if( this.map[ keyword ] == null ) {
57             //if( window != parent ) {
58             //    this.parent.postMessage( { "keyword": keyword },
59             //                           this.url );
60             console.log( "new keyword : " + keyword );
61             this.map[ keyword ] = [ new nylonfunc( fn, thi
62         } else {
63             this.map[ keyword ].push( new nylonfunc( fn, thi
64         }
65     }
66
67 /**
68  * イベントを起こす関数

```

```

69      * @param { string[] } keys - キーワードの配列
70      * @param { hash } params - パラメータ（ハッシュで与
    える）
71      */
72      emitByArray( keys, params ) {
73          if( window.nylon.passthrough != null ) {
74              window.nylon.passthrough( keys, params, this )
75          }
76          if( params == null ) {
77              params = {};
78          }
79          console.log( "-->" + params["keywords"] );
80
81          if( params["keywords"] == null ) {
82              console.log("params.keywordsがnull");
83              params["keywords"] = keys;
84          }
85
86          for( let key of keys ) {
87              if( this.map[ key ] == null ) {
88                  console.log( "Invarid keyword " + key
89              } else {
90                  for( let element of this.map[ key ] )
91                      //console.log( element );
92                      //console.log( "-->" + element
93                      if( element.obj != this ) {
94                          element.fn( key, param
95                      }
96                  }
97              }
98          }
99      }
100
101 /**
102  * イベントを起こす関数
103  * @param { string } keyword - キーワード. |で区切る

```

ここで複数キーワードを与える

```
104         * @param { hash } params - パラメータ（ハッシュで与  
える）  
105     */  
106     emit( keyword, params ) {  
107         var keys = keyword.split( "|" );  
108         this.emitByArray( keys, params );  
109     }  
110 }  
111  
112 //export default nylon;  
113 //exports = module.exports = nylon;  
114  
115 /**  
116 * nylonに登録する関数のクラス  
117 */  
118 class nylonfunc {  
119  
120     /**  
121     * コンストラクタ  
122     * @param { function } func - コールバック関数  
123     * @param { nylon } object - nylonオブジェクト  
124     */  
125     constructor( func, object ) {  
126         /** @type { function } */  
127         this.fn = func;  
128         /** @type { nylon } */  
129         this.obj = object;  
130     }  
131 }  
132 //export default nylonfunc;
```

## page.js

```
1 const pages = ["#classes", "#new_report"] ;/*ここで移動する変  
数の宣言*/  
2  
3  
4  
5  
6 /* ボタンを押されたときの処理*/  
7 class Page {  
8     /**  
9      * 描述  
10     * @date 2019-12-13  
11     * @param {Array} elms    ページ切り替えを行うIDの配列      ex: ["#classes"  
12     * @param {string} cls    切り替えを行うページのクラス      ex: '.page'  
13     * @returns {any}  
14     */  
15     constructor(elms, cls) {  
16         this.elms = elms;  
17         this.cls = cls;  
18  
19         let nl = new nylon();  
20         nl.on('page', (key, value) => {  
21             this.change(value.page);  
22         });  
23     }  
24     /*change を宣言 => ボタンなどを押された後の処理 */  
25     /**  
26      * 描述  
27      * @date 2019-12-13  
28      * @param {string} page  切り替えるID  ex: '#classes'  
29      * @returns {any}  
30      */  
31     change( page ) {  
32         if (this.elms.includes(page)) {  
33             let pages1 = document.querySelectorAll(this.cls); /*
```

```
クラスを指定する (.page はクラス名) */
34     for (p of pages1) {
35         p.style.display = "none";
36     }
37     console.log(page);
38     document.querySelector(page).style.display = "block";
39 } else {
40     throw "ページがありません";
41 }
42 }
43 }
```

## serviceworker.js

```
1 var urlsToCache = [
2     '/',
3     '/index.html',
4     '/spa.css',
5     '/spa.js'
6 ];
7
8 // インストール処理
9 self.addEventListener('install', function(event) {
10     event.waitUntil(
11         caches
12             .open(CACHE_NAME)
13             .then(function(cache) {
14                 return cache.addAll(urlsToCache.map(url => new Request
//ここを追記
15                     url))
16             })
17     );
18
19 // リソースフェッチ時のキャッシュロード処理
20 self.addEventListener('fetch', function(event) {
21     event.respondWith(
22         caches
23             .match(event.request)
24             .then(function(response) {
25                 return response ? response : fetch(event.request);
26             })
27         );
28     });

```

## manifest.json

```
1  {
2      "name": "あうとらいん",
3      "short_name": "あうとらいん",
4      "theme_color": "#2196f3",
5      "background_color": "#2196f3",
6      "display": "standalone",
7      "scope": "/",
8      "start_url": "/index.html",
9      "icons": [
10         {
11             "src": "janken192.png",
12             "sizes": "192x192",
13             "type": "image/png"
14         },
15         {
16             "src": "janken512.png",
17             "sizes": "512x512",
18             "type": "image/png"
19         }
20     ]
21 }
```