2コマでできる！

「アプリ開発」教材

群馬県立
高崎高等学校

×

System Creators
株式会社システムクリエイターズ

共同開発

高崎高校では株式会社アシアルの「Monaca Education」を導入しています（2025年度現在）。
アプリ開発のための参考資料として、高崎高校オリジナルで「Monaca Education」ポータルサイトも公開しています。内容は適宜修正されますが、こちらも活用してください。

＜MonacaEducationポータルサイト（高崎高校オリジナル）＞

https://sites.google.com/edu-g.gsn.ed.jp/ssh-monaca-education/%E3%83%9B%E3%83%BC%E3%83%A0

授業の参考テキスト

HTML基礎
P.20～P.33

CSS基礎
P36～P47

JavaScript基礎
P50～P60

今日の講義は上記までを終わっていることを
前提にしています。

演習の目的

WEBアプリケーションの画面製作を通して
アプリケーションの基本構造と構築方法を学ぶ

発展

自分で作りたいアプリの画面イメージを立案できる為の
素地を身に付ける。

課題テーマ

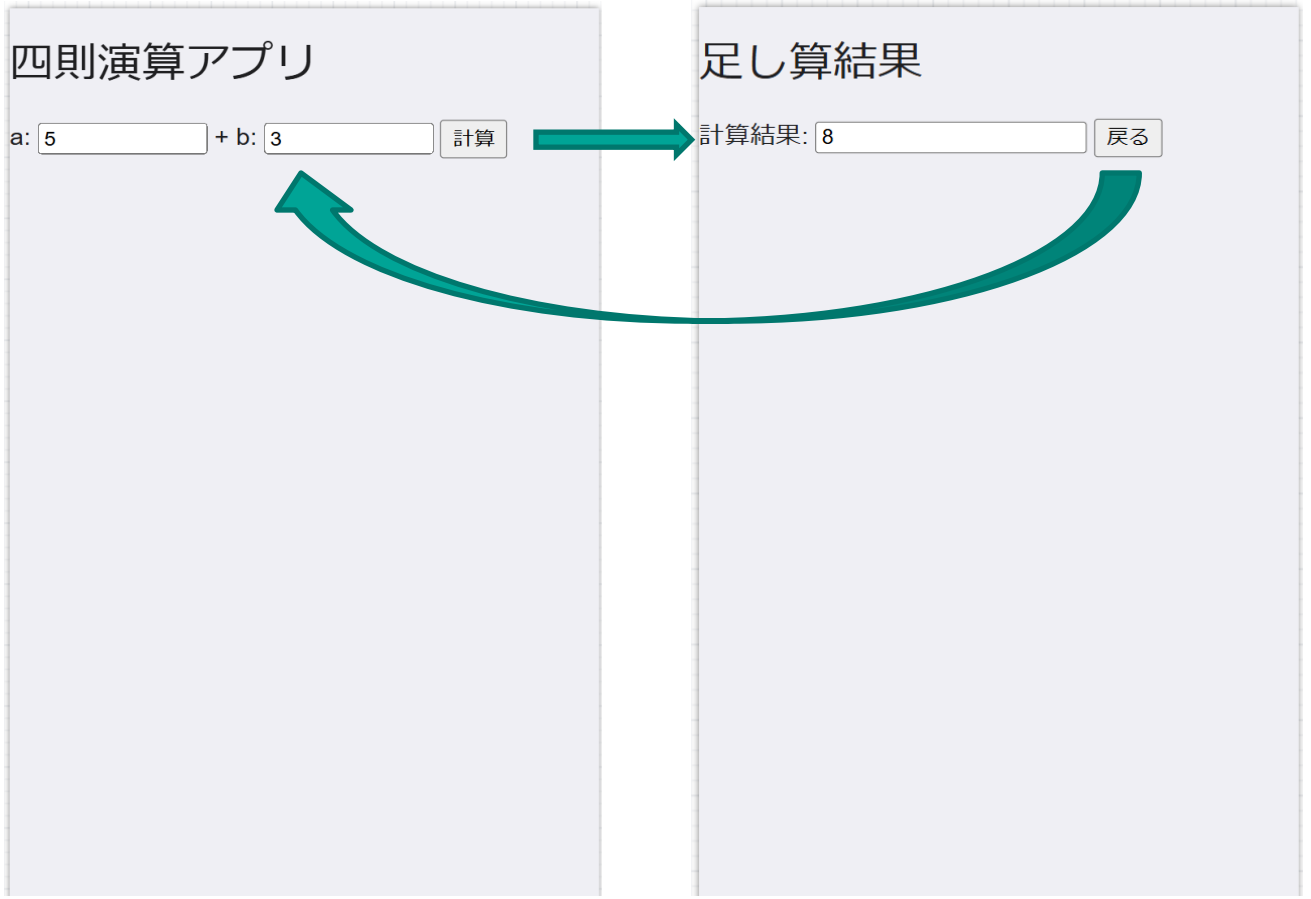
次の課題を自分のペースで自由に取り組んでください。
課題は2つ用意しましたが、自分でできる所まででOKです。

プログラミングに自信がある人
　スライドの解説編をみないで、インターネットで調べたりしながら自分の力で挑戦してみてください。

プログラミングに自信のない人
　スライドの解説を見ながら１ステップずつ取り組んでいきましょう！

作成課題 1

「次の画面のようなアプリを作ってください。
aとｂの入力ボックスを作り、計算ボタンを押すとページが切り替わりa+bの値が表示されます。」

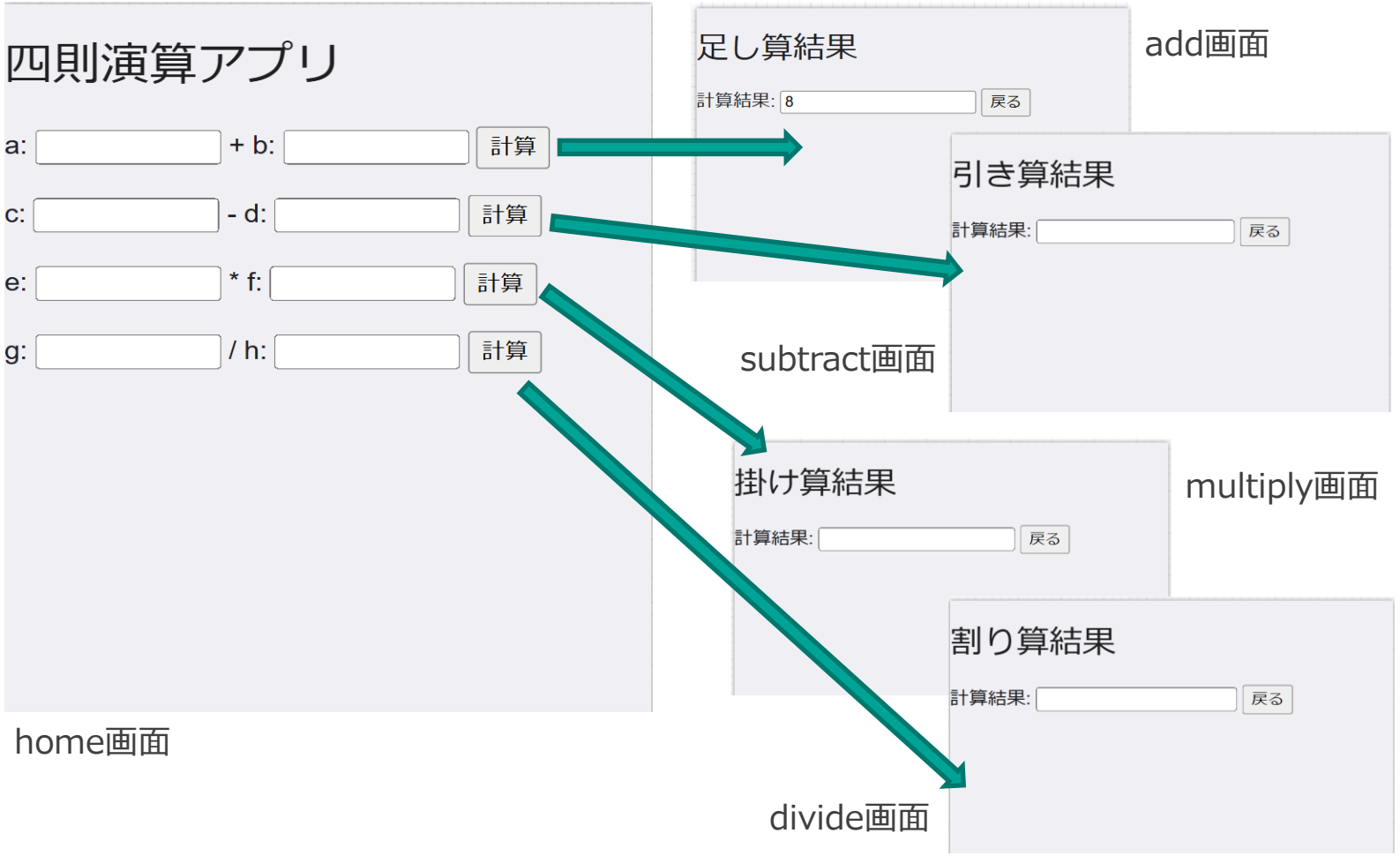


home画面　　add画面

動作確認
プレイ用

課題２

「次の画面のようなアプリを作ってください。
先ほどのアプリを四則演算ができるように拡張してください。」



動作確認
プレイ用

課題 1

解説編

ワークショップ

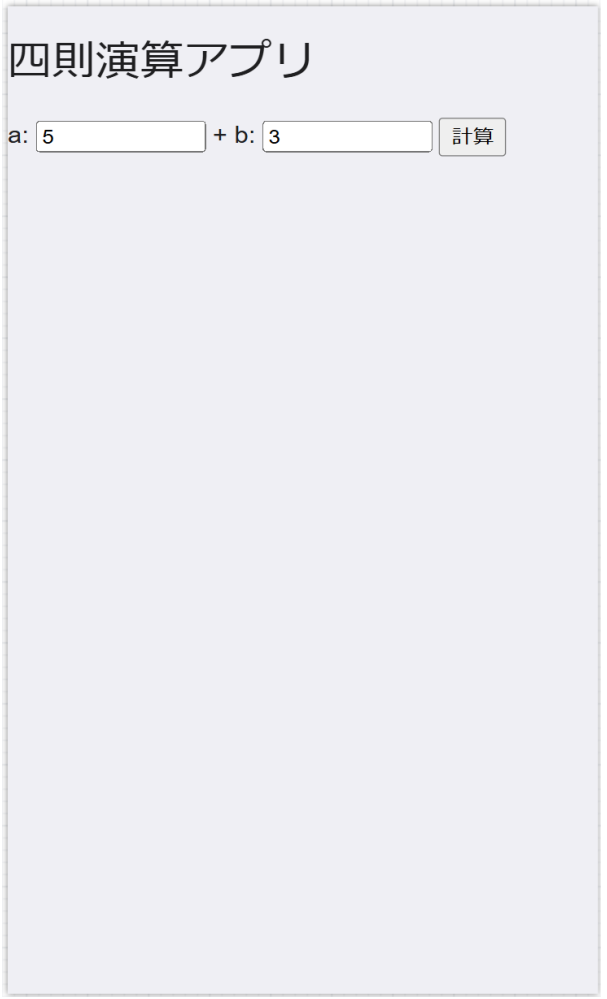
新しいプロジェクトを作ってください。

どっち
でもOK

https://onl.tw/Lk5K68T

テンプレートをインポートしてください。

画面を作ってみよう。



左のようなWEBアプリの画面を作ろうと思います。
各自で4つの手順に従って画面を完成させてみてください。

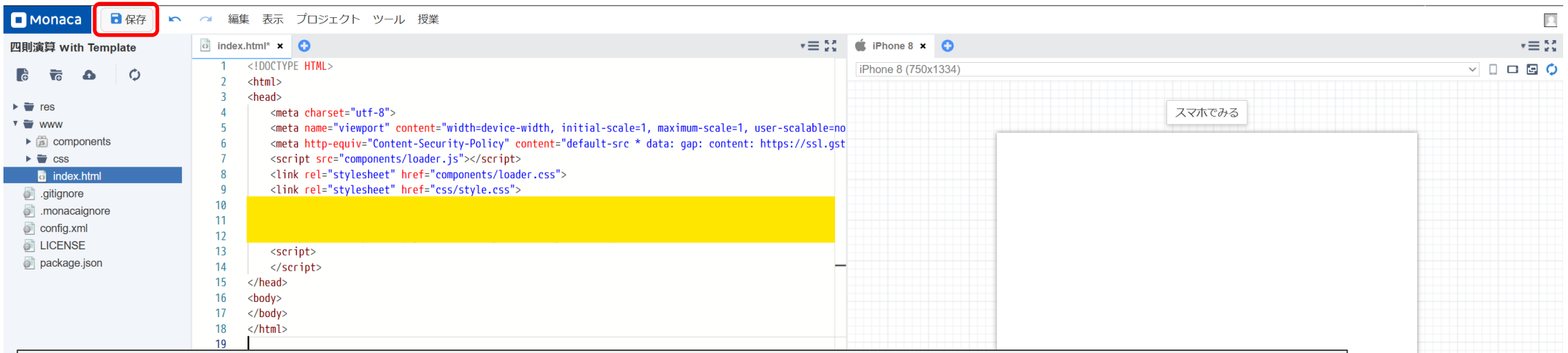
①入力項目の作り方

②ボタン項目の作り方

③ボタン項目にイベントを付与する方法

④Add ページの作り方

Onsen UI を入れよう。



<head> ～ </head>で囲まれている範囲内（黄色背景）に以下を記述します。
アプリで Onsen UI を使うための設定です。

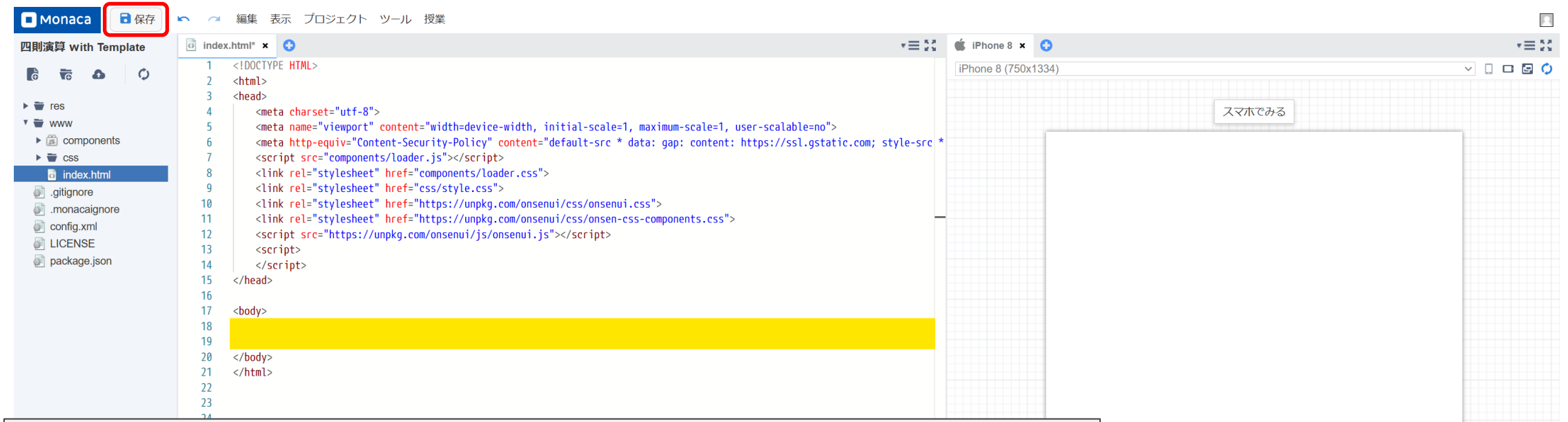
<link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsenui.css">
<link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsen-css-components.css">
<script src="https://unpkg.com/onsenui/js/onsenui.js"></script>

記述したら「保存ボタン」をおしましょう。

まめ知識

Onsen UI とは
モバイルアプリ開発に特化したオープンソースのUI**フレームワーク**
今回は、画面移動などを簡単に作るために使用します。

Onsenナビゲーターを用意しよう。



<body> ～ </body>で囲まれている範囲内（黄色背景）に以下を記述します。

<ons-navigator id="myNavigator" page="home"></ons-navigator>

記述したら「保存ボタン」をおしましょう。

まめ知識

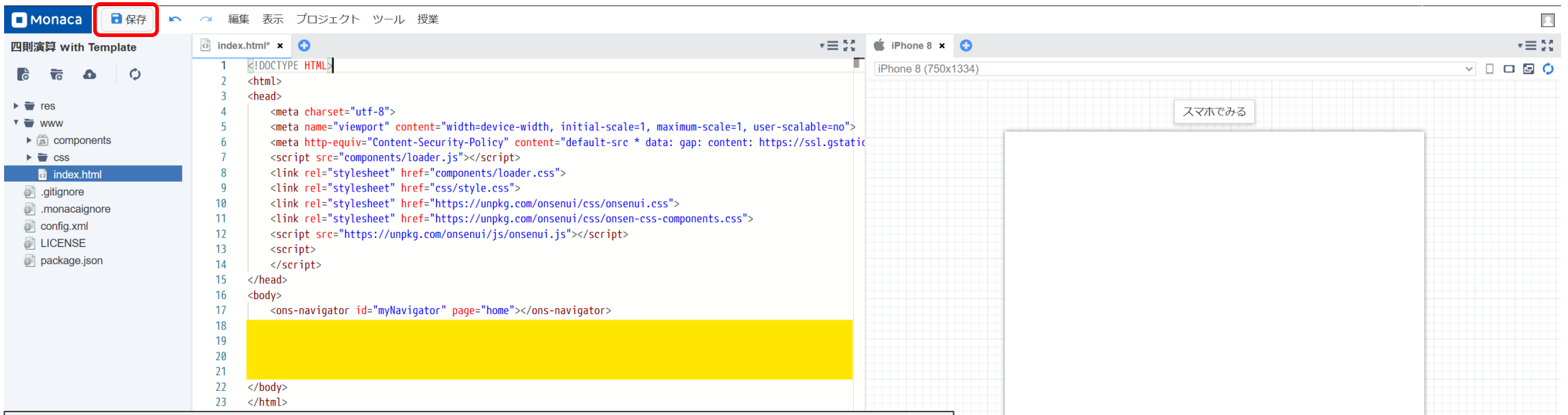
<ons-navigator> とは

Onsen UI に含まれるコンポーネントの1つ。
これを使うと簡単に画面移動を行うことができます。

<ons-navigator id="myNavigator" page="home"></ons-navigator>

page="○○" で指定した画面がアプリの最初に表示されます。
次のページから home 画面を作っていきます。

ホーム画面を作ろう。

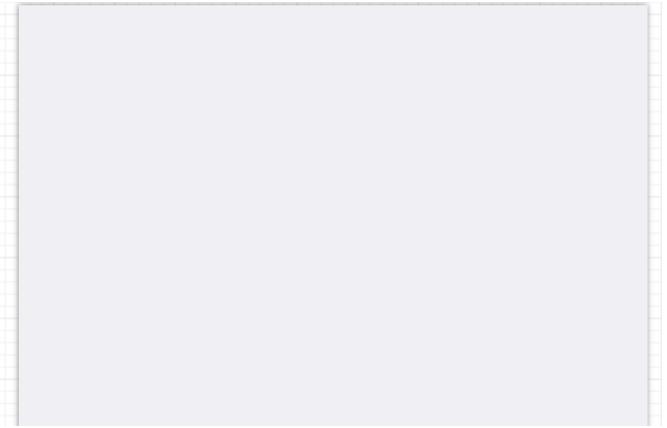


<body> ～ </body>で囲まれている範囲内（黄色背景）に
ホーム画面に当たるテンプレートタグを用意します。

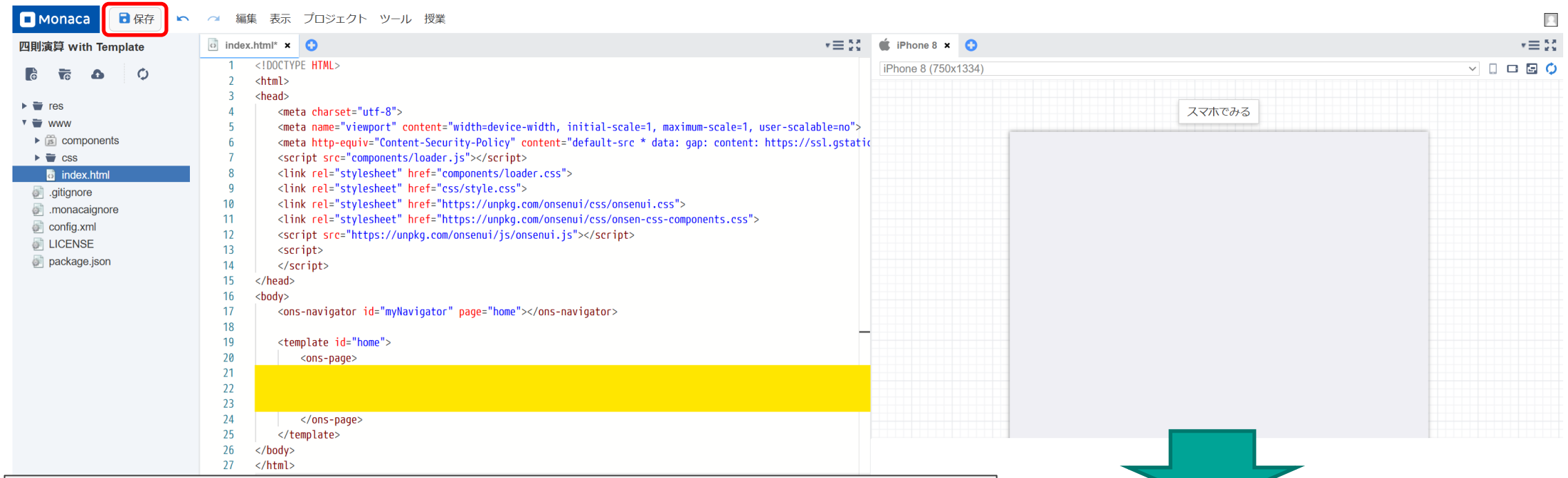
```
<template id="home">
    <ons-page>
    </ons-page>
</template>
```

記述したら「保存ボタン」をおしましょう。
画面がグレーにならない場合、これまでの記述のどこかに
間違いがあります。よく確認しましょう。

画面がグレーになります。



画面項目を作ってみよう。



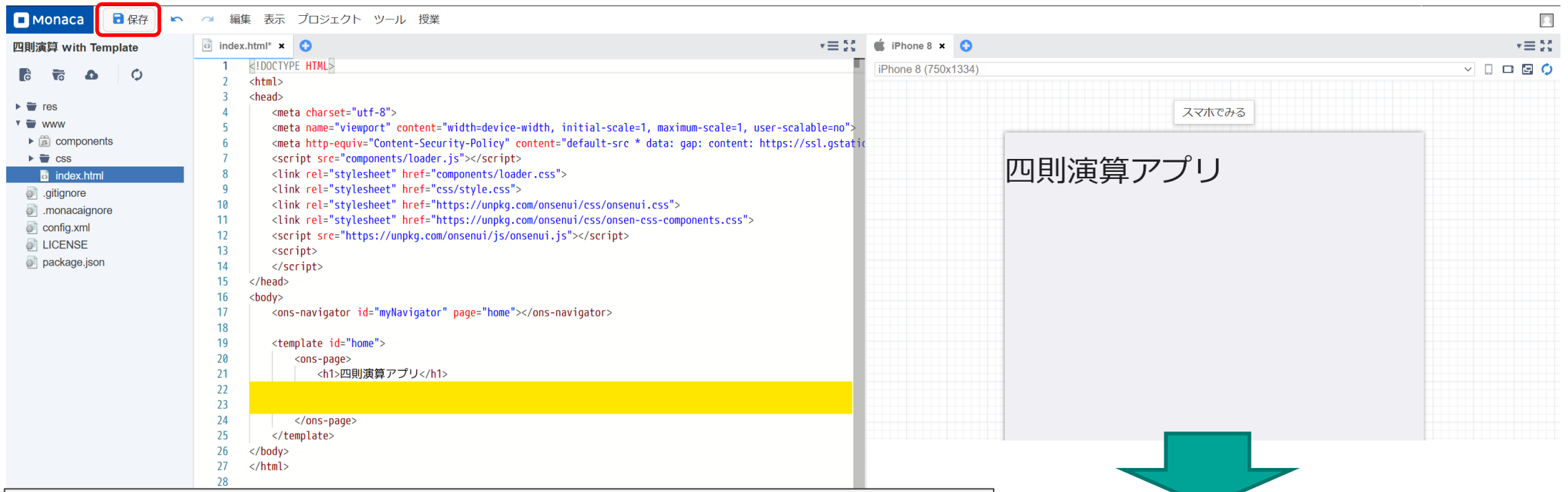
ホーム画面の項目を作るときは、
<template id="home"> 内の **<ons-page> ～ </ons-page>** で
囲まれている範囲内(黄色背景)に画面をつくるプログラムは記述します。

```
<h1>四則演算アプリ</h1>
```

記述したら「保存ボタン」をおしましょう。
保存を押す事で作ったプログラムが右側の画面に反映されます。

四則演算アプリ

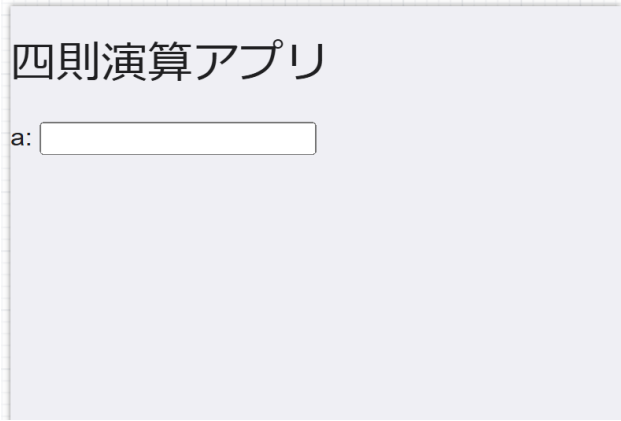
画面項目を作ってみよう。



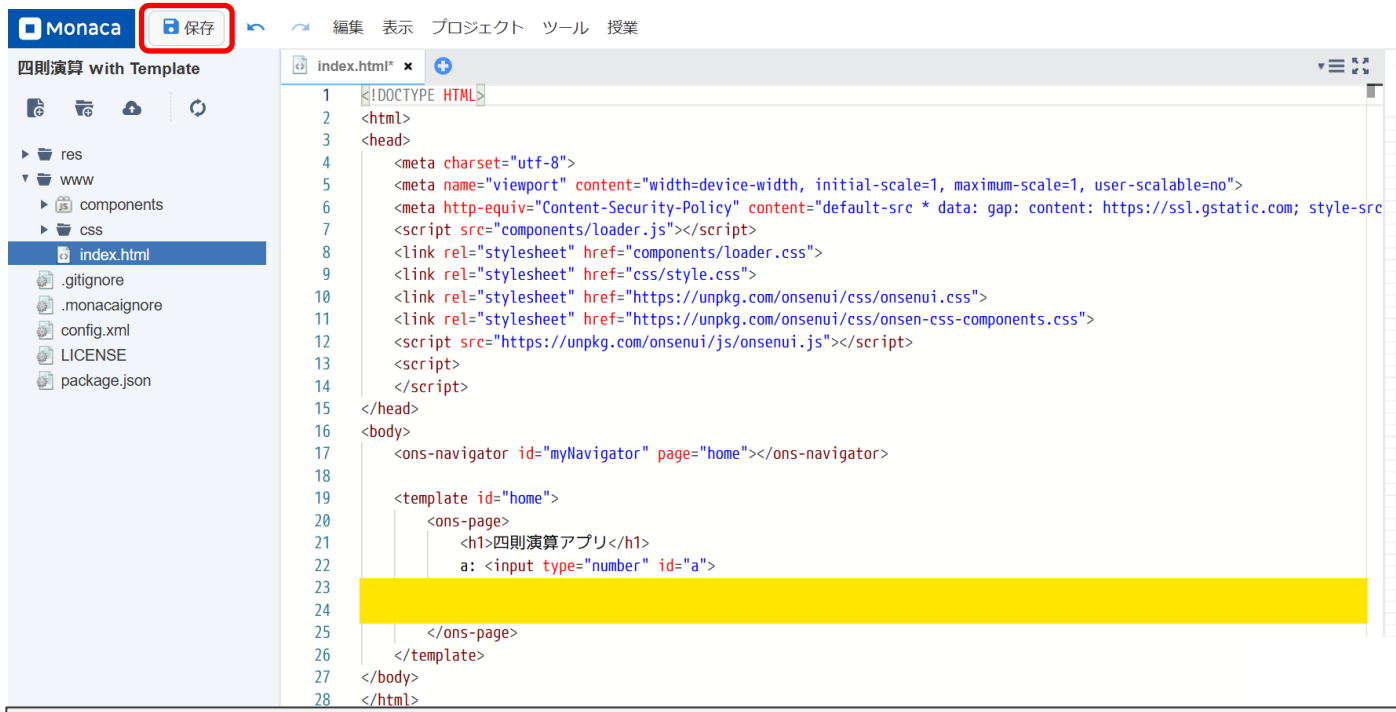
画面にタイトルが表示されましたでしょうか。
　次は入力項目を画面につくるプログラムを記述します。

　a: <input type="number" id ="a">

記述したら「保存ボタン」をおしましょう。
保存を押す事で作ったプログラムが右側の画面に反映されます。



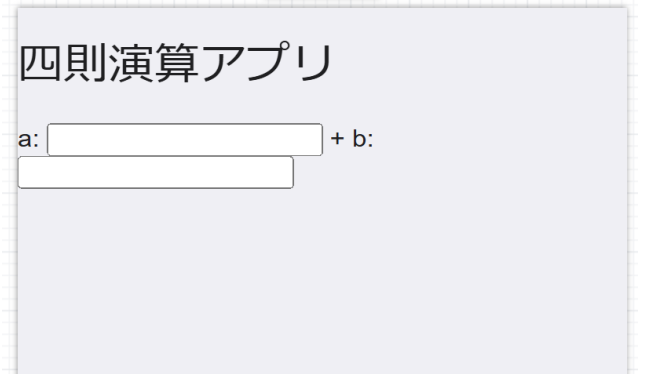
画面項目を作ってみよう。



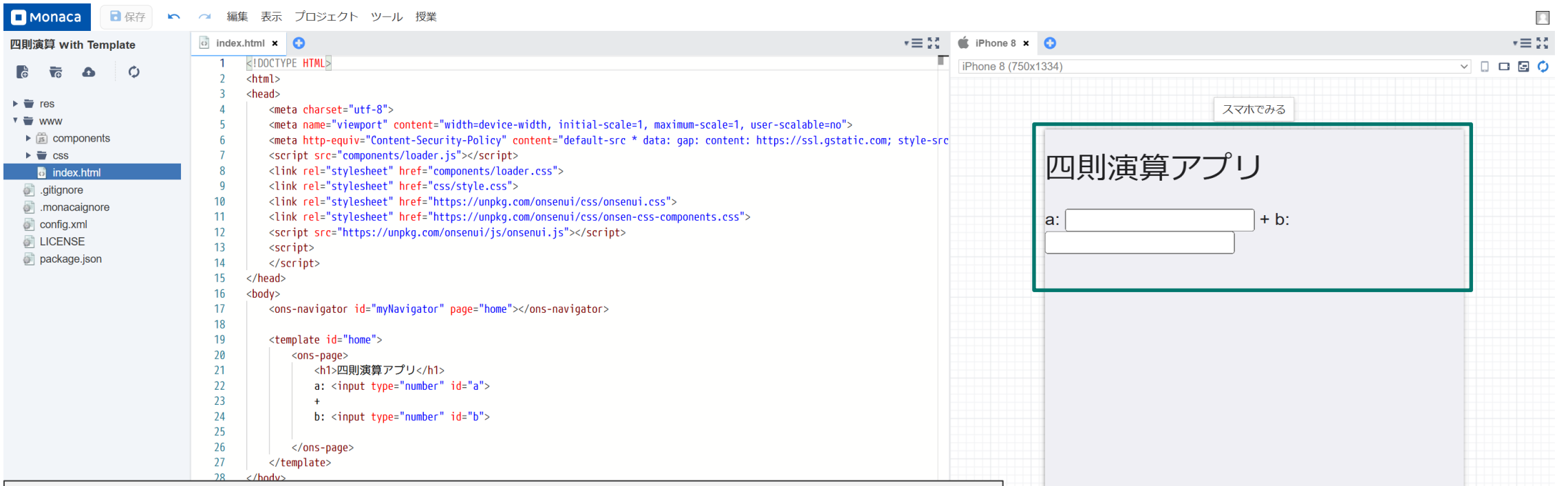
<ons-page> ～ </ons-page>で囲まれている範囲内（ 黄色背景）に
同じように画面をつくるプログラムを記述します。
“＋”とＢ項目を追加しましょう。

+b: <input type="number" id ="b">

記述したら「保存ボタン」をおしましょう。
保存を押す事で作ったプログラムが右側の画面に反映されます。



画面項目を作ってみよう。



右側の画面のように表示されたでしょうか。
これだと表示が不格好なので次のページの解説を
読みながら入力項目の長さを調整してみましょう。

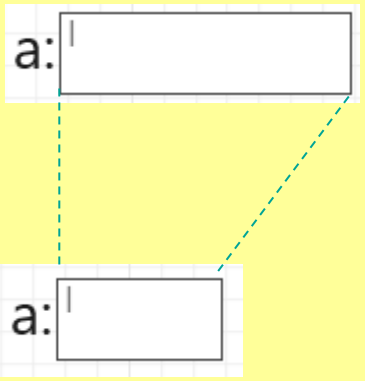
画面項目を作ってみよう。

入力項目の作り方

覚えて！！

<input type="number">　　<input type="text">

画面イメージ



幅を指定したい場合

HTMLでの書き方

a: <input type="number"　id ="a">

入力項目を数字にする。という意味

IDはプログラムで管理する名前のような物で同じ名前は使えません。

a: <input type="number" id="a" style="width: 100px;">

幅を100px（ピクセル）にする。

↑数字を変えて幅が変わることを試してみてください

まめ知識

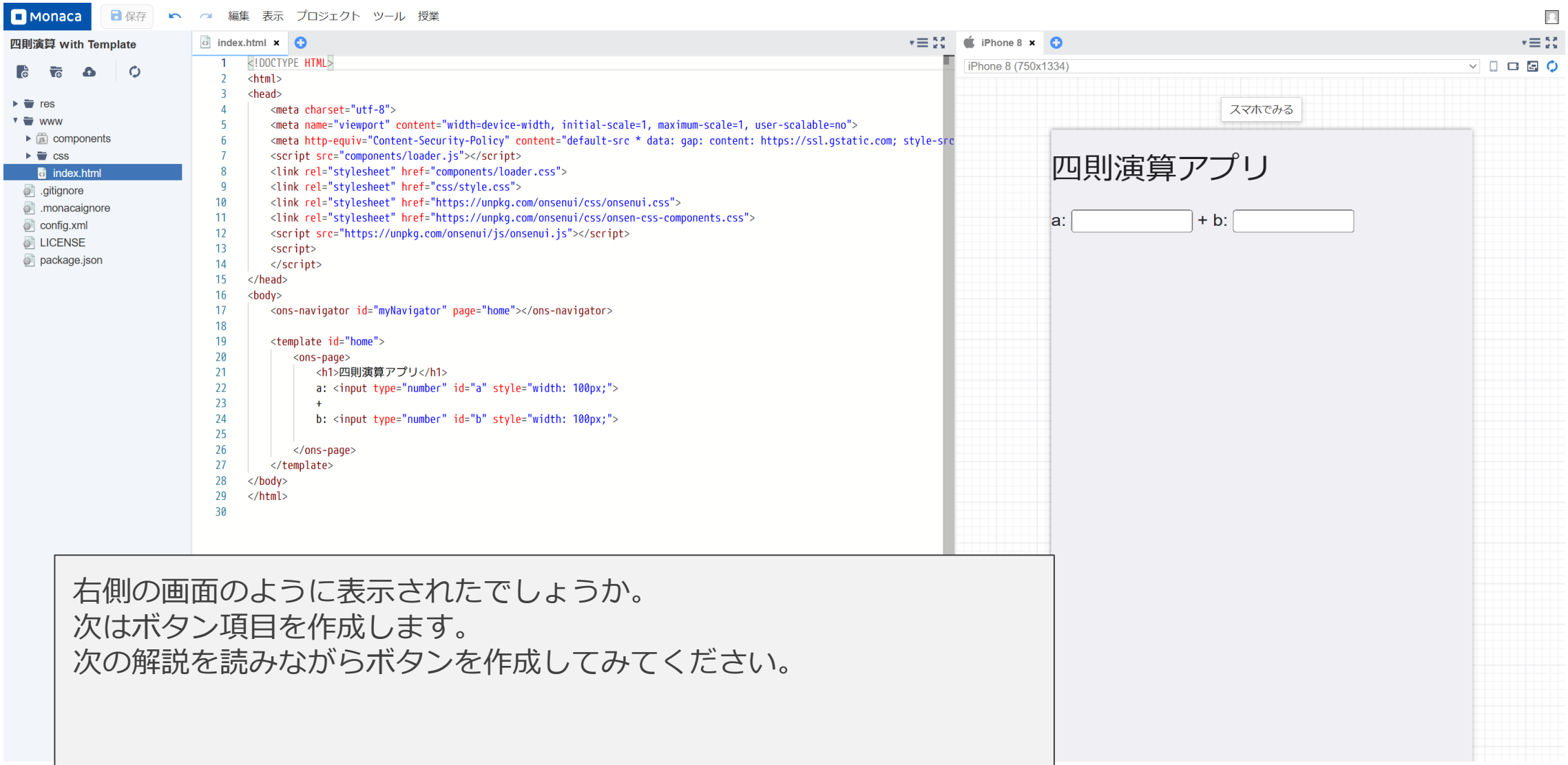
プログラムの　=（イコール）について

プログラムには＝がたくさん出てきますが、一般的な数学の等号の意味ではなく、
左辺に右辺を代入するという意味で使われます。
A = 1;などと記述されているプログラムはAに 1 を代入するという命令です。

px（ピクセル）とは

HTML、CSSの基本単位です。 ピクセルは、パソコンやスマホのモニタ画面を構成している粒のことです。 ピクセルは、cmやmmといった絶対値と違い、モニタの解像度の変更によって表示サイズが変化します。

参考・・・ここまで



右側の画面のように表示されたでしょうか。
次はボタン項目を作成します。
次の解説を読みながらボタンを作成してみてください。

画面項目を作ってみよう。

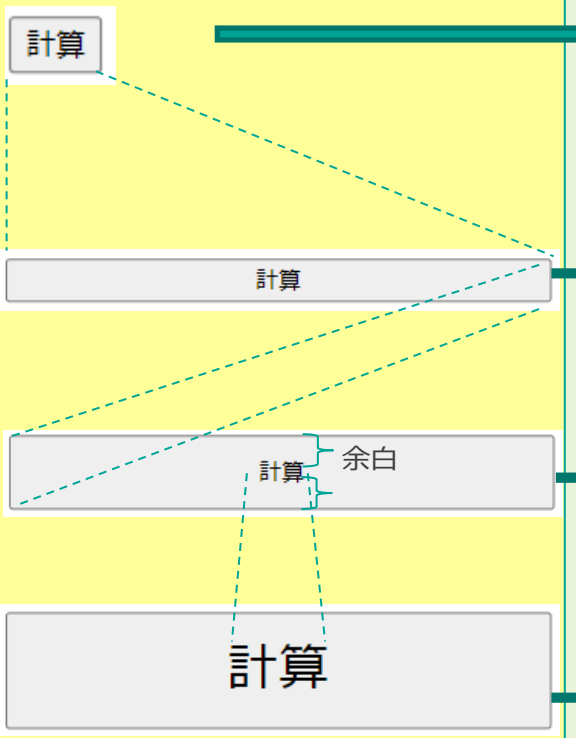
ボタン項目の作り方

覚えて！！

<input type="button" value="表示したい文字">

画面イメージ

HTMLでの書き方



<input type="button" value="計算">

入力項目をボタンにする。名前を計算と表示する意味

<input type="button" style="width:90%; " value="計算">

幅を画面全体の90%にする。

<input type="button" style="width: 90%; padding: 10px; " value="計算" />

要素内側の余白 を10pxにする。

<input type="button" style="width:90%;padding:10px;font-size:30px; " value="計算" />

文字の大きさを30pxにする。

まめ知識

ボタンの作り方は<input type=“button”>と<button></button>で２種類あります。

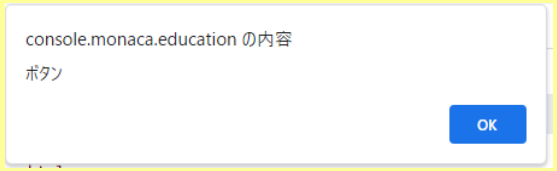
今回はわかりやすさを重視してInput type = “button”で記述していますが、ボタンの作り方としては
<button type=“button” >計算</button>という書き方もできます。<button>タグで記述した場合はボタンの中に画像を貼る事ができますので、ボタンにアイコンなどを付けたい場合には <button type=“button”><img src=“画像パス” >計算</button>と記述します。

イベントを作ってみよう。

ボタン項目にイベントを付与する方法　覚えて！！　<input type="button" value="計算"onclick="実行する関数名">

画面イメージ

ボタンを押したらメッセージが表示。



HTMLでの書き方

<input type="button" value="計算">
　　計算ボタンを設置します。

<input type="button" value="計算"onclick=alert("ボタン")>

どちらの書き方でもOK

<button onclick=alert("ボタン")>計算</button>

クリックされたら メッセージアラートを実行。

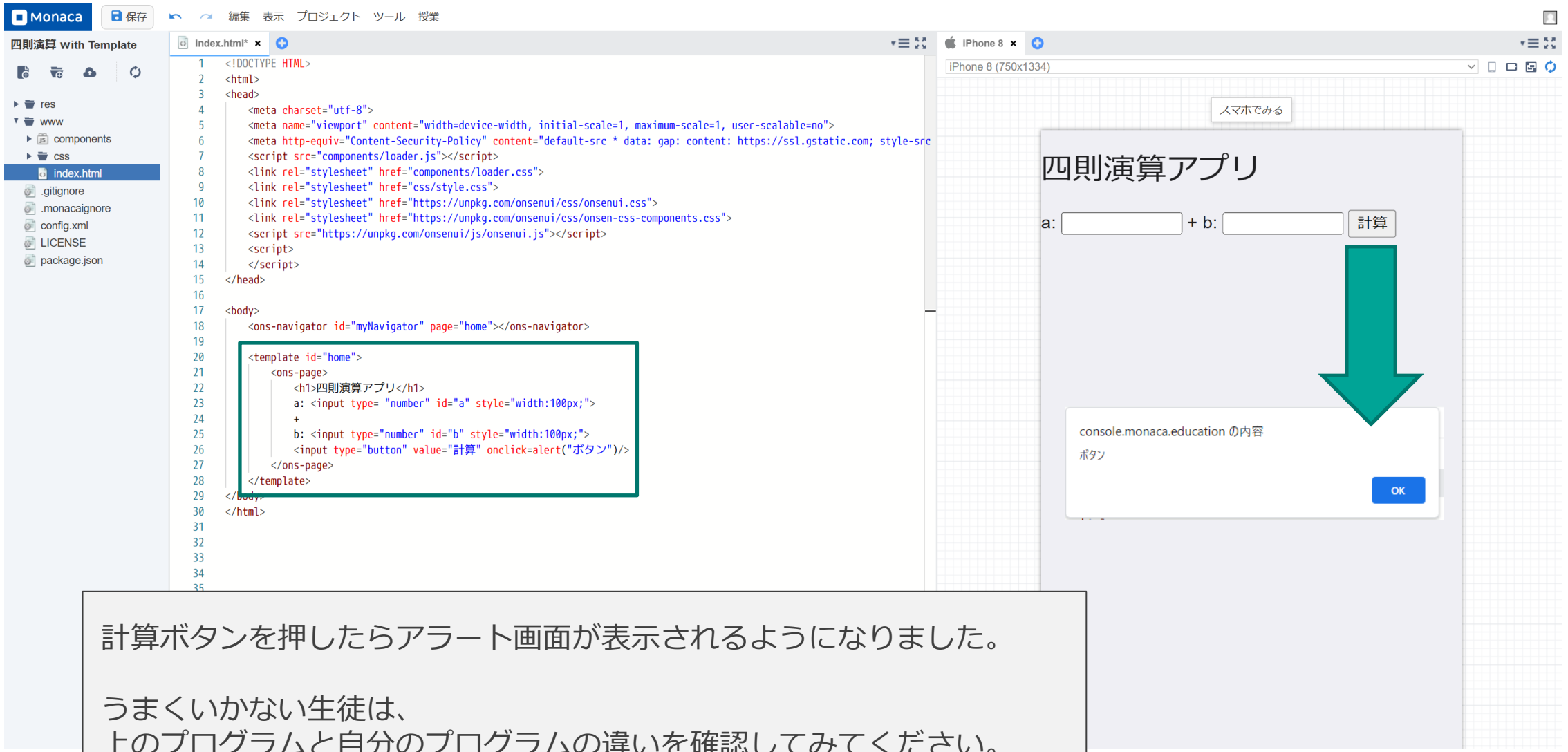
クリックされたら メッセージアラートを実行。

まめ知識

イベントは画面や画面項目毎に付与できます。

今回はボタン項目にボタンがクリックされた時（onclick=""）のイベントの作り方を説明しました。
それ以外のイベントではダブルクリックされた時（ondblclick=""）、入力欄のテキストが選択された時（onselect=""）、
フォーカスを失う際に値が変化していた時（onchange=""）などがあります。
それ以外にも非常にたくさんのイベントを作る事ができますので、気になった方は下記を参考にしてみてください。
　☛ https://www.tagindex.com/html_tag/basic/event.html

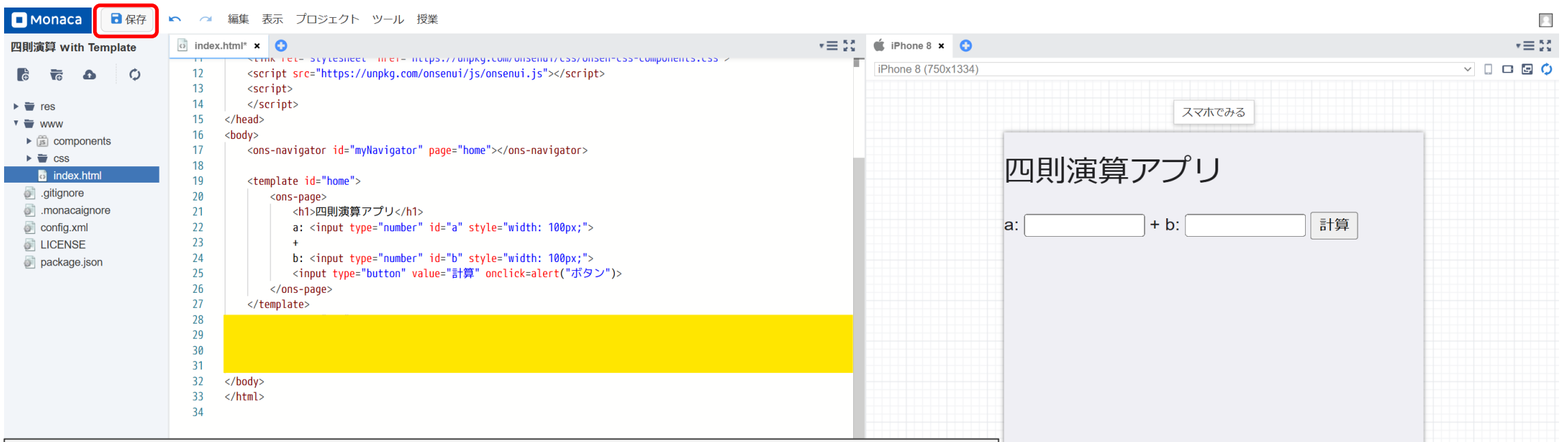
参考・・・ここまで



計算ボタンを押したらアラート画面が表示されるようになりました。

うまくいかない生徒は、
上のプログラムと自分のプログラムの違いを確認してみてください。

add画面を作ろう

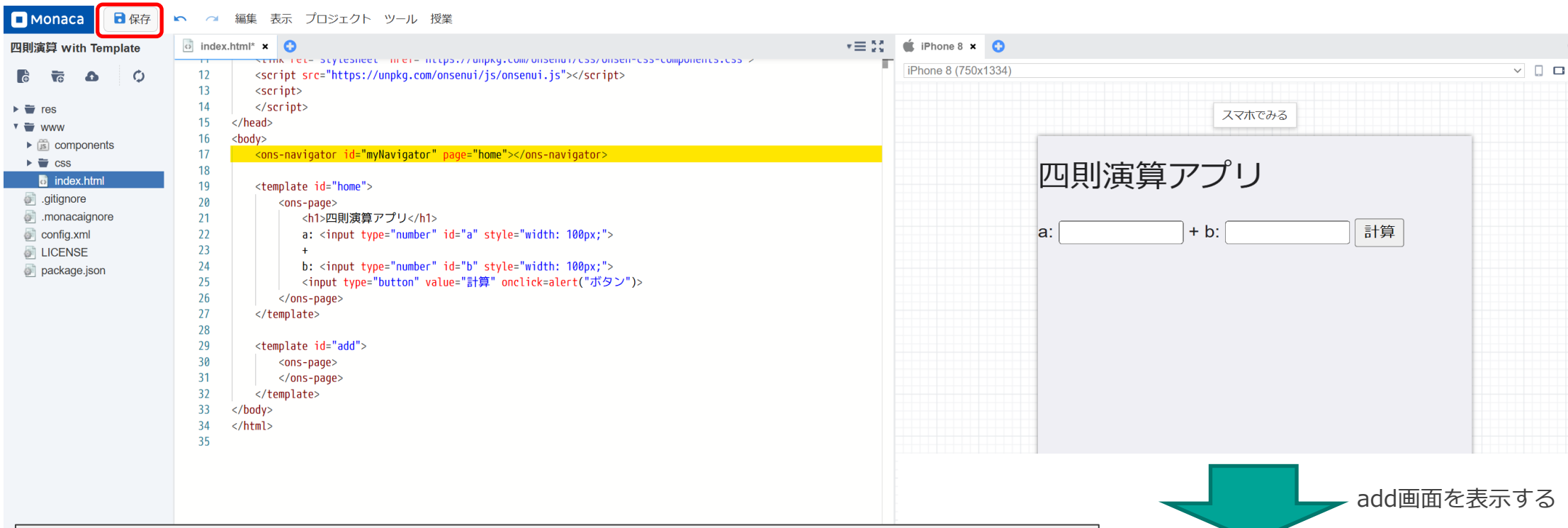


ホーム画面 (<template id="home"> ～ </template>) の下に
add画面にあたるテンプレートを用意します。

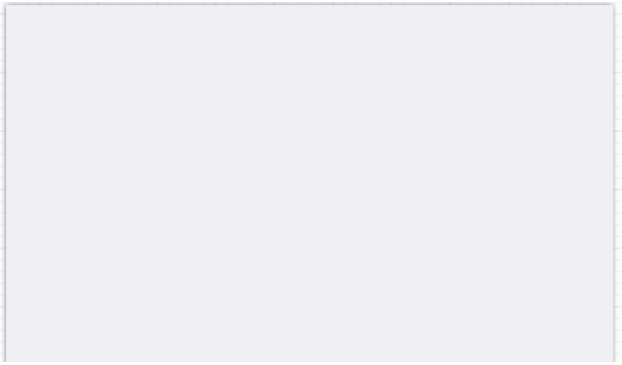
```
<template id="add">
    <ons-page>
    </ons-page>
</template>
```

記述したら「保存ボタン」をおしましょう。

add画面が表示されるようにしよう



add画面を表示する

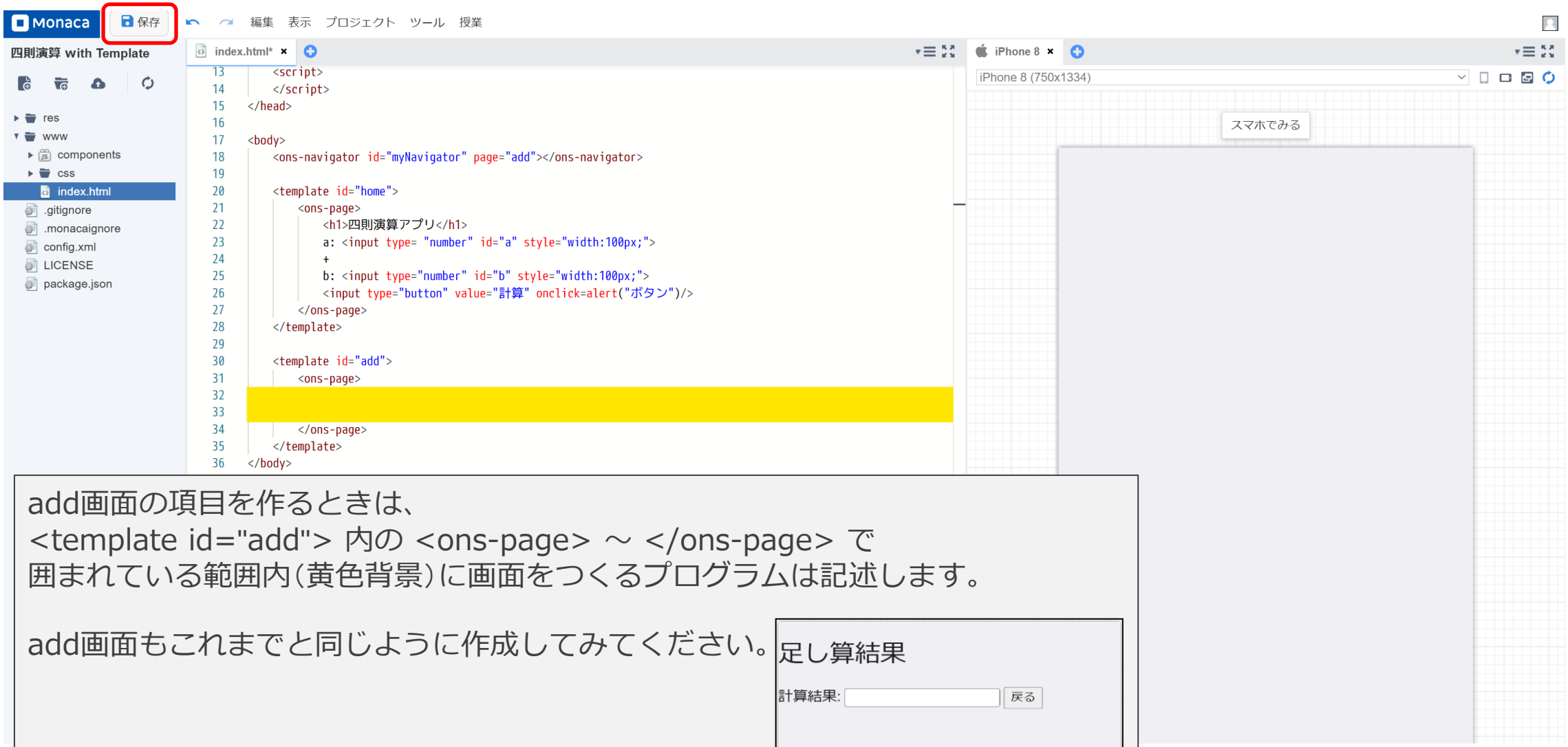


これから add画面を作っていくので一時的にadd画面が最初に表示されるようにしましょう。黄色い箇所を書き換えます。

<ons-navigator id="myNavigator" page="~~home~~"></ons-navigator>
<ons-navigator id="myNavigator" page="add"></ons-navigator>

記述したら「保存ボタン」をおしましょう。

add画面の画面項目を作ろう



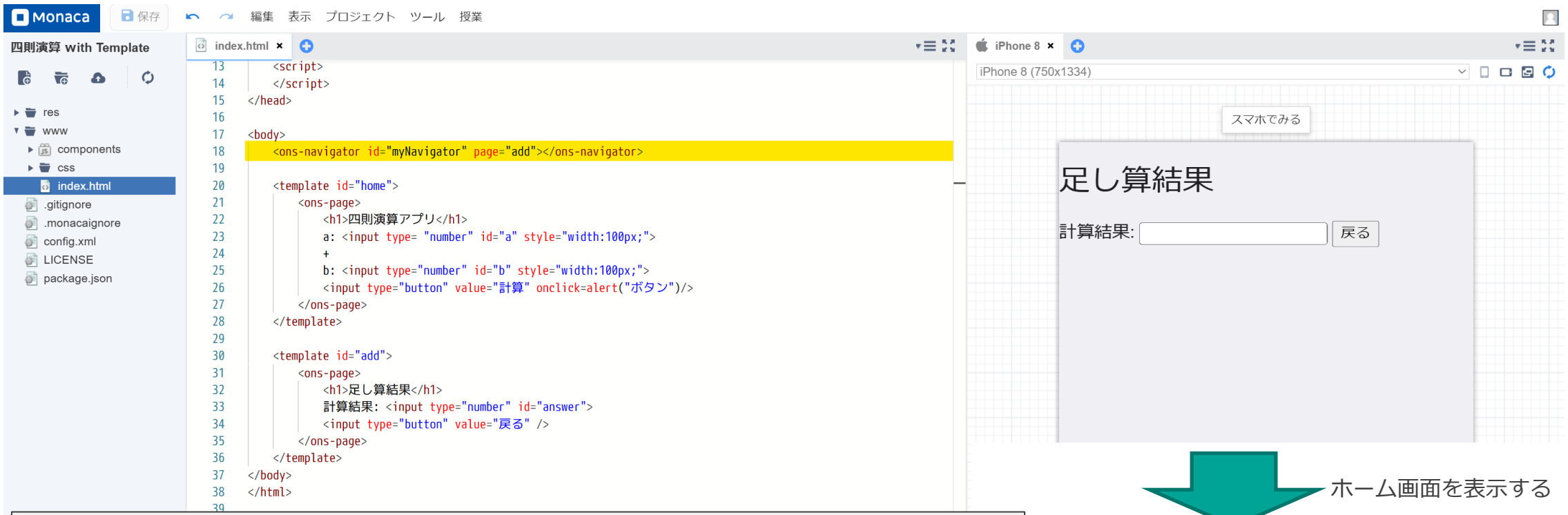
add画面の項目を作るときは、
<template id="add"> 内の <ons-page> ～ </ons-page> で
囲まれている範囲内(黄色背景)に画面をつくるプログラムは記述します。

add画面もこれまでと同じように作成してみてください。

add画面

記述したら「保存ボタン」をおしましょう。

参考・・・ここまで

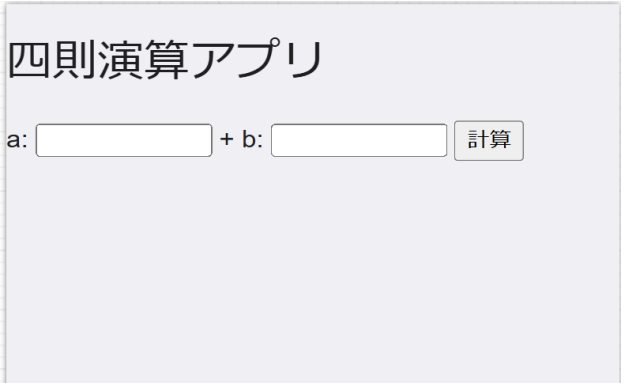


右側の画面のように作成できたでしょうか。

add画面が作れたら、最初にホーム画面が表示されるように
黄色箇所を下の記述のようにもとに戻しましょう。

<ons-navigator id="myNavigator" page="home"></ons-navigator>

ホーム画面を表示する

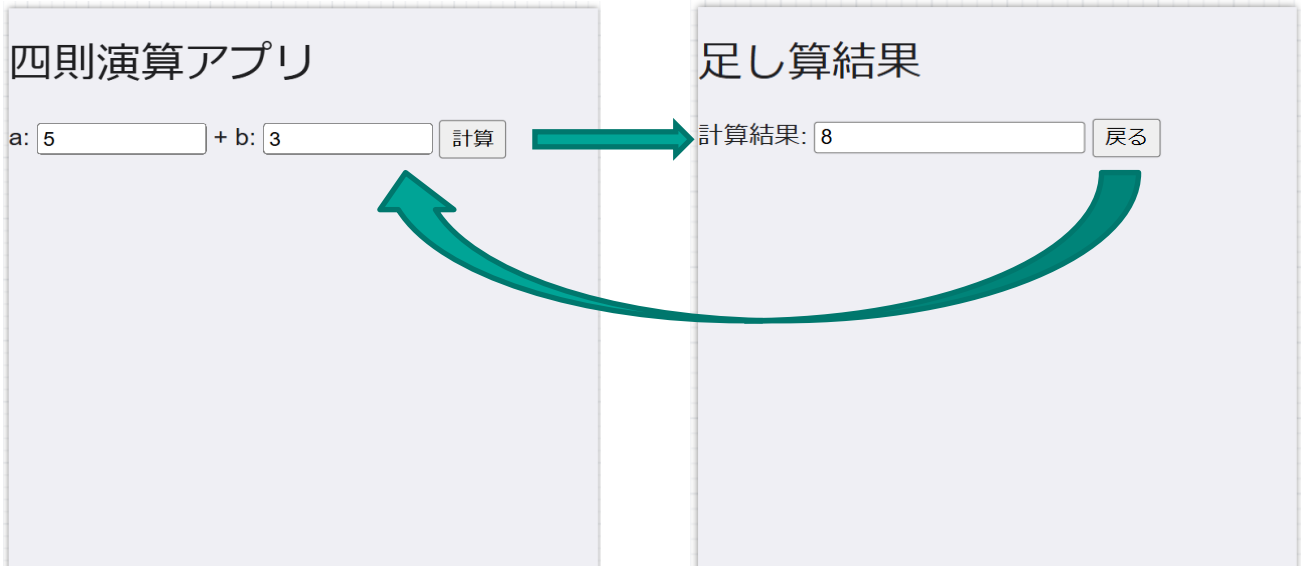


画面に動きを作ってみよう。

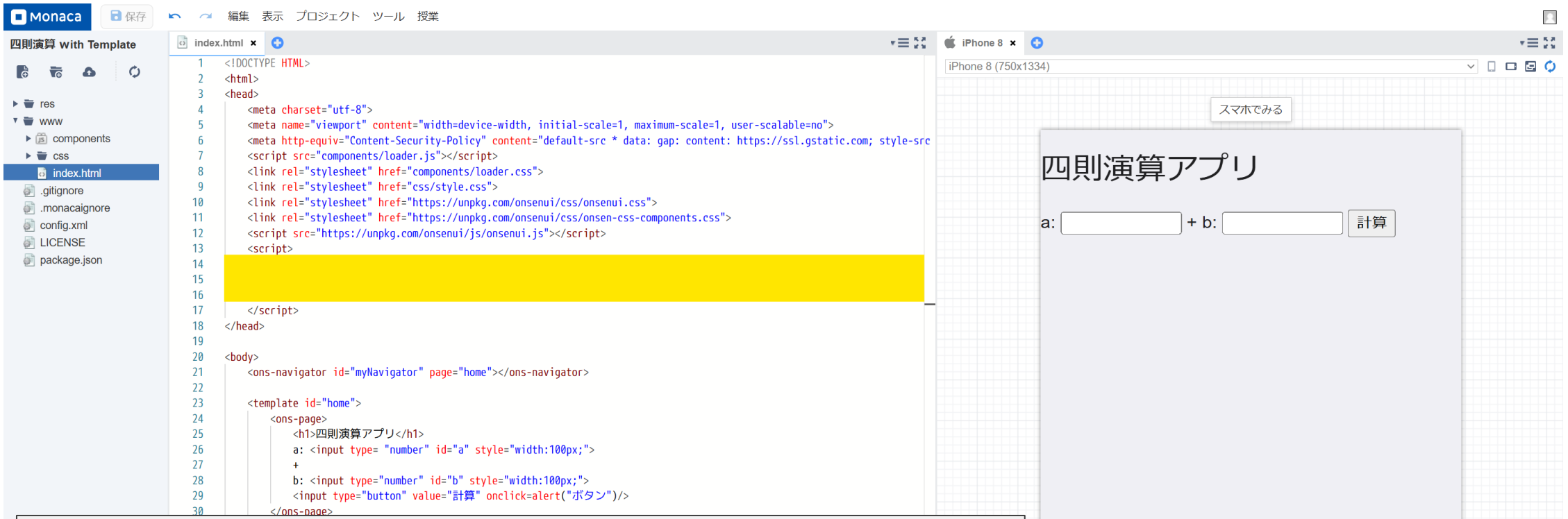
下のようなWEBアプリの画面にページ移動の動きと足し算をする動きを作ろうと思います。
各自で2つの手順に従って画面に動きを加えてみてください。

①ページ切り替え関数の作り方

②足し算関数の作り方



プログラムを記述しよう。



次はいよいよプログラムを記述します。

ボタンや画面に文字を表示したり、計算をさせたり、処理をさせるには JavaScriptで関数(function {})を作る必要があります。

JavaScriptの関数は**<script> </script>**タグ内（黄色枠内）に記載する必要があります。

次の解説を読みながらボタンを押した時のページ移動プログラムを作成してみてください。

プログラムを記述しよう。

ボタンや画面に文字を表示したり、計算をさせたり、処理をさせるにはプログラムを作る必要があります。
プログラムは**<script> </script>**タグ内に記載する必要があります。
まずは、home画面 から add画面 へのページ移動を作りましょう。みなさんも作成してみてください。

add画面

```
<template id="add">
    <ons-page>
        <h1>足し算結果</h1>
        計算結果: <input type="number" id="answer">
        <input type="button" value="戻る"/>
    </ons-page>
</template>
```

```
document.getElementById("myNavigator").pushPage('add')
```

document.getElementById(“取得する項目のID”) で
画面項目を取得できる
ページ移動するために <ons-navigator> を取得する

add画面に進む命令

```
<ons-navigator id="myNavigator" page="home"></ons-navigator>
```

```
10      <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsenui.css">
11      <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsen-css-components.css">
12      <script src="https://unpkg.com/onsenui/js/onsenui.js"></script>
13      <script>
14
15          <scriptタグ内に追加
16      </script>
17  </head>
```

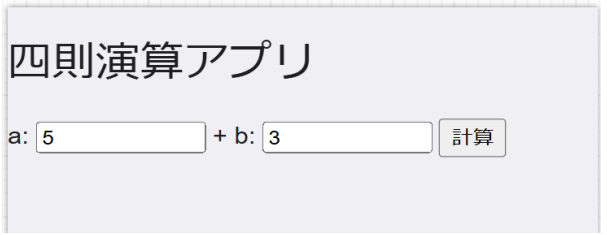
```
function navigate_to_add() {
    document.getElementById('myNavigator').pushPage('add');
}
```

イベントを直して完成

```
<input type= "button" value="計算" onclick=alert("ボタン") />
<input type= "button" value="計算" onclick=navigate_to_add() />
```

プログラムを記述しよう。

同じように add 画面から home 画面へ戻るプログラムを作ってみましょう。




```
document.getElementById("myNavigator").popPage()
```

1つ前のページに戻る命令

ページ移動するときは <ons-navigator> を取得する

```
<ons-navigator id="myNavigator" page="home"></ons-navigator>
```

```
13    <script>
14        function navigate_to_add() {
15            document.getElementById('myNavigator').pushPage('add');
16        }
17
18
19    </script>
```

先ほどの関数の下に追加

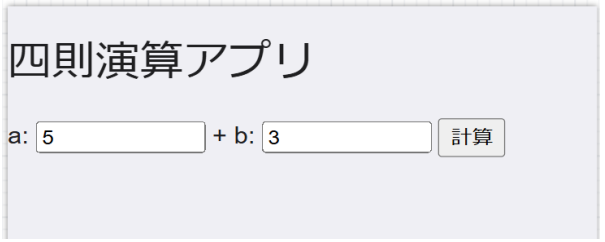
```
function back() {
    document.getElementById('myNavigator').popPage();
}
```

クリック時の
イベントに設定

```
<input type="button" value="戻る" />
<input type="button" value="戻る" onclick=back() />
```

プログラムを記述しよう。

足し算機能を作りましょう。add 画面で足し算するために、まずはホーム画面で入力した値を add 画面でも使えるようにしましょう。



ページ移動時に a, b の値を
add画面でも使えるように保存する

① <script> タグ内に変数を定義します。

```
<script>
    var num1, num2;
    function navigate_to_add() {
        document.getElementById('myNavigator').pushPage('add');
    }
    function back() {
        document.getElementById('myNavigator').popPage();
    }
</script>
```

<head> タグ内の **<script>** タグ内の変数には、
どの画面からでもアクセスができます。

② 変数に画面項目の値を保存します。

a: <input type="number" id ="a">

a:項目の入力値を取得するため、
a項目のIDを指定

num1 = document.getElementById("a").value

Idが”a”の項目の値を取得しろという命令です。
左の画面の場合は **5** という文字が取得できます。

```
function navigate_to_add() {
    document.getElementById('myNavigator').pushPage('add');
}
```

変更

```
var num1, num2;
function navigate_to_add() {
    num1 = document.getElementById("a").value;
    num2 = document.getElementById("b").value;
    document.getElementById('myNavigator').pushPage('add');
}
```

プログラムを記述しよう。

add画面が表示されたときに計算結果を表示するプログラムを作成してみましょう。

```
26  <body>
27      <ons-navigator id="myNavigator" page="home"></ons-navigator>
28      <template id="home">
29          <ons-page>
30              <h1>四則演算アプリ</h1>
31              a: <input type= "number" id="a" style="width:100px;">
32              +
33              b: <input type="number" id="b" style="width:100px;">
34              <input type="button" value="計算" onclick="navigate_to_add()"/>
35          </ons-page>
36      </template>
37
38      <template id="add">
39          <ons-page>
40              <h1>足し算結果</h1>
41              計算結果: <input type="number" id="answer">
42              <input type="button" value="戻る" onclick="back()"/>
43              add 画面の <ons-page> タグ内に書く
44
45          </ons-page>
46      </template>
47  </body>
```

ページを読み込んだ時に行う処理：

```
ons.getScriptPage().onInit = function() {}
```

ページを読み込んだときに行う処理を設定する命令
<ons-page> ～ </ons-page> タグ内でしか使用できません。

```
<script>
    ons.getScriptPage().onInit = function() {
        answer = parseInt(num1) + parseInt(num2);
        document.getElementById('answer').value = answer;
    }
</script>
```

文字を数値に変換するという命令です。

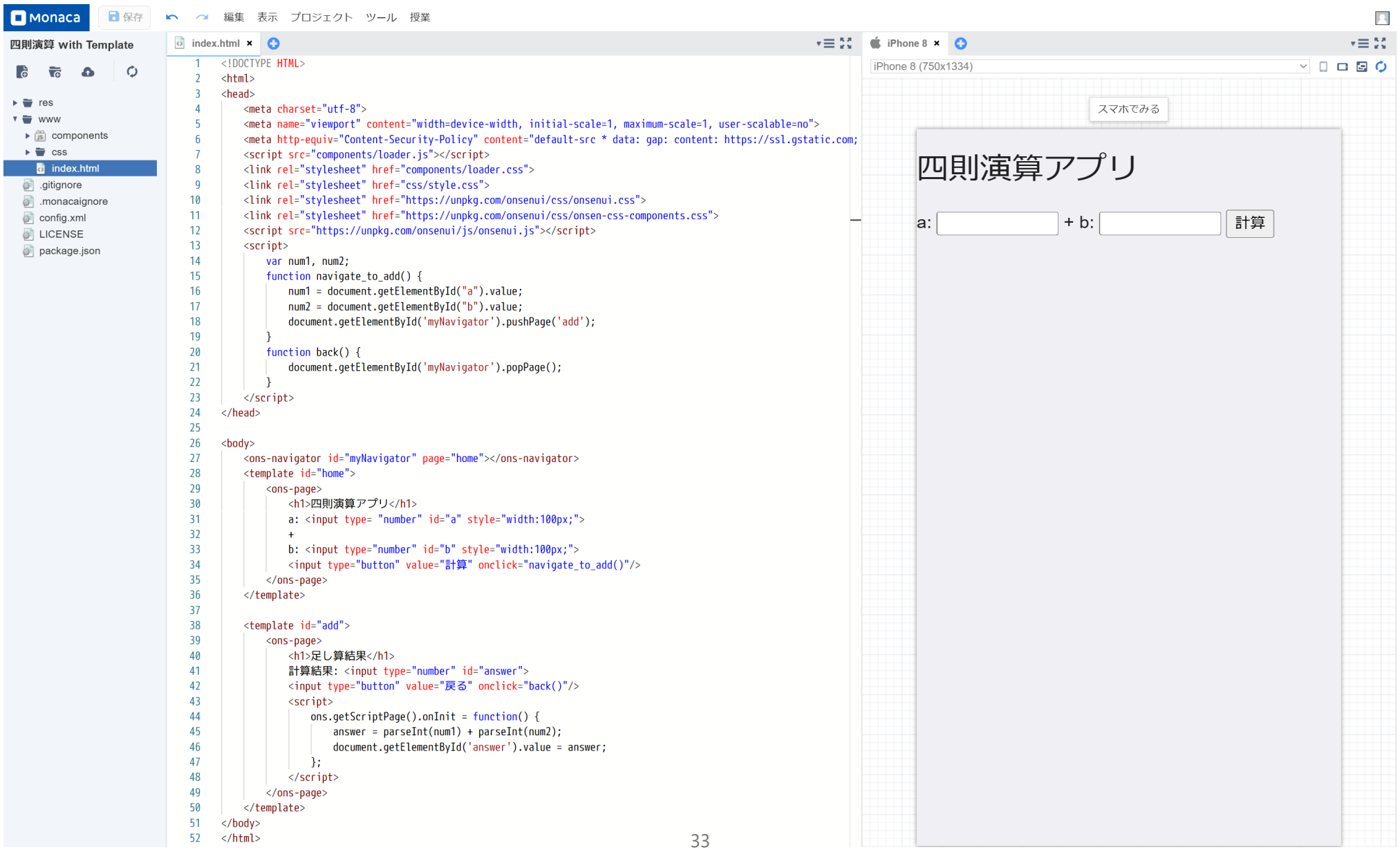
プログラムは **<script> </script>** タグ内に記載する必要があります。

まめ知識

ページが切り替わるときのイベントには4種類あります。

```
ons.getScriptPage().onInit = function() { ページを読み込んだとき（表示される前）の処理 }
ons.getScriptPage().onShow = function() { ページが表示されたときの処理 }
ons.getScriptPage().onHide = function() { ページが見えなくなった（他ページがpushPageされた）ときの処理 }
ons.getScriptPage().onDestroy = function() { ページが削除された(popPageされた)ときの処理
```

完成




```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: https://ssl.gstatic.com;
    <script src="components/loader.js"></script>
    <link rel="stylesheet" href="components/loader.css">
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsenui.css">
    <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsen-css-components.css">
    <script src="https://unpkg.com/onsenui/js/onsenui.js"></script>
    <script>
        var num1, num2;
        function navigate_to_add() {
            num1 = document.getElementById("a").value;
            num2 = document.getElementById("b").value;
            document.getElementById('myNavigator').pushPage('add');
        }
        function back() {
            document.getElementById('myNavigator').popPage();
        }
    </script>
</head>

<body>
    <ons-navigator id="myNavigator" page="home"></ons-navigator>
    <template id="home">
        <ons-page>
            <h1>四則演算アプリ</h1>
            a: <input type= "number" id="a" style="width:100px;">
            +
            b: <input type="number" id="b" style="width:100px;">
            <input type="button" value="計算" onclick="navigate_to_add()"/>
        </ons-page>
    </template>

    <template id="add">
        <ons-page>
            <h1>足し算結果</h1>
            計算結果: <input type="number" id="answer">
            <input type="button" value="戻る" onclick="back()"/>
            <script>
                ons.getScriptPage().onInit = function() {
                    answer = parseInt(num1) + parseInt(num2);
                    document.getElementById('answer').value = answer;
                };
            </script>
        </ons-page>
    </template>
</body>
</html>
```

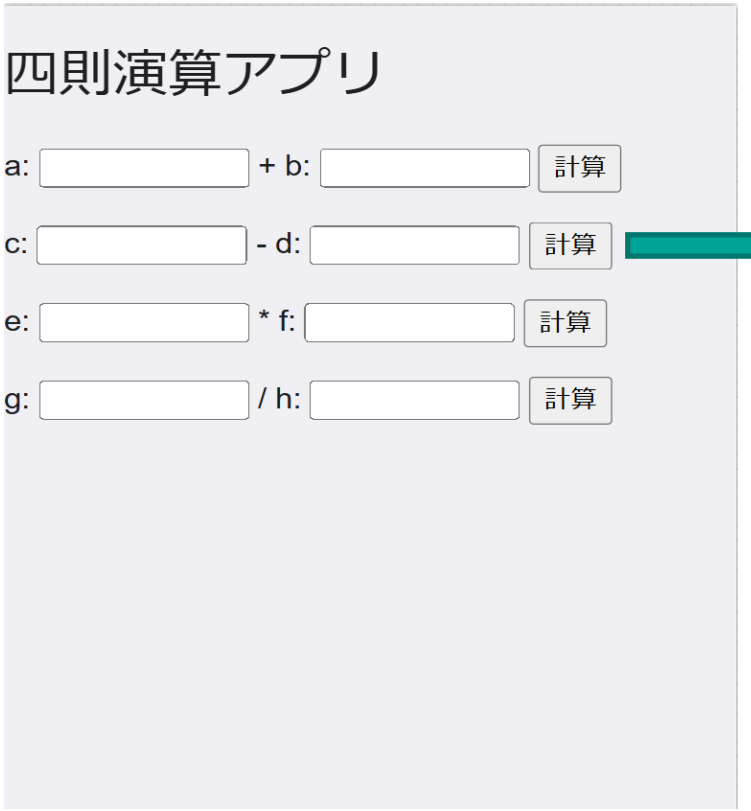
課題２、

解説編

ChatGPTを使ってみよう

課題2：「次の画面のようなアプリを作ってください。
先ほどのアプリを四則演算ができるように拡張してください。」

課題2 の subtract 画面を ChatGPT を使って作ってみましょう。



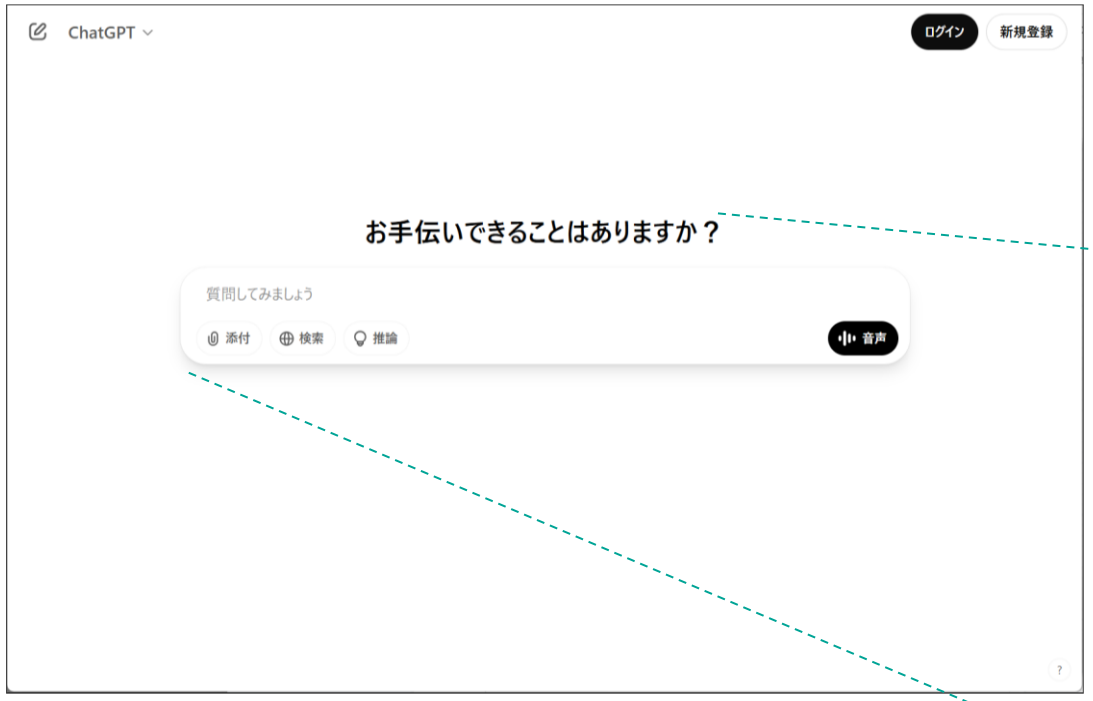
home画面

subtract画面

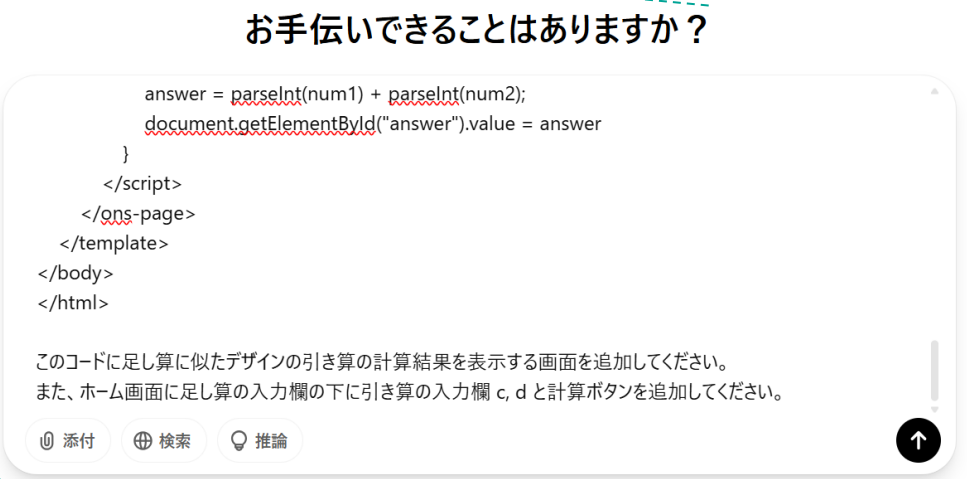
ChatGPTを使ってみよう

https://chatgpt.com/
このURL から ChatGPT にアクセスします。
ChatGPTはログインしなくても使用できます。

上のような表示は、
「ログアウトしたままにする」
を選択します。



ChatGPTページ



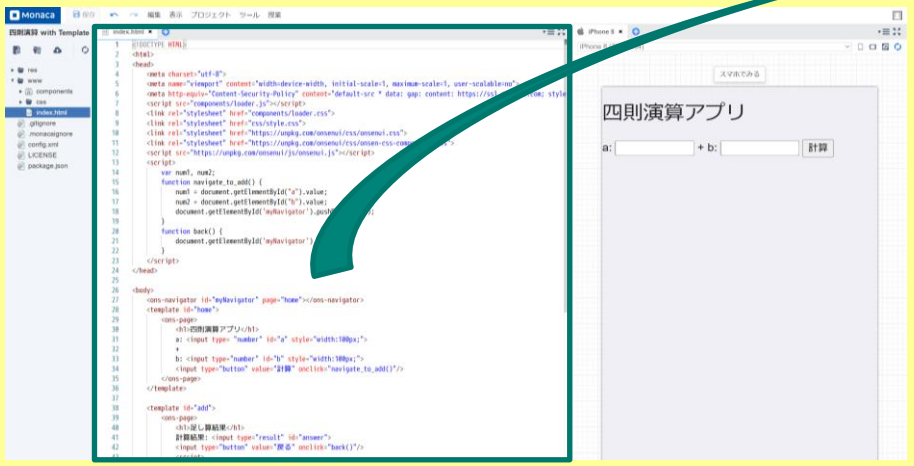
作りたい機能を記入して作ってもらいましょう。

ChatGPTを使ってみよう

ChatGPT に質問するときは、下のように質問内容を明確にしましょう。

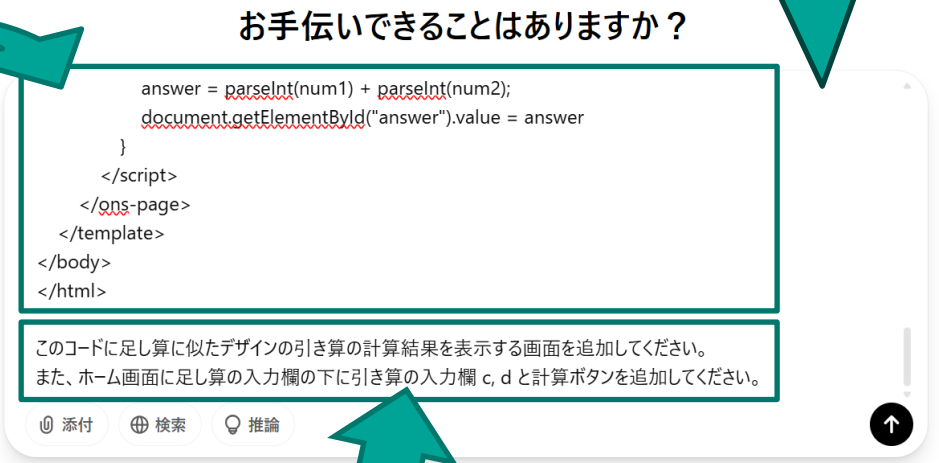
質問欄は Ctrl + Enter で改行できます

・プログラム全文を載せる



プログラム全文

作りたい機能



・作りたい機能を具体的に書く

✕ このコードに引き算画面を追加してください。

○ このコードに足し算に似たデザインの引き算の計算結果を表示する画面を追加してください。
また、ホーム画面に足し算の入力欄の下に引き算の入力欄 c, d と計算ボタンを追加してください。

追加してほしいことは具体的に書こう

ChatGPTを使ってみよう

質問の回答をそのまま、コピーしてMonacaに貼り付けて動きを見てみましょう。

お手伝いできることはありますか？

```
            answer = parseInt(num1) + parseInt(num2);
            document.getElementById("answer").value = answer
        }
    </script>
  </ons-page>
</template>
</body>
</html>
```

プログラム全文

このコードに足し算に似たデザインの引き算の計算結果を表示する画面を追加してください。
また、ホーム画面に足し算の入力欄の下に引き算の入力欄 c, d と計算ボタンを追加してください。

作りたい機能

添付　検索　推論

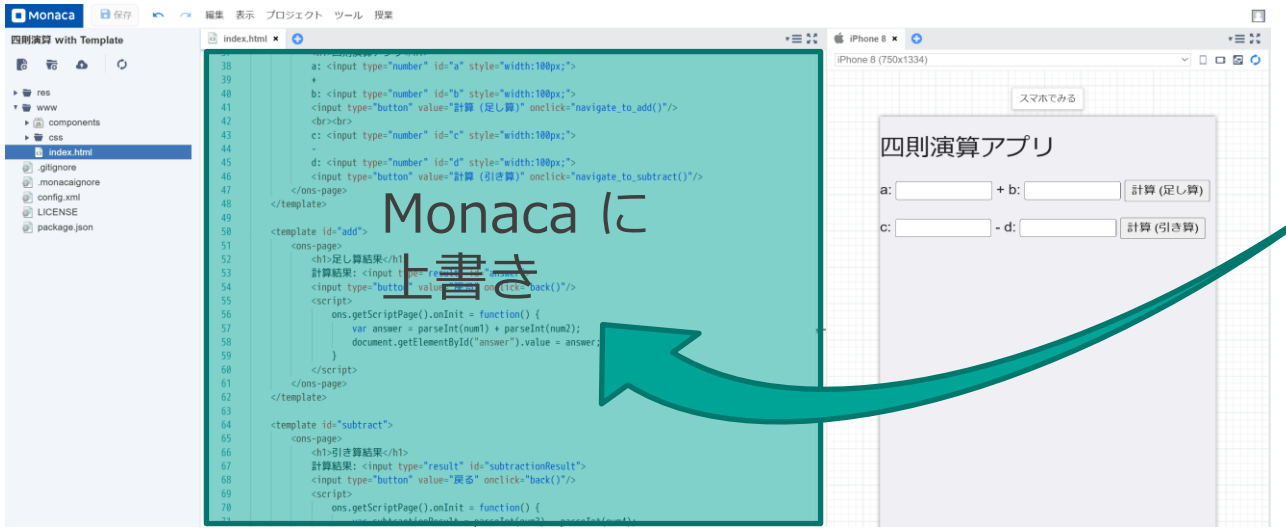
質問

ChatGPTからの回答

以下のようにコードを修正してみてください：

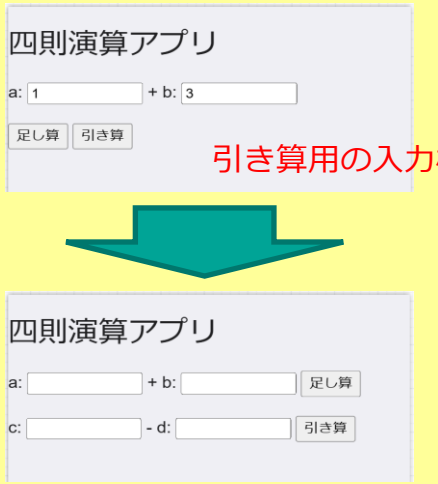
html　コピーする

```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-s
    <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: https
    <script src="components/loader.js"></script>
    <link rel="stylesheet" href="components/loader.css">
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsenui.css">
    <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsen-css-components.css">
    <script src="https://unpkg.com/onsenui/js/onsenui.js"></script>
    <script>
        var num1, num2, num3, num4;
        function navigate_to_add() {
            num1 = document.getElementById("a").value;
```



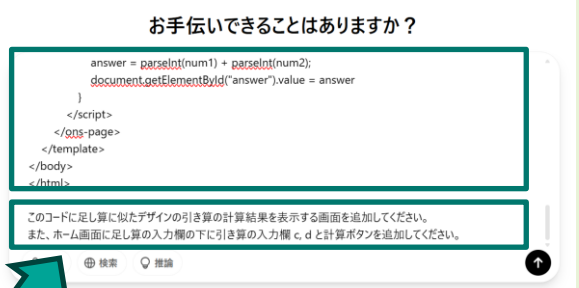
うまくいかなかったときは…

・UIが違うときは



違う箇所をより具体的にして聞き直しましょう。

最初に作った(元の)
プログラム全文



質問内容

より具体的な作りたい機能

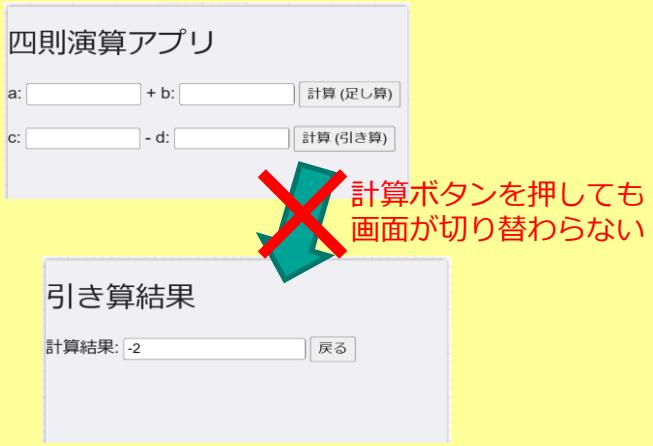
このコードに引き算画面を追加してください。

より具体的に

このコードに引き算画面を追加してください。
ホーム画面には、引き算用の入力欄c, d と計算ボタンを追加してください

違う箇所を具体的に追記する

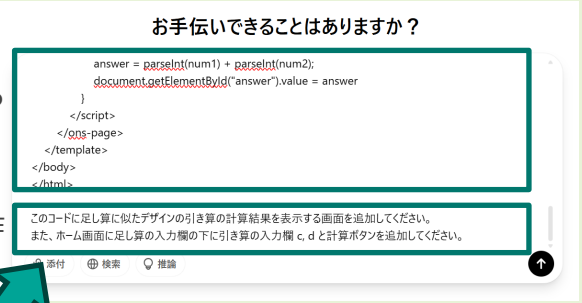
・プログラムの動きが違うときは



期待する動作と現在の動作を説明して、
修正してもらいましょう。

問題が起きている
プログラム全文

直してほしい動作



質問内容

引き算の計算ボタンを押すと引き算画面に切り替わり、計算結果を表示してほしいのですが、

計算ボタンを押してもページが切り替わりません。修正したプログラムを教えてください。

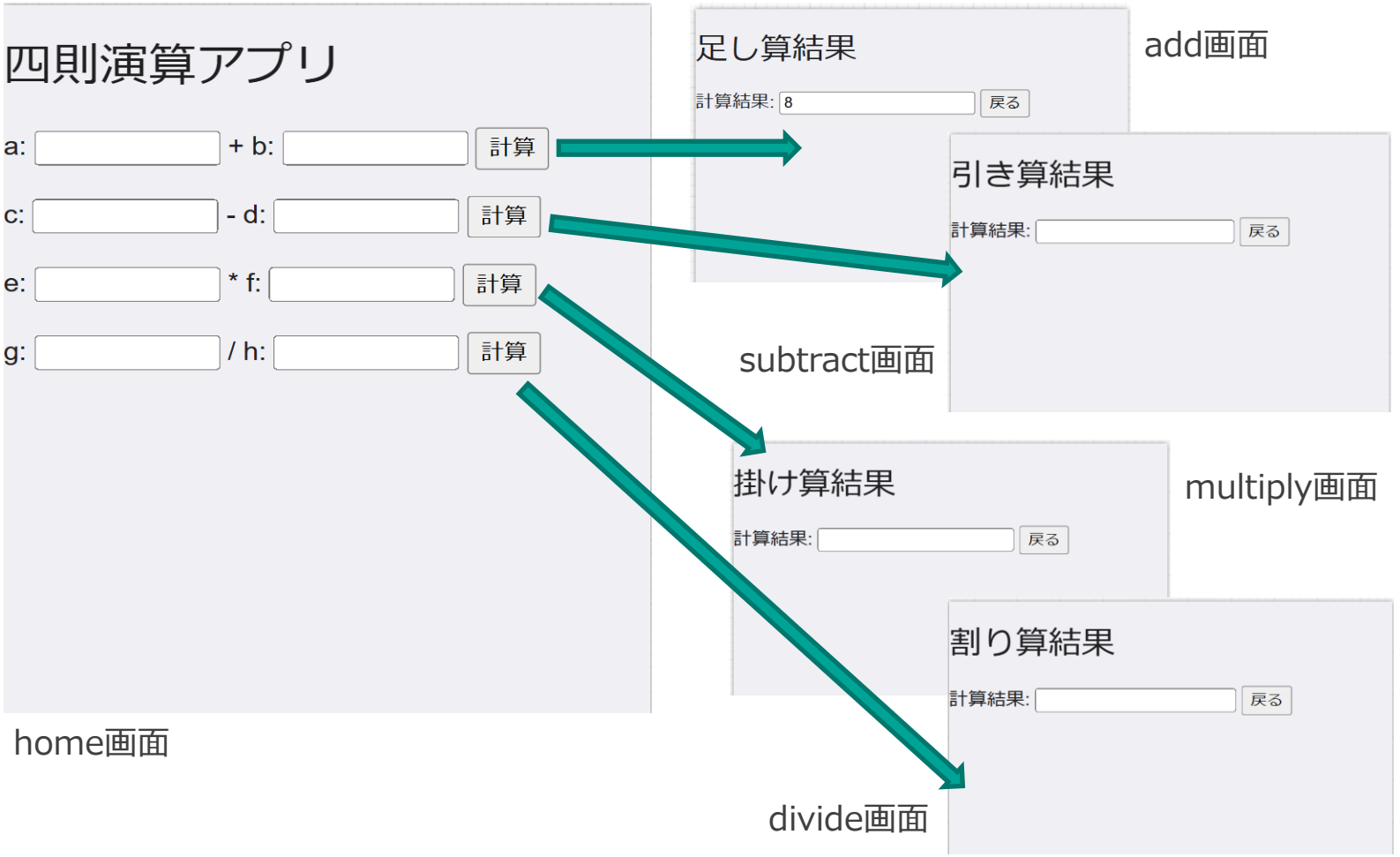
現在の動作を伝える

期待する動作を伝える

課題２

引き続き他の画面も作ってみましょう。

ChatGPTを使っても、自分で作っても構いません。



課題２、

解答編

課題2　解答

https://x.gd/5zWA3c

回答

参考

- OnsenUIを使ったこんなやり方もあります。
  - この記事を参考に興味のある生徒は試してみてください。

https://note.com/fast_chimp4740/n/n5c4385fedc3e

まとめ

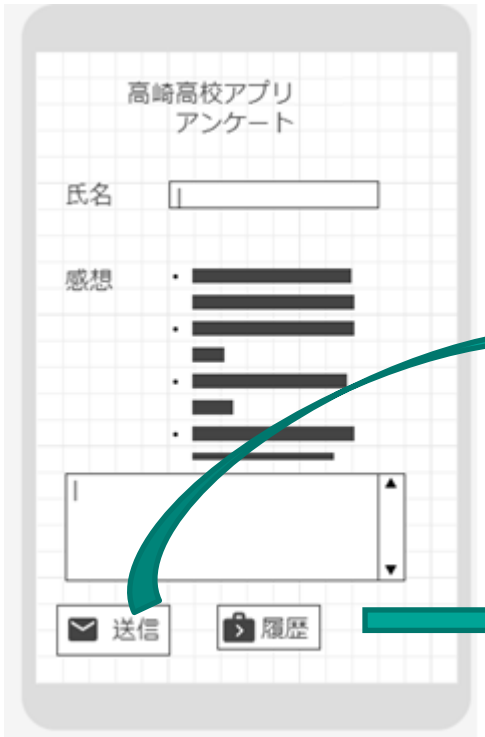
貴重なお時間をいただきありがとうございました。

今回の授業を通してアプリ作成に少しでも興味が持てた生徒は

具体的に手書きでも良いのでワイヤーフレームを考えてみましょう。

余談：ワイヤーフレームを作成しましょう。

- ワイヤーフレーム　https://wireframe.cc/
  - アプリを作成するにあたり、まずはワイヤーフレームを作成します。簡素なものですが実際に作成したワイヤーフレームを下記に掲載します。今回はスマートフォンを対象に縦向きの画面操作を前提に作成しています。みなさんも是非作成してみてください。



https://wireframe.cc/AcCeeT

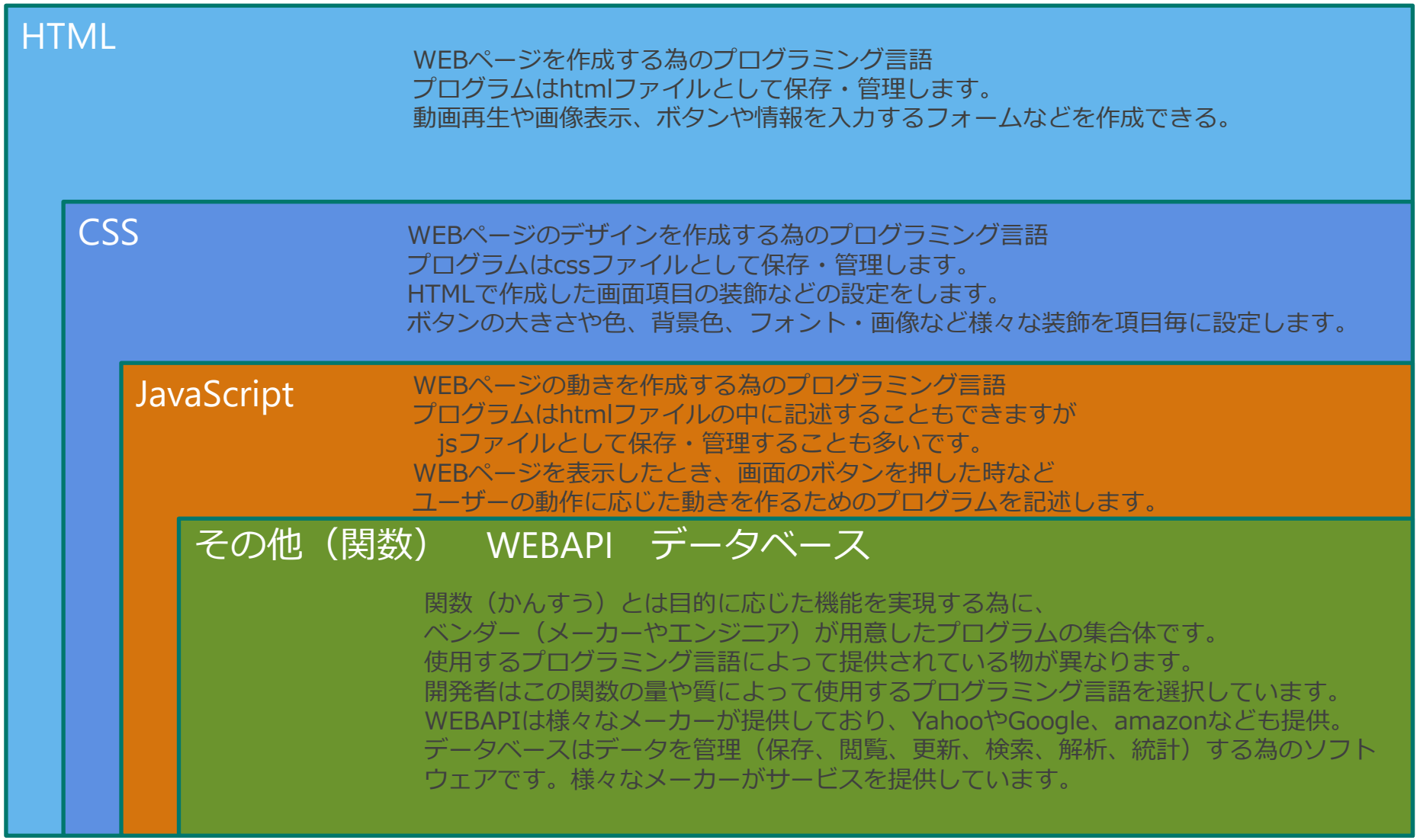
https://wireframe.cc/6BipCP

https://wireframe.cc/tblTUv

基礎知識

参考資料

基本知識：WEBアプリの基本機能と構成について



基本知識：プログラムの関数や引数について

ボタンや画面に文字を表示したり、計算をさせたり、処理をさせるにはJavascriptで関数(function {})を作る必要があります。
Javascriptの関数は**<script> </script>**タグ内に記載する必要があります。

覚えて！！

関数の作り方

function 好きな名前(引数,引数….) { }

①関数を定義します。
function 関数名 (引数)
{ 関数の開始コード

} 関数の終了コード

②引数（ひきすう）を定義します。
使う場合のみで大丈夫です。
今回は使いません。

引数についてはこちらを見てください。
https://blog.senseshare.jp/argument.html

```
<script>
   function calc_sum(      ){

   }
</script>
```

好きな関数名を記述します。ここではcalc_sum関数としました。

まめ知識

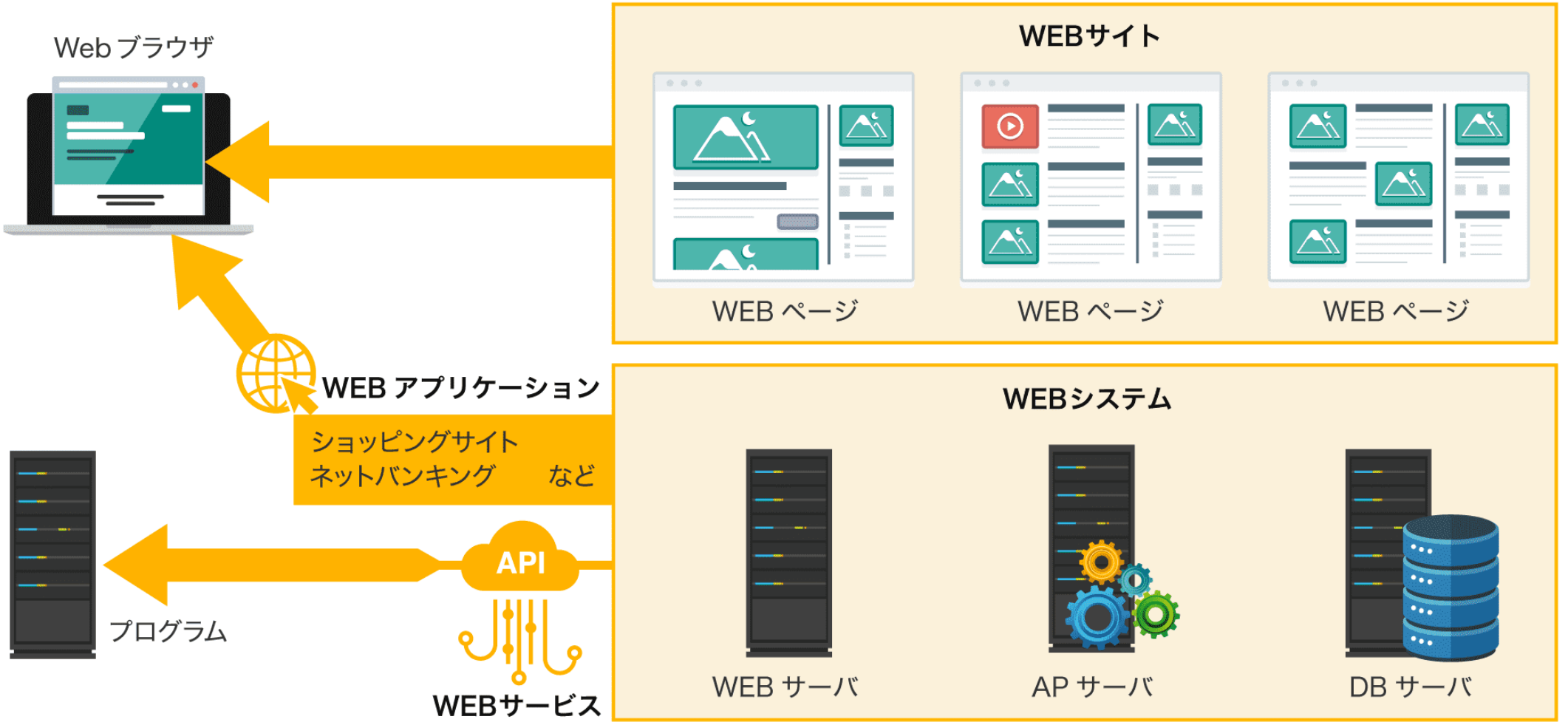
関数の定義には２種類あります。Function(ファンクション) と procedure（プロシージャ）の違い

「値を返すか返さないか」ということです。関数から返値（結果、値）を返せるものがファンクションです。
もちろん返値を返さなくても大丈夫です。 Functionとreturnはセットになることが多いです。
また、一般に procedure のことを「手続き」と呼び、 function のことを「関数」と呼びます。
これは全てのプログラミング言語で共通の知識となるので覚えておいて損はありません。

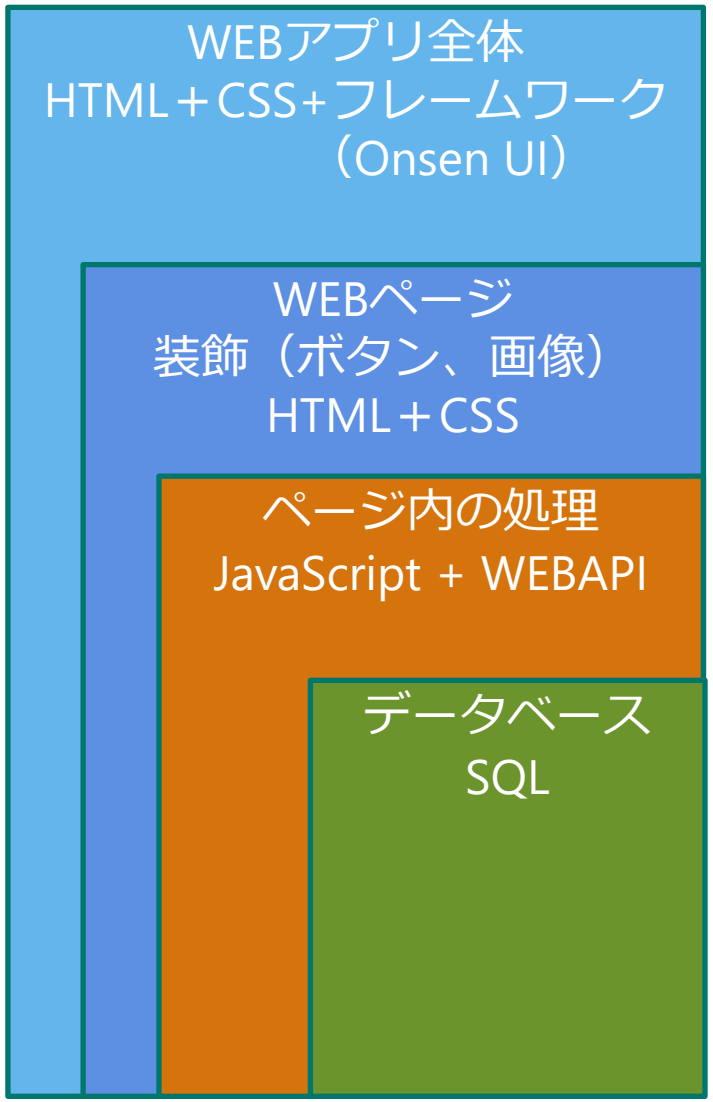
WEBアプリケーションとは？

WEBアプリケーションとは？

WEBサービス、WEBサイト、WEBシステムの違い



WEBアプリの基本機能と構成について



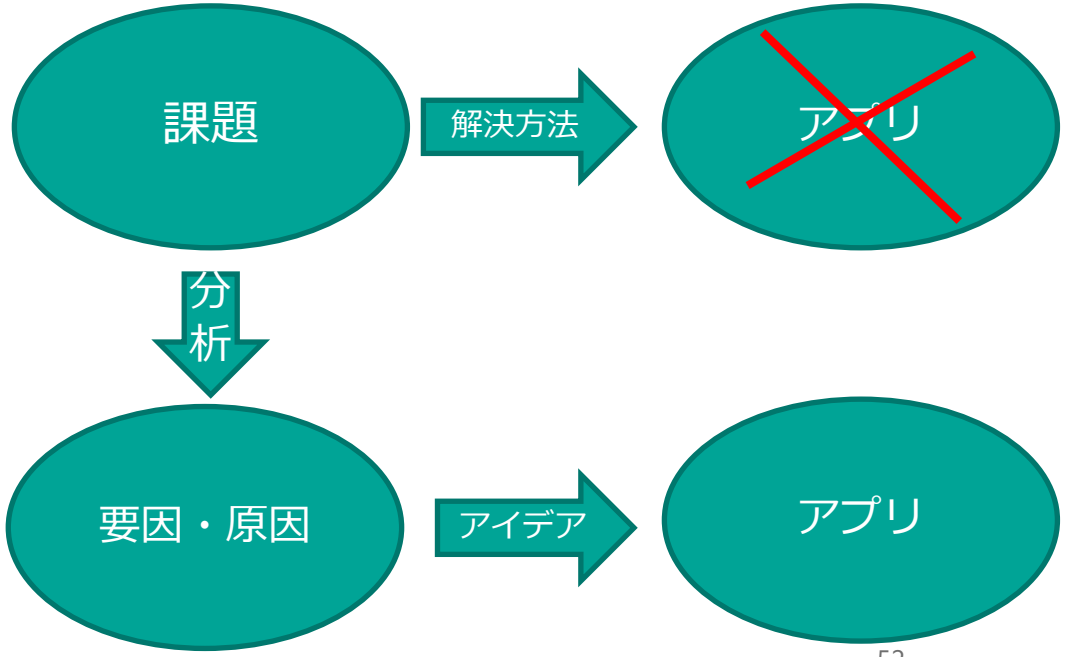
Onsen UI：https://ja.onsen.io/

モバイルアプリ開発に特化したオープンソースのUI**フレームワーク**

アプリ製作の流れ

★テーマを決める。

誰のために
どんなアプリを作りたいのか
どんなサービスを提供したいのか
どんな機能を作りたいのか具体的なイメージを持つ



アプリ製作の流れ

①静的ページを作る。

WEBサイトの骨格を作る。
WEBページ全体をざっくり作成する。

②静的ページに動的ページを追加する。

①で作成した各WEBページに画面遷移など動きを作っていく。
WEBページ全てに画面遷移などの動きが作れたら③へ。

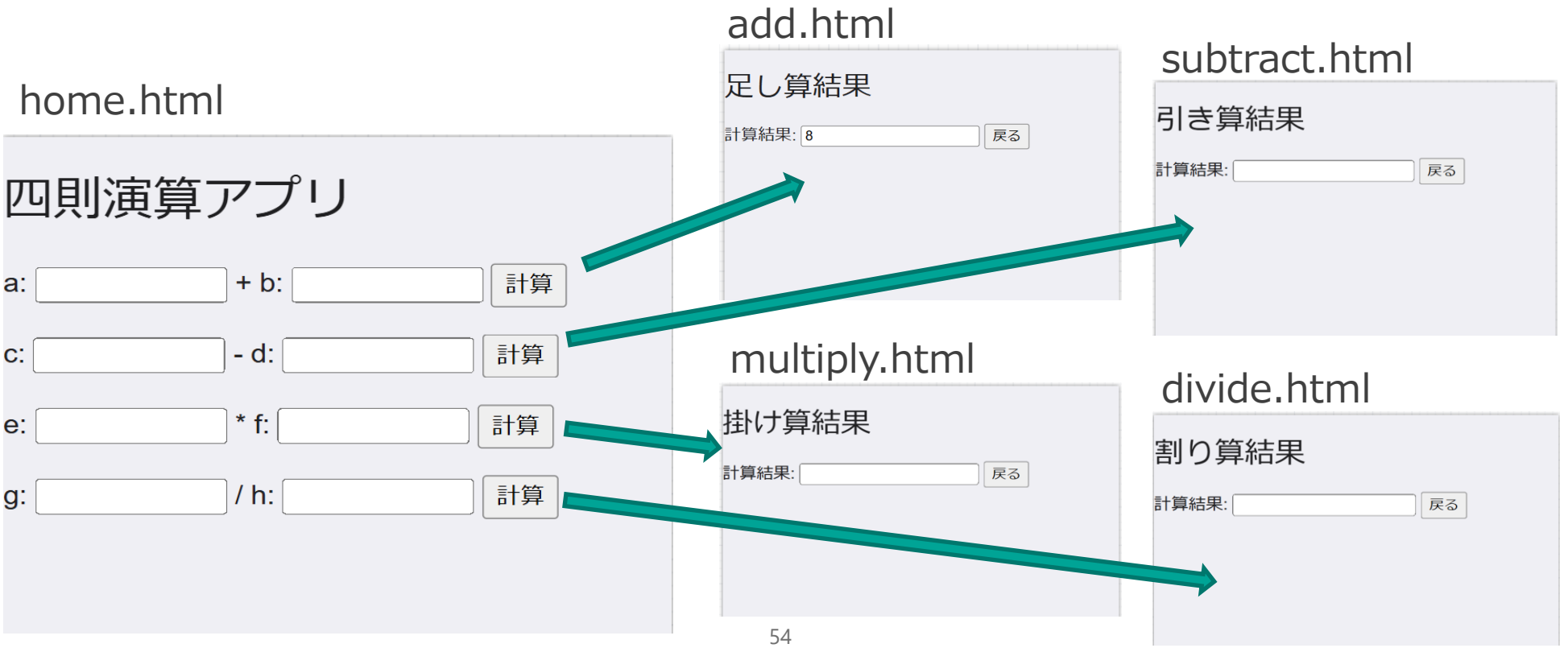
③動的ページに様々な機能を入れる。

アプリケーションサーバーの機能を追加する。
　（地図表示、AI解析など）
データベースサーバーの機能を追加する。
　（データなどを格納したり検索したり表示したい）

基本骨格（WEBサイト）を理解する。

①HTML,CSSを理解する。

1-1 index.html・・・一番最初に実行されるhtmlです。
1-2 home.html・・・ホーム画面のhtmlになります。
1-3 xxxx.html ・・・その他の画面になります。



基本骨格の作りを理解する。（index.htmlページ編）

index.html

```
1  <!DOCTYPE HTML>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
       user-scalable=no">
7      <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: https://
       ssl.gstatic.com; style-src * 'unsafe-inline'; script-src * 'unsafe-inline' 'unsafe-eval'">
8      <script src="components/loader.js"></script>
9      <link rel="stylesheet" href="components/loader.css">
10     <link rel="stylesheet" href="css/style.css">
11
12     <!--OnsenUI-->
13     <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsenui.css">
14     <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsen-css-components.css">
15     <script src="https://unpkg.com/onsenui/js/onsenui.js"></script>
16
17     <script>
18     </script>
19 </head>
20 <body>
21     <!--onsenuiのナビゲ―タ―-->
22     <!--home.htmlを読み込み-->
23     <ons-navigator id="myNavigator" page="home.html"></ons-navigator>
24 </body>
25 </html>
26
```

onsenUIを使用する為の設定

Onsenナビゲーターを使用する為の設定

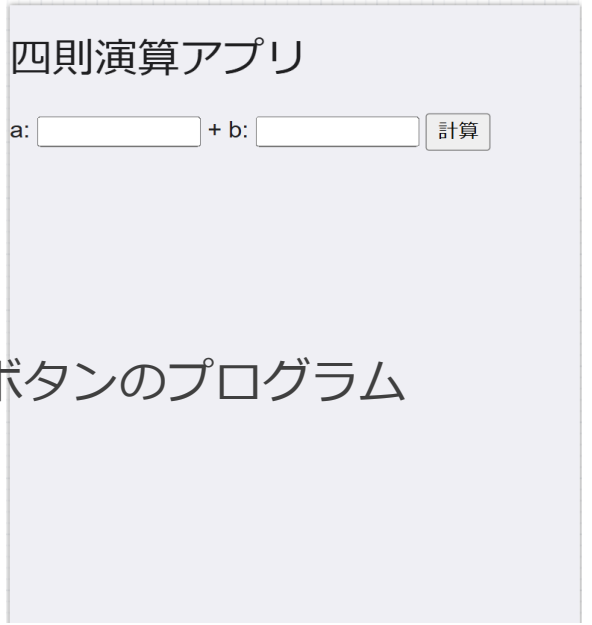
詳細：https://ja.onsen.io/v2/guide/fundamentals.html#pjiwokanrisuru

基本骨格の作りを理解する。（home.htmlページ編）

index.html　home.html ×

```
1  <ons-page>
2      <h1>四則演算アプリ</h1>
3      a: <input type= "number" id="a" style="width:100px;">
4      +
5      b: <input type="number" id="b" style="width:100px;">
6      <input type="button" value="計算"
7          onclick="document.getElementById('myNavigator').pushPage('add.html')"/>
8  </ons-page>
```

計算ボタンのプログラム



document.getElementById(‘myNavigator’).pushPage(‘add.html’)

index.html の <ons-navigator>タグ　　add.html に進む命令

pushPageという命令は、現在表示しているページから、次のページに進むための命令です。
引数に指定したHTMLファイルを画面上に表示します。

Onsen HTML詳細：https://onsen.io/v2/api/js/ons-page.html
Onsen CSS詳細：https://ja.onsen.io/v2/api/css.html

基本骨格の作りを理解する。

template を使うと 複数の html ファイル を 1つのまとめて
index.html のみで作成することができます。

index.html
```
<!DOCTYPE HTML>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <meta http-equiv="Content-Security-Policy" content="default-src * data: gap: content: https://ssl.
    gstatic.com; style-src * 'unsafe-inline'; script-src * 'unsafe-inline' 'unsafe-eval'">
    <script src="components/loader.js"></script>
    <link rel="stylesheet" href="components/loader.css">
    <link rel="stylesheet" href="css/style.css">
    <!--OnsenUI-->
    <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsenui.css">
    <link rel="stylesheet" href="https://unpkg.com/onsenui/css/onsen-css-components.css">
    <script src="https://unpkg.com/onsenui/js/onsenui.js"></script>

    <script>
    </script>
</head>

<body>
    <!--onsenuiのナビゲータ-->
    <!--home.htmlを読み込み-->
    <ons-navigator page="home.html" id="navi"></ons-navigator>

</body>
</html>
```

<template> で囲んだ内容を<body>～</body>で
囲まれている範囲内（黄色背景）に追加します。

home.html の内容を
<template>～</template>で囲みます。
id=“○○” は ○○.html と同じ名前にします。

home.html
```
<template id="home">
    <ons-page>
        <h1>四則演算アプリ</h1>
        a: <input type= "number" id="a" style="width:100px;">
        +
        b: <input type="number" id="b" style="width:100px;">
        <input type="button" value="計算"
            onclick="document.getElementById('myNavigator').pushPage('add.html')"/>
    </ons-page>
</template>
```

基本骨格の作りを理解する。

ページ切り替え方法は同じです。

```
<body>
    <ons-navigator id="myNavigator" page="home"></ons-navigator>

    <template id="home">
        <ons-page>
            <h1>四則演算アプリ</h1>
            a: <input type= "number" id="a" style="width:100px;">
            +
            b: <input type="number" id="b" style="width:100px;">
            <input type="button" value="計算"
                onclick="document.getElementById('myNavigator').pushPage('add')"/>
        </ons-page>
    </template>

    <template id="add">
        <ons-page id="answer" oninit="calc_add()">
            <h1>足し算結果</h1>
            計算結果: <input type="result" id="c">
            <input type="button" value="戻る" onclick="back()"/>
        </ons-page>
    </template>
</body>
```

<ons-navigator id="myNavigator" page="home.html"></ons-navigator>
<ons-navigator id="myNavigator" page="home"></ons-navigator>

document.getElementById('myNavigator').pushPage('add.html')
document.getElementById('myNavigator').pushPage('add')

<template id="add">

ページの指定方法がファイル名(add.html) から id (add) に変わります。

引用・参考

- Monacaで学ぶアプリ製作入門動画
  - https://www.youtube.com/playlist?list=PL5PYiMzYA6FvWdzTfULgtRxZm6OoqW-C0
- JavaScriptで学ぶプログラミング入門
  - https://www.youtube.com/playlist?list=PL5PYiMzYA6FshAmrnAnP_YFB--rFDAz7V

開発者のための参考ドキュメント

- Google

  https://www.google.com

- MDN 開発者向けのウェブ技術

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

- chatGPT

  https://chatgpt.com/

- JQueryライブラリ

  https://releases.jquery.com/jquery/