

WEB フロントエンド Chapter-10

「問い合わせページの作成（HTML 編）」

< 想定開発環境 >

OS：Windows 10 または 11

エディタ：Microsoft Visual Studio Code（通称 VSCode）

ブラウザ：Google Chrome

※ 参考教科書紹介

「HTML & CSS & Web デザインが 1 冊できちんと身につく本」：技術評論社

※ 注意事項

- ・ 講師の解説が始まったら私語は慎み、解説に集中してください。
- ・ 一度行った説明を再度行うケースが多く、授業進行が遅れてしまいます。
- ・ 教室にはマイクが無く、他の方の話声で解説内容が聞き取れない場合があります

① フォームとは？

一概に WEB サイトと言っても、そのサイトが持つ機能によって必要な技術は変わってきます。単なる自己紹介ページであれば、今まで学習してきた知識だけでも十分作成出来るでしょう。

しかし掲示板サイトや通販サイトなどでは、閲覧者からの情報発信を反映させたり、レスポンスを返さなければなりません。掲示板の書き込みやアカウント情報は自分の PC ではなく、サーバという別の PC に保存されるからです。

このデータを送る際に必要になるのが HTML の「フォーム」という機能です。

フォーム機能は HTML 上でフォーム用のタグを記述して利用します。

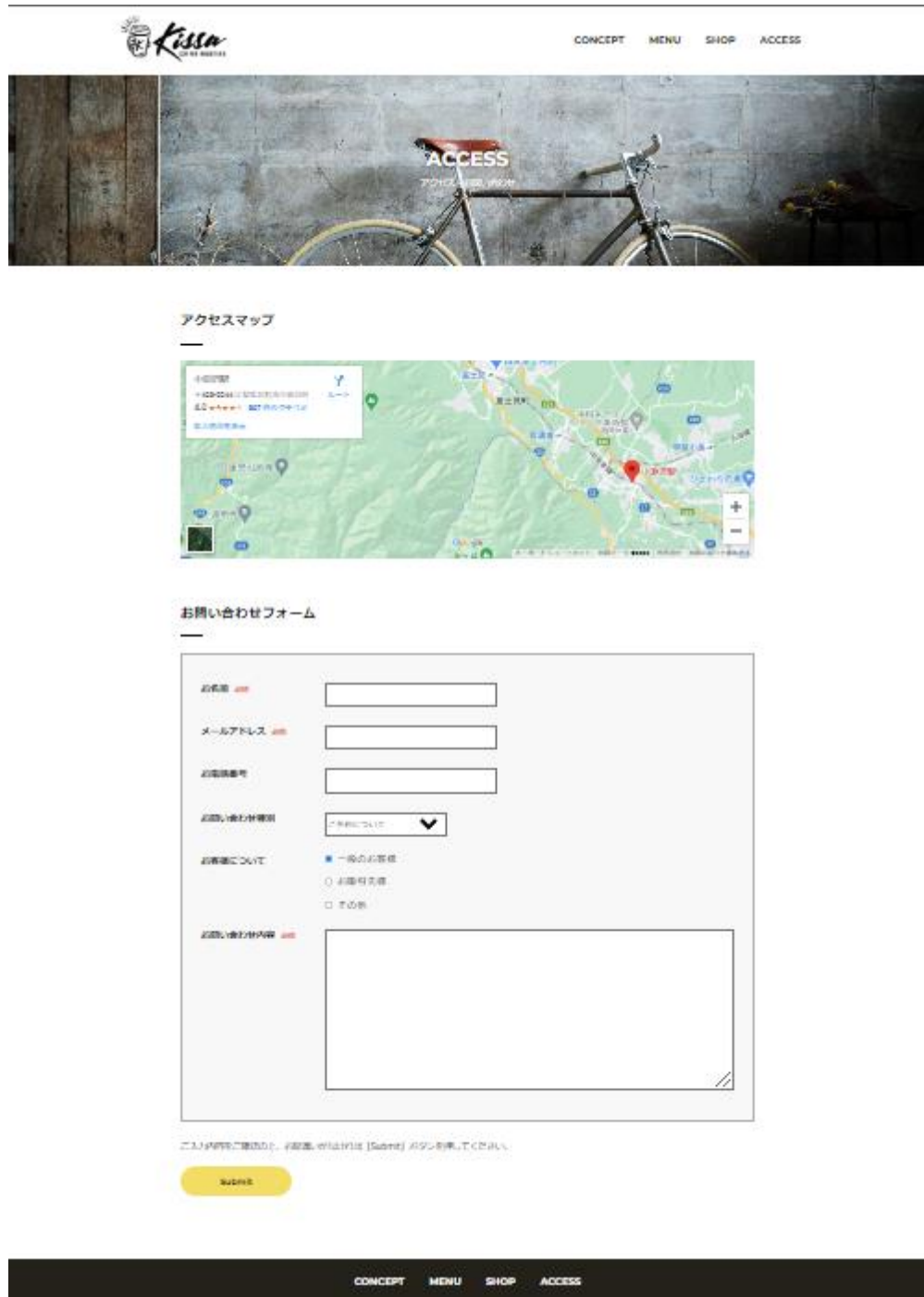
今回は access ページに地図を載せ、お店への問い合わせを想定したフォームを作成していきます。

既に access ページはありますが、前回同様 common.html を複製して作成します。古い access.html を削除し、common.html を複製して access.html にリネームしましょう。

やり方が分からない方は WEB フロントエンド 08 を参考にしてください。

② HTML タグの記述

<完成イメージ>



一番上に背景画像を使ったページタイトル、次にお店の地図、そして問い合わせフォームという構成です。まずはHTML タグを記述していきます。

◇ 概要と css へのパス記述

複製した access.html の head 内を修正します。ページ概要を変更し、access ページ用の css のパスを追加してください。css ファイルは後程作成します。

<access.html>

```
<title>KISSA official website</title>
●<meta name="description" content="アクセスマップとお問い合わせ">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script src="./js/toggle-menu.js"></script>
<link href="./css/common.css" rel="stylesheet">
●<link href="./css/access.css" rel="stylesheet">
```

head 内の変更なので、ブラウザ表示は特に変わりません。

◇ main への記述

main 内には「タイトル」「地図」「問い合わせフォーム」の3つを配置します。

<access.html>

```
<main class="main">
  <div class="title">
    <h1>ACCESS</h1>
    <p>アクセス・お問い合わせ</p>
  </div>
  <div class="map">
    <h2>アクセスマップ</h2>
  </div>
  <div class="contact">
    <h2>お問い合わせフォーム</h2>
  </div>
</main>
```

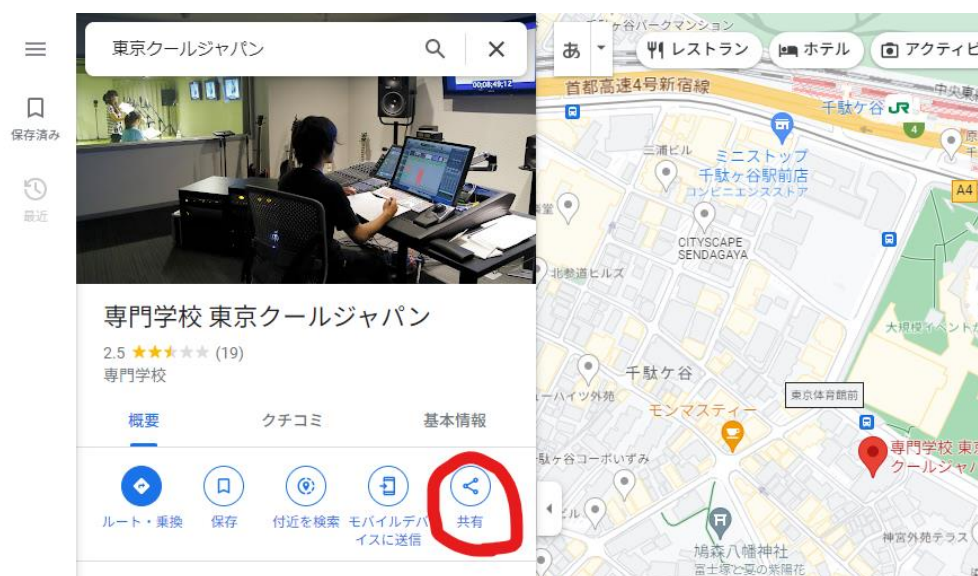
今回のように既にデザインが決まっている場合、初めに部分ごとにタグを記述してそれぞれのエリアを明確に分けておくと作りやすいです。

実はこういったパーツの切り分けというのはプログラミングではとても重要で、切り分ける事でチーム制作での分担や、バグなどの問題の特定に非常に役に立ちます。

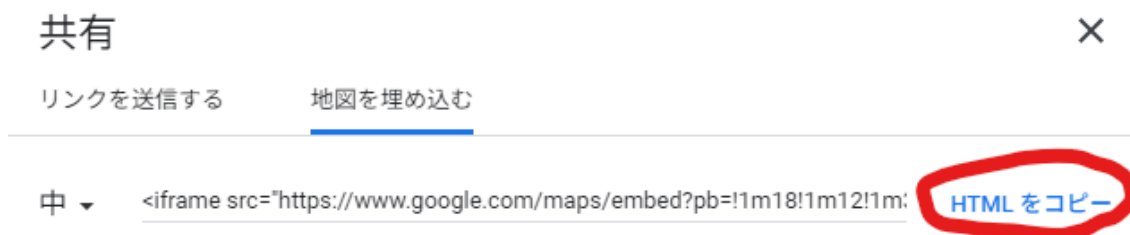
◇ Google マップの埋め込み

地図サイトはいくつかありますが、今回はおそらく最も利用者の多い Google Map を利用します。埋め込む手順は次の通りです。

- ・ブラウザで google マップを検索し表示。
- ・左上の検索窓に、表示させたい場所の名前や住所を入力し検索。
思いつかない人は「東京クールジャパン」などで検索してください。
- ・左側のウィンドウにある「共有」をクリック（下記の写真参照）



- ・「地図を埋め込む」を選択し、「HTML をコピー」をクリック



- ・ access.html の以下の位置に貼り付ける（下記の位置。長いため省略表示しています）

```

<div class="map">
  <h2>アクセスマップ</h2>
  <iframe src="https://www.google.com/maps/embed?
pb=!1m18!1m12!1m3!1d3240.9449555141055!2d139.
70845807611025!3d35.

```

<ブラウザ表示>



- ・ この時点でも地図が表示されますが、高さとは CSS で指定するので削除しておく

```

2msqcn10aduseS6r00CF-0bV00Dq-0C0000-0bke0DSW!5e0!3m2!1sja
v1685607569769!5m2!1sja!2sjp" width="600" height="450"
style="border:0;" allowfullscreen="" loading="lazy"

```

<ブラウザ表示>



- ・表示が小さいですが、ウィンドウに合わせて伸び縮みするように後程修正します。
- ・というわけで、地図の埋め込みはこれで完成です！

◇ フォームの記述

次に問い合わせフォーム部分の概要を記述します。フォームの概要については1ページ目を参照してください。

フォームには色々な機能がありますが、全ての機能は form タグの中に記述する決まりがあります。そのため次のように form 要素を追加します。

<access.html>

```
.....<div class="contact">
.....
.....<h2>お問い合わせフォーム</h2>
.....
.....<form action="#"></form>
.....
.....</div>
```

action という属性が書かれていますが、ここではフォームからデータを送信する際に使用するプログラムを指定します。

今回は送信部分の記述は行いませんので、自ページの一番上を意味する"#"を指定しています。リンク先が決まっていなくても仮の URL として使う事が多いので、覚えておくの良いでしょう。

なお form の中身はまだ無いため、ブラウザ表示に変化はありません。

それでは form の中身を書いていきましょう。dl 要素の中を記述していきます。

<access.html>

```
... <form action="#">
... <dl class="form-area">
...   <dt><span class="required">お名前</span></dt>
...   <dd></dd>
...   <dt><span class="required">メールアドレス</span></dt>
...   <dd></dd>
...   <dt>お電話番号</dt>
...   <dd></dd>
...   <dt>お問い合わせ種別</dt>
...   <dd></dd>
...   <dt>お客様について</dt>
...   <dd></dd>
...   <dt><span class="required">お問い合わせ内容</span></dt>
...   <dd></dd>
... </dl>
... </form>
```

途中いくつかの要素に「required」という class 名がつけられていますが、これは「必須入力項目」を表すために付けられたものです。

【dl, dt, dd】要素

説明リストと呼ばれる要素群です。Q&A や FAQ のように、タイトルと解説が対になる形式の一覧を表すために使われます。

- | | | | |
|------|---|------------------------|-----------------------|
| ・ dl | : | definition list | この中が説明リスト |
| ・ dt | : | definition title | 説明要素のタイトル。Q&A の Q。 |
| ・ dd | : | definition description | 上のタイトル対する説明文。Q&A の A。 |

表示項目が2つになったリスト（ul や ol）だと考えれば良いでしょう。

デフォルトで何か特殊なスタイルが付けられているわけではないのでスタイルという観点からすると div 要素などと変わりません。

この要素を使う理由としては可読性を高めるためという点と、SEO 対策という点が挙げられます。

これらの要素についてはこちらのサイトが分かりやすかったので、気になる方は参考にしてください。

prograshi さん

<https://prograshi.com/language/html/dl-dt-dd/>

【span】要素

div 要素と同様、グループ化するための要素です。div 要素とは違って span はインライン要素になります。そのため改行させたくない場合に使用します。

今回の場合はタイトルの横に「必須 (required)」という文字を並べて表示させたいので、class 名に required と名付けるために使用しています

(次の input 要素に出てくる「required」属性とは直接関係ありませんので注意！)

◇ テキスト入力フォームの記述

それではいよいよ各種入力フォームの記述に入りましょう。

入力フォームには「テキスト入力」「ラジオボタン・チェックボタン」「セレクトボックス」など、いくつかの種類があります。

初めはテキスト入力の記述から行います。お名前欄の dd 要素の中に、input 要素を次のように追加してください。

<access.html>

```
<dl class="form-area">
  <dt><span class="required">お名前</span></dt>
  <dd><input class="input-text" type="text" name="name" required></dd>
```

<ブラウザ表示>

お問い合わせフォーム
お名前

四角いボックスが追加されましたね。これは type 要素に text を設定したため、このような形の入力フォームになったからです。他にも日付型やチェックボタン型など、様々な形の入力フォームが用意されています。

【input】要素

入力フォーム用の部品を配置する要素。type を変える事で様々なフォーマットでの入力が可能になります。

とても種類が多いので、詳細はこちらを参照してください。

msn web docs（おそらく最も信頼出来る web 系の解説サイト）

<https://developer.mozilla.org/ja/docs/Web/HTML/Element/input>

【type】属性 default 値 : text

入力フォームの型を指定するための属性です。テキスト型、ボタン型、日付型など、たくさんの種類があります。これも詳しくは上記のサイトを参照してください。

【name】属性

この属性は主にデータ送信した先のプログラムが、送られてきたものが何のデータなのかを判別する為に使われる属性です。例えば上の例の場合は「名前」を入力し送信しているので、name 属性を「name（名前）」にしています。

そのため入力フォームに応じたわかりやすい名前を付けるのが鉄則です。

【required】属性

この属性を指定するとフォームへの入力が必須になります。現在はまだ送信ボタンが無いのでわかりませんが、required 指定されている場合に空白のままデータ送信しようとすると、データ送信前にアラートが表示されるようになります。

これについては後程 submit の所で再び解説します。

それでは続いて、メールアドレスと電話番号の入力フォームを同じように追加していきます。

<access.html>

```
...<dd><input class="input-text" type="text" name="name" required></dd>
...<dt><span class="required">メールアドレス</span></dt>
● <dd><input class="input-text" type="email" name="email" required></dd>
...<dt>お電話番号</dt>
● <dd><input class="input-text" type="tel" name="tel"></dd>
```

<ブラウザ表示>

お問い合わせフォーム

お名前

メールアドレス

お電話番号

今回はメールアドレスは必須ですが、電話番号は任意としています。

◇ セレクト入力フォーム（セレクトボックス）の記述

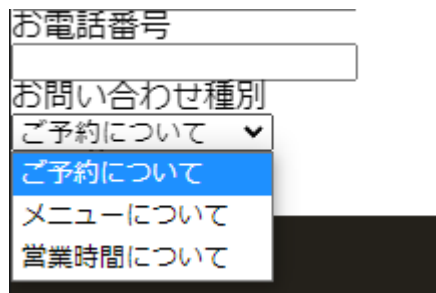
入力する値がいくつか決まっている場合、キーボードで入力するより選んだほうが楽ですし、送信先へ誤字脱字無く、正しく意図を伝えられます。

こういった場合によく使われるのが「セレクトボックス」です。次の赤い部分のコードを、お問い合わせ種別の欄に追加してください。

<access.html>

```
...<dt>お問い合わせ種別</dt>
...<dd>
[ <select class="select-box" name="genre">
  ...<option value="ご予約について" selected>ご予約について</option>
  ...<option value="メニューについて">メニューについて</option>
  ...<option value="営業時間について">営業時間について</option>
  ...</select>
</dd>
```

<ブラウザ表示>



お問い合わせ種別の下にプルダウン形式のボックスが出来ました。
select 要素は、要素内にある option 要素を選択項目として利用します。

【select】要素

「セレクトボックス」とも呼ばれます。選択するだけで入力可能な入力フォームです。option とセットで使います。

【option】要素

select 要素の中で使われる選択肢のための要素です。option 要素の value 属性を設定すると、その value の値が送信されます。そのためブラウザでは閲覧者がわかりやすい表示にしておいて、送信するデータはサーバ側の都合の良いデータを設定しておくといった使い方が可能です。

一番最初の「ご予約について」の option に selected 属性が記述されていますが、これはページを表示した際、selected 指定された選択肢をあらかじめ選択状態にしておくための属性です。最もよくある問い合わせ内容に記述しておくといいでしょう。

セレクトボックスは選択肢が沢山ある場合に有効な入力フォームです。

例えば「出身県」や「年齢・年代」など、送信するデータ形式は決まっているが、選択肢が多すぎて全て表示させるのが難しい場合などによく使われます。

今回の例では選択肢が3つしかありませんが、今後問い合わせ内容が多様化していった場合、選択肢が増える可能性もあります。そういった場合にはセレクトボックスを採用しておいたほうが良いかもしれません。

逆に「性別」のように選択肢が数個しかないような場合は、次の「ラジオボタン」を使用したほうが、閲覧者からするとわかりやすいと思います。

◇ ラジオボタンの記述

ラジオボタンは排他選択（一つしか選べない）形式のボタンで、選択肢が少ない場合に有用な入力フォームです。次の赤い部分を追加してください。

<access.html>

```
<dt>お客様について</dt>
<dd>
  <label class="radio-button">
    <input type="radio" name="user-type" value="一般のお客様" checked>
    一般のお客様
  </label>
  <label class="radio-button">
    <input type="radio" name="user-type" value="お取引先様">
    お取引先様
  </label>
  <label class="radio-button">
    <input type="radio" name="user-type" value="その他">
    その他
  </label>
</dd>
```

<ブラウザ表示>

お客様について

● 一般のお客様 ○ お取引先様 ○ その他

「一般のお客様」が checked となっているのはセレクトボックスの selected と同様に、デフォルトの選択肢を指定したものです。

【label】要素

フォームにラベル（何ボタンなのかを記述したテキスト文）を付ける為の要素です。今回の場合、radio ボタン自体を label 要素で囲んでいるため、ボタンだけでなくラベルをクリックしてもボタンが反応するようになっています。クリックで反応する範囲が広くなり操作性が向上するため、このような記述はよく見られます。

今回 input 要素中で使われている type 属性の"radio"は、排他選択（他のものは選べず、一つのみ選択可）の場合に使われるボタンです。先程も書きましたが閲覧者が選択肢を全て確認出来るので、選択肢がさほど多くない場合は select よりもこちらを使うと良いでしょう。

因みに複数を選択させたい場合には"checkbox"を使用します。

◇ テキストエリアの記述

input 要素の text タイプは短い文章を送信するためのフォーム部品ですが、入力欄が一行しかないため、長文を送信するには不向きです。

そういった場合に利用される要素が「textarea」というフォームです。

次の赤い部分のコードを追加してください。

<access.html>

```
<dd>  
[ <textarea class="message" name="message" required>  
  </textarea>  
</dd>
```

<ブラウザ表示>

お問い合わせ内容

input 属性よりも一回り大きな入力欄が表示されました。入力欄の右下に斜め線のようなものが入っていますが、ここをドラッグすると入力欄の大きさを変える事が出来ます。

また入力欄の大きさは手動以外にも、"rows"で行数（高さ）を、"cols"で列数（横幅）を指定する事も出来ます。

【textarea】要素

複数行を入力するためのフォームで、改行も可能です。input 要素でタイトルを、textarea 要素で本文を入力するというような使い方をします。

◇ 送信ボタンの記述

さて HTML の記述もあと少しで終わりです。頑張って記述していきましょう！
フォーム要素の終了タグの直前に、次のコードを追加してください。

<access.html>

```
</dl>
<p class="confirm-text">ご入力内容をご確認の上、お間違いがなければ [Submit] ボタンを押してください。</p>
<button class="submit-button" type="submit">Submit</button>
</form>
```

<ブラウザ表示>

ご入力内容をご確認の上、お間違いがなければ [Submit] ボタンを押してください。
Submit

WEB サイトを利用する人には様々な方がいらっしゃいますので、閲覧者が何をすれば良いかを明記しておく事は重要です。PC 操作に慣れていない人の中には、フォームに文字を入力しただけで連絡が取れると勘違いしてしまう方もいらっしゃるからです。

ただし、そういった観点からすると[submit]という表記よりも[送信]としておいたほうが親切だとは思いますが。今回は WEB ページ作成の学習が目的ですので、敢えて submit という言葉を使っているのかもしれませんが。

button 要素はハンバーガーメニューで使用しましたが、今回は type 属性に"submit"と入れる事で、フォーム内容の送信ボタンであることを明示しています。

実はフォームの中に button 要素が1しかない場合、その button は自動的に submit ボタンとして扱われるのですが、人為的ミス無くす為にも必ず記述するようにしましょう。

さて、HTML 部分は全て入力が終わりました！

折角ですので、ここで実際に submit ボタンを押してみましょう。

ブラウザではどのように表示されるでしょうか？

<ブラウザ表示>

ブラウザ上に、入力を促す警告が表示されましたね。

これは required 属性を付けた入力欄に、正しく入力されていない時に出る表示です。

試しにメールアドレス欄にアルファベット一文字だけ入れて submit ボタンを押しても、正しいメールアドレスの形式では無いので警告が出ます。

このように、入力された値が正しい形式かをチェックする事を「バリデーション (validation)」と呼びます。

ブラウザに最初からついている機能でもある程度のバリデーションは可能ですが、あまり細かな制御までは出来ません。例えば電話番号や、パスワード入力時に大文字と小文字を必ず両方含めるといったバリデーションを行うには、**正規表現**や Javascript で記述したりする必要があります。

今回はとりあえずブラウザ標準のバリデーションだけで済ましていますが、本格的な開発では、更に細かなバリデーション対応が必須だということを覚えておきましょう。

③ まとめ

WEB サイトの作成もいよいよ大詰めになってきました。

次回は今回記述したタグにスタイルを当てて行きます。

あと少しですので、最後まで完走していきましょう！