

# コーディング 講座

HTML/CSS/  
jQuery

# Today's Agenda:

1. コーディングとは
2. HTML/CSS
3. JSとjQuery
4. 動的/静的サイト
5. コーディングの手順
6. VSコードの使い方
7. コーディングのコツ
8. コーディング実況

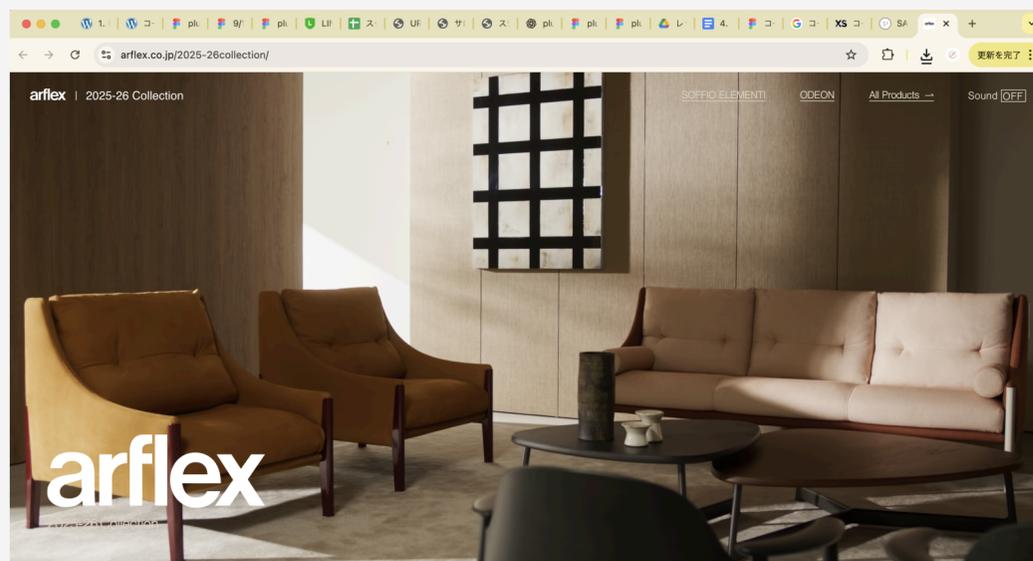
1. コーディングとは
2. HTML/CSS
3. JSとjQuery
4. 動的/静的サイト
5. コーディングの手順
6. VSコードの使い方
7. コーディングのコツ
8. コーディング実況

# O1

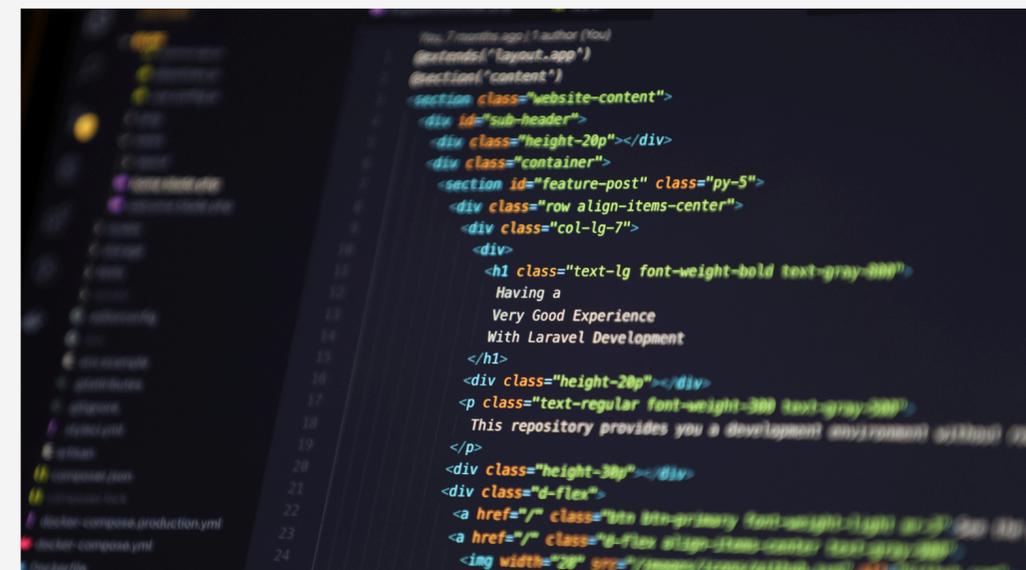
# コーディングとは...

ホームページを表示するために必要な  
「ソースコード（コンピューターに命令する文章）」を書くこと

表側



裏側



コーディングをする人 = コーダー

コーディングをする人 = コーダー

WEBデザイナーはコーディングスキル必須？

# コーディングスキルを学ぶメリット

- **表現の幅が広がる:**  
実装面を考慮したデザインができるようになる
- **仕事の幅が広がる:**  
デザインから実装まで一貫して担当できる案件もこなせるようになり、仕事の機会が増える
- **市場価値が高まる:**  
コーディングもできるデザイナーは、スキルアップや収入向上につながる可能性がある

## コーディングスキルが歓迎条件や必須条件に入っていることも多々

### Webデザイナー

アトム法律情報株式会社 | 東京都千代田区永田町 | 月給 36万円

応募画面に進む



学歴不問

転勤なし

副業・WワークOK

ブランクOK

第二新卒歓迎

研修あり

＼業種未経験歓迎！／

- Webサイトデザインの実務経験が3年以上ある方
- 学歴不問

【あると望ましい経験・能力】

＼以下に当てはまる方は歓迎！／

- Figma、AdobeXDを用いた実務経験がある方
- Photoshop、Illustratorを用いた実務経験がある方
- html/cssのコーディング経験がある方
- スマホを意識したデザイン作成ができる方
- デザインの引き出しが多く、目的に応じて使い分けられる方
- サイトデザインが好きで、他サービスのデザインをよく見る方
- サイト全体の構造、階層を意識したデザイン作りができる方

### Webデザイナー

株式会社貴瞬 | 東京都台東区台東 | 月給 23.5万円 ~ 40.0万円

応募画面に進む



学歴不問

転勤なし

第二新卒歓迎

未経験者歓迎

#### 応募資格：

- illustrator、Photoshopを使用したデザインの実務経験をお持ちの方
- デザインの新しいトレンドや用語へのアンテナが高い方
- htmlを使用できる方
- cssを使用できる方
- 学歴不問

#### 歓迎する職歴・スキル：

- 3DCADの使用経験をお持ちの方（商品のデザインにもチャレンジ可能です！）
- WordPressでのコード入力の実務経験をお持ちの方
- デザイン業務の中でのマーケティング経験をお持ちの方
- ディレクション業務のご経験をお持ちの方
- ワードプレスのテーマ作成を行ったことある方
- JavaScriptを使用できる方

#### 雇用形態：

# コーディングとプログラミング

**プログラミング**=プログラムの設計、コード記述（コーディング）、テスト、デバッグ、修正など、開発プロセス全体の作業。

## コーディングもプログラミングの一部

### コーディング

主にホームページを表示するためのソースコードを作成

使用言語HTML、CSS

### プログラミング

主にプログラムを動作させるためのソースコードを作成

使用言語PHP、Java

1. コーディングとは
- 2. HTML/CSS**
3. JSとjQuery
4. 動的/静的サイト
5. コーディングの手順
6. VSコードの使い方
7. コーディングのコツ
8. コーディング実況

A large, bold, black graphic of the number '02'. The '0' is a simple circle, and the '2' is a stylized, thick, curved shape.

## HTMLとは...

HTML（エイチ・ティー・エム・エル）は「Hyper Text Markup Language」の略称で、ホームページ制作には欠かせないマークアップ言語です。

マークアップとは、文章や画像などをコンピューターが認識できるように、「タグ」を用いて意味付けすることです。

# HTMLの基本的な構造

**タグ**... 「<」と「>」で囲まれた英単語

**開始タグ**... 文章の前にある<head>や<body>

**閉じタグ**... 文章の後ろにある</head></body>などの「/」がついているタグ

HTMLではこの**開始タグ**と**閉じタグ**で**文章などを囲むことで、その中身に役割を与えます。**

```
<h1>
```

**例**    この文章が見出しになります

```
</h1>
```

# 属性について

HTMLでは、要素に**属性**という追加の情報をつけられます。属性はHTMLの開始タグの中に属性="値"という形で記述します。属性は複数つけることもできます。

**例**

```
<a href="index.html">リンク</a>
```

```

```

# HTMLの基本的なタグ

- <html> : このタグで囲んでいる部分がHTMLであることを表します
- <head> : HTMLの設定などを記述します
- <title> : Webページのタイトルを記載します。headの中で使用します。
- <header> : Webページのヘッダーを表します。headと混同しないようにしましょう
- <main> : webページのメインコンテンツを表します
- <footer> : webページのフッターを表します
- <body> : 実際に表示される部分です
- <h1> ~ <h6> : 見出しを表します
- <p> : 段落を表します
- <ul> ・ <ol> : リストを表します。ulは順不同のリスト、olは順番があるリストです。
- <li> : リストの項目を表します。必ずulやolの中に入ります。
- <img> : 画像を表します。閉じタグなし。src属性は必須、alt属性はほぼ必須です。
- <a> : リンクを表します。行先はhref属性です。
- <br> : 改行します。閉じタグなし。
- <div> : 特定の意味を持たず、レイアウトのために使用します
- <table> : 表を作成します
- <tr> : 表の横1行を表します
- <th> : 表の見出しセルです
- <td> : 表の通常セルです
- <span> : 特定の意味を持たないタグです。inline表示なので1行の中で使用します

**これ全部覚えておけば  
基本大丈夫！**

下の画像では、「見出しタグ (<h2>)」「段落タグ (<p>)」  
「画像タグ (<img>)」を用いています。

### テキストエディタ (VS コード)

```
<> index.html > html
1  <!doctype html>
2  <html>
3
4  <head>
5  <meta charset="utf-8">
6  <meta name="viewport" content="width=device-width,initial-scale=1.0" />
7  <meta name="description" content="このページの説明が入ります" />
8  <link rel="stylesheet" href="css/reset.css" type="text/css" media="all" />
9  <link rel="stylesheet" href="css/style.css" type="text/css" media="all" />
10 <title>タイトルが入ります</title>
11 </head>
12
13 <body>
14 <h2>ABOUT</h2>
15 <p>わたしたちについて</p>
16 <p>本文が入ります</p>
17 <p>LUMINA BEAUTYは、<br>
18 美容サロン運営・化粧品開発・スクール事業を通じて、<br>
19 人々の「美しくありたい」という想いを<br>
20 サポートしています。</p>
21 
22 </body>
23
24 </html>
```

### ブラウザの表示

ABOUT

わたしたちについて

本文が入ります

LUMINA BEAUTYは、

美容サロン運営・化粧品開発・スクール事業を通じて、  
人々の「美しくありたい」という想いを  
サポートしています。



# HTMLファイルについて

HTMLのファイル形式...**.html(主流)** or .htm

webサイトで最初に表示されるトップページのファイル名だけは必ず**index.html**にします。2ページ目以降は自由につけて構いませんが、以下ルールがあります。

- 半角英数字
- ハイフン (-) 、 アンダーバー ( \_ )
- スペースは使用できません

※画像のファイル名も同様のルール。わかりやすい名前にするのがポイント！

## CSSとは...

CSS（シー・エス・エス）とは「Cascading Style Sheets」の略称で、ホームページのスタイル（見た目）を装飾するときに用いるスタイルシート言語です。

基本的には、HTML内の「どれ」に対して「どの部分」を「どうする」かを指定していきます。

例

```
p {
  color: #000;
  font-size: 16px;
}
```

# CSSを書く位置

1. HTMLファイルの<head>要素内
2. HTMLの開始タグ内
3. **HTMLとは別のCSSファイル(主流)**

別のCSSファイルに記述する場合は以下のようなコードをHTMLのhead内に記載してCSSファイルを読み込みます

```
<link rel="stylesheet" href="css/style.css" >
```

# CSSのファイル名

一般的には**style.css**というファイル名にすることが多い

## 要素に名前をつける「class」

例 HTML

```
<div class="section-heading about__heading">
  <h2>ABOUT</h2>
  <p>わたしたちについて</p>
</div>
```

CSS

```
.section-heading h2 {
  color: #806E47 ;
  font-size: 64px ;
}
```

**一つの要素に  
複数のクラスをつける  
こともできます！**

# 親要素と子要素

例 HTML

```
<div class="section-heading about__heading">  
  <h2>ABOUT</h2>  
  <p>わたしたちについて</p>  
</div>
```

section-headingが**親要素**、その中のh2・pが**子要素**

# よく使われるCSSのプロパティ

- color : テキストの色を指定します
- font-size : フォントのサイズを指定します
- font-family : フォントの種類を指定します
- text-align : テキストの水平方向の配置を指定します
- background-color : 背景色を指定します
- width : 要素の幅を指定します
- height : 要素の高さを指定します
- margin : 要素の外側の余白を指定します
- padding : 要素の内側の余白を指定します
- border : 要素の境界線を指定します
- display : 要素の表示形式（ブロック、インラインなど）を指定します

**他にも色々あるよ！  
全て覚えるのは無理だから  
調べる力をつけよう**

# WORK

Q このデザインはどんなタグで表現できる？

## SERVICE

— サービス



### サロン事業

心と体を癒すトータルビューティーサロン。ヘア・ネイル・スキンケアまで幅広く対応し、お客様一人ひとりのライフスタイルに合わせたパーソナルな施術をご提供しています

[Read more](#)



### コスメ事業

肌にやさしいオリジナル化粧品を企画・開発。自然由来の成分にこだわり、敏感肌の方でも安心して毎日のケアに取り入れていただけるアイテムを展開。研究者と美容のプロが共同で開発を行っています。

[Read more](#)



### スクール事業

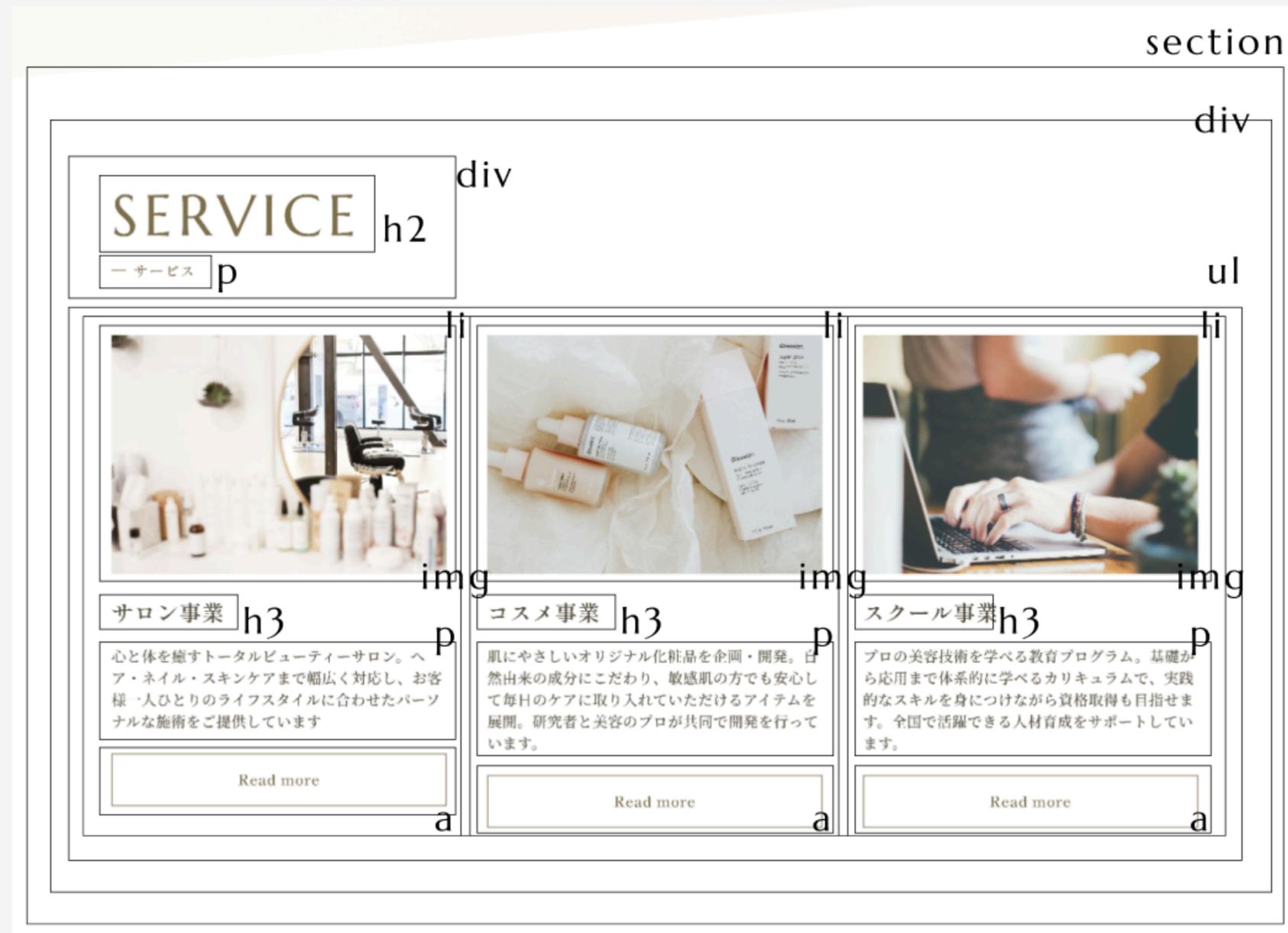
プロの美容技術を学べる教育プログラム。基礎から応用まで体系的に学べるカリキュラムで、実践的なスキルを身につけながら資格取得も目指せます。全国で活躍できる人材育成をサポートしています。

[Read more](#)

# WORK

A 右図は私ならこうします！  
という参考です

できるだけタグは少なく  
divを使いすぎないのが  
ポイントです



## テキストエディタ (VS コード)

## HTML

```

<> index.html > html
1 <!doctype html>
2 <html>
3
4 <head>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width,initial-scale=1.0" />
7 <meta name="description" content="このページの説明が入ります" />
8 <link rel="stylesheet" href="css/reset.css" type="text/css" media="all" />
9 <link rel="stylesheet" href="css/style.css" type="text/css" media="all" />
10 <link rel="preconnect" href="https://fonts.googleapis.com">
11 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
12 <link
13 href="https://fonts.googleapis.com/css2?family=Marcellus&family=Noto+Ser
14 rel="stylesheet">
15 <title>タイトルが入ります</title>
16 </head>
17
18 <body>
19 <section class="about">
20 <div class="about_inner inner">
21 <div class="about_content">
22 <div class="section-heading">
23 <h2>ABOUT</h2>
24 <p>わたしたちについて</p>
25 </div>
26 <p class="about_text">LUMINA BEAUTYは、<br>
27 美容サロン運営・化粧品開発・スクール事業を通じて、<br>
28 人々の「美しくありたい」という想いを<br>
29 サポートしています。</p>
30 </div>
31 <div class="about_image">
32 
34 </div>
35 </section>
36
37 </body>
38
39 </html>

```

## CSS①

```

css > # style.css > .inner
41
42 .inner {
43   width: 84%;
44   margin: 0 8%;
45 }
46
47 .section-heading {
48   margin-bottom: 80px;
49 }
50
51 .section-heading h2 {
52   font-size: 64px;
53   font-weight: 400;
54   font-family: "Marcellus", serif;
55   letter-spacing: 3px;
56   margin-bottom: 20px;
57   line-height: 1;
58 }
59
60 .section-heading p {
61   font-family: "Shippori Mincho B1", serif;
62   line-height: 1;
63   display: flex;
64   align-items: center;
65 }
66
67 .section-heading p::before {
68   content: '';
69   display: block;
70   width: 16px;
71   height: 1px;
72   background: #564E3D;
73   margin-right: 8px;
74 }
75
76 /* -----
77 about
78 ----- */
79 .about {
80   background-image: url(/images/about-bg.jpg);
81   padding: 180px 0;
82 }
83

```

## CSS②

```

css > # style.css > .inner
75
76 /* -----
77 about
78 ----- */
79 .about {
80   background-image: url(/images/about-bg.jpg);
81   padding: 180px 0;
82 }
83
84 .about_inner {
85   display: flex;
86   flex-direction: row-reverse;
87   justify-content: space-around;
88 }
89
90 .about_content {
91   width: 33%;
92 }
93
94 .about_image {
95   width: 42%;
96 }
97
98 .about_text {
99   line-height: 3;
100 }
101
102 /* @media screen and (min-width: 1024px) {} */
103
104 /* -----
105 sec02
106 ----- */
107
108 /* @media screen and (min-width: 1024px) {} */
109
110 /* -----
111 sec03
112 ----- */
113
114 /* @media screen and (min-width: 1024px) {} */
115
116 /* -----
117

```

### お手本デザイン



### ブラウザの表示



1. コーディングとは
2. HTML/CSS
- 3. JSとjQuery**
4. 動的/静的サイト
5. コーディングの手順
6. VSコードの使い方
7. コーディングのコツ
8. コーディング実況

# O3

# JSとjQuery

JavaScript（ジャバスクリプト）は、主にホームページで動きを付けるときに使用するプログラミング言語です。略称で、JS（ジェーエス）と呼ばれます。

## jQueryとは？JSとの違い

JavaScriptは「プログラミング言語」で、jQueryは「JavaScriptを簡単に使うためのライブラリ」です

## jQueryで実装できること

- フェードイン（ふわっと出現）
- ハンバーガーメニュー
- アコーディオンメニュー（FAQでよく使われている）
- スライダー（メインビジュアルやカードデザインで）
- タブ切り替え（料金表やプランなどで）

## JSのファイル名

一般的には**script.js**というファイル名にすることが多い

## jQueryのCDNとJSファイルの読み込み

基本的には「`</body>` の直前」に以下2つのコードを書く

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"
  integrity="sha256-/JqT3SQfawRcv/BIHPThkBvs00EvtFFmqPF/IYI/Cxo="
  crossorigin="anonymous"> </script>
```

```
<script src="js/script.js"> </script>
```

CDN：【公式サイト】 <https://releases.jquery.com/>

1. コーディングとは
2. HTML/CSS
3. JSとjQuery
4. **動的/静的サイト**
5. コーディングの手順
6. VSコードの使い方
7. コーディングのコツ
8. コーディング実況

A large, bold, black outline of the number '04' is centered on the right side of the slide. The '0' is a simple circle, and the '4' is a stylized, blocky digit.

## 静的サイトと動的サイトの違い

- 静的サイト
  - 「HTML」や「CSS」と呼ばれる言語を用いて制作される
  - 「誰が」「いつ」「どこで」見ても同じ情報が表示されるホームページ
- 動的サイト
  - 「WordPress」を用いて制作されることが多い
  - アクセスした「ユーザー」「時間」「場所」といった特定の条件に応じて、異なる情報を自動で表示するホームページ
  - (例)お知らせ、お問い合わせフォーム、予約フォーム、掲示板、サイト内検索等

1. コーディングとは
2. HTML/CSS
3. JSとjQuery
4. 動的/静的サイト
- 5. コーディングの手順**
6. VSコードの使い方
7. コーディングのコツ
8. コーディング実況

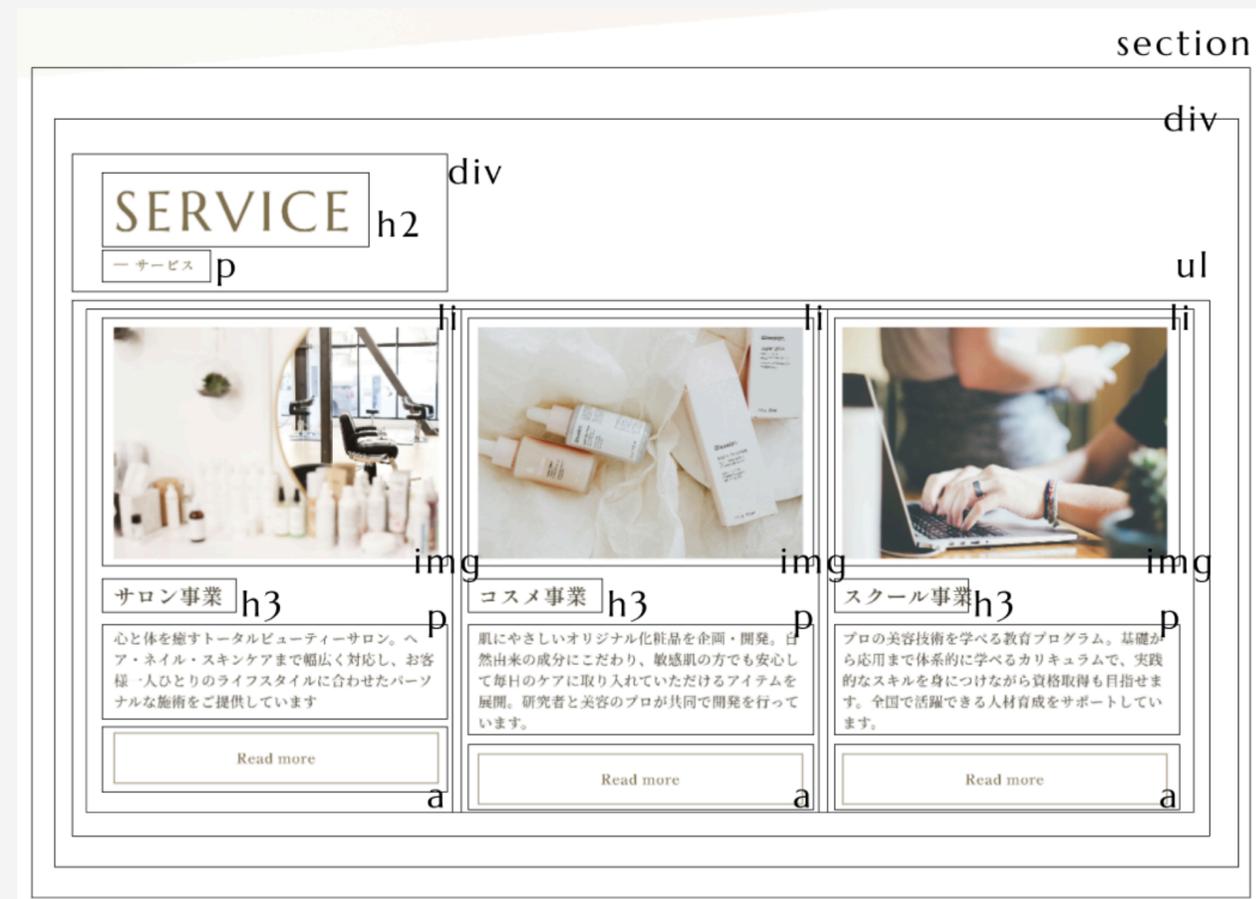
# O5

## コーディングに使うツール

- デザインデータ (**Figma**, Photoshop, Illustratorなど)
- テキストエディター (**Visual Studio Code**, Atomなど)
- WEBブラウザ (Chrome, Safari, FireFox)
- Local (WordPressの構築に)

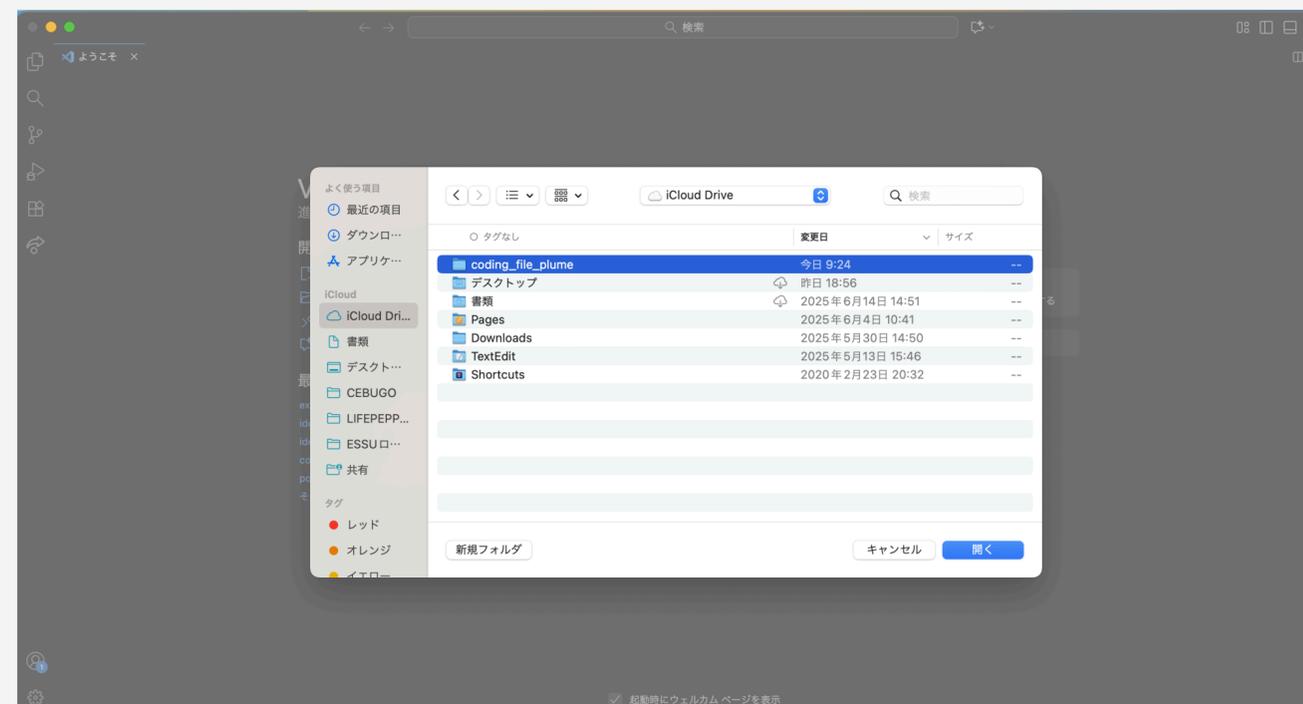
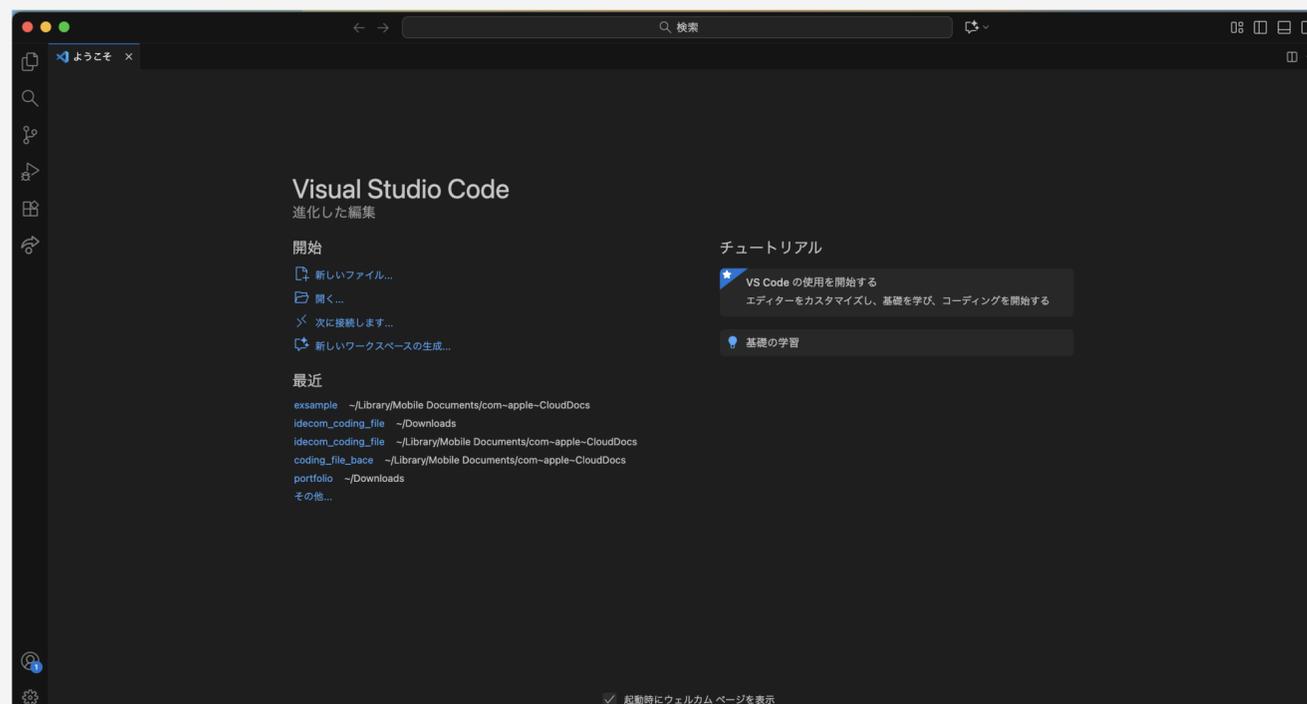
# ① デザインデータを用意

まずはデザインデータを用意します。  
初心者のうちはいきなりコーディング  
するのではなく  
タグのマークアップをしてからコード  
を書くことをお勧めします。



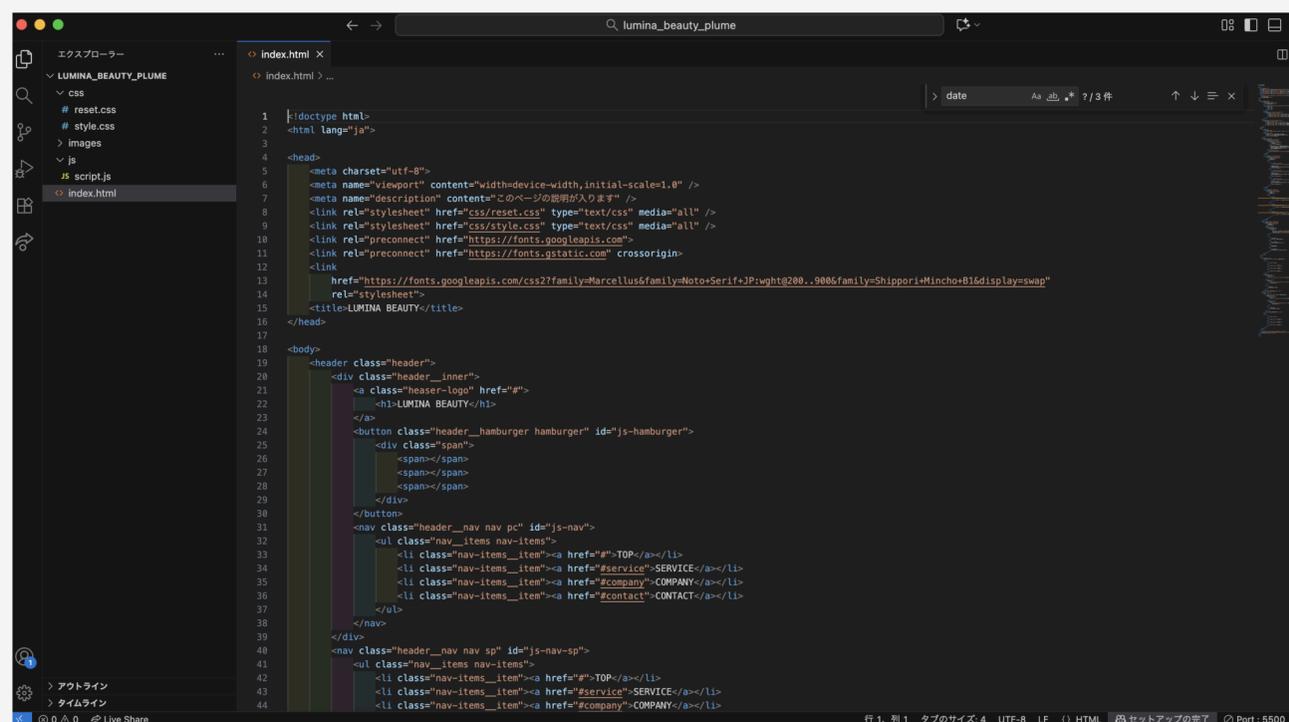
## ②VSコードでファイルを開く

コーディングファイルを開きます



## ③ デザインデータを見ながらHTMLを書く

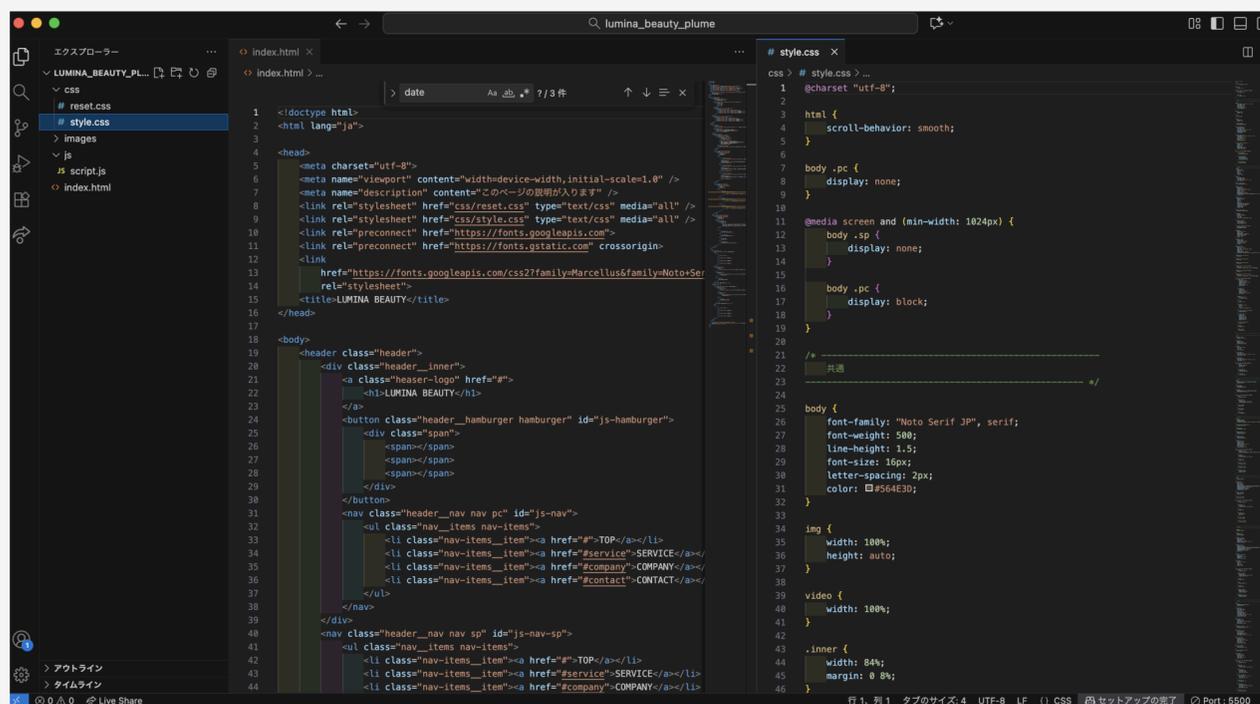
①のデザインデータを見ながらindex.htmlにHTMLを書いていきます



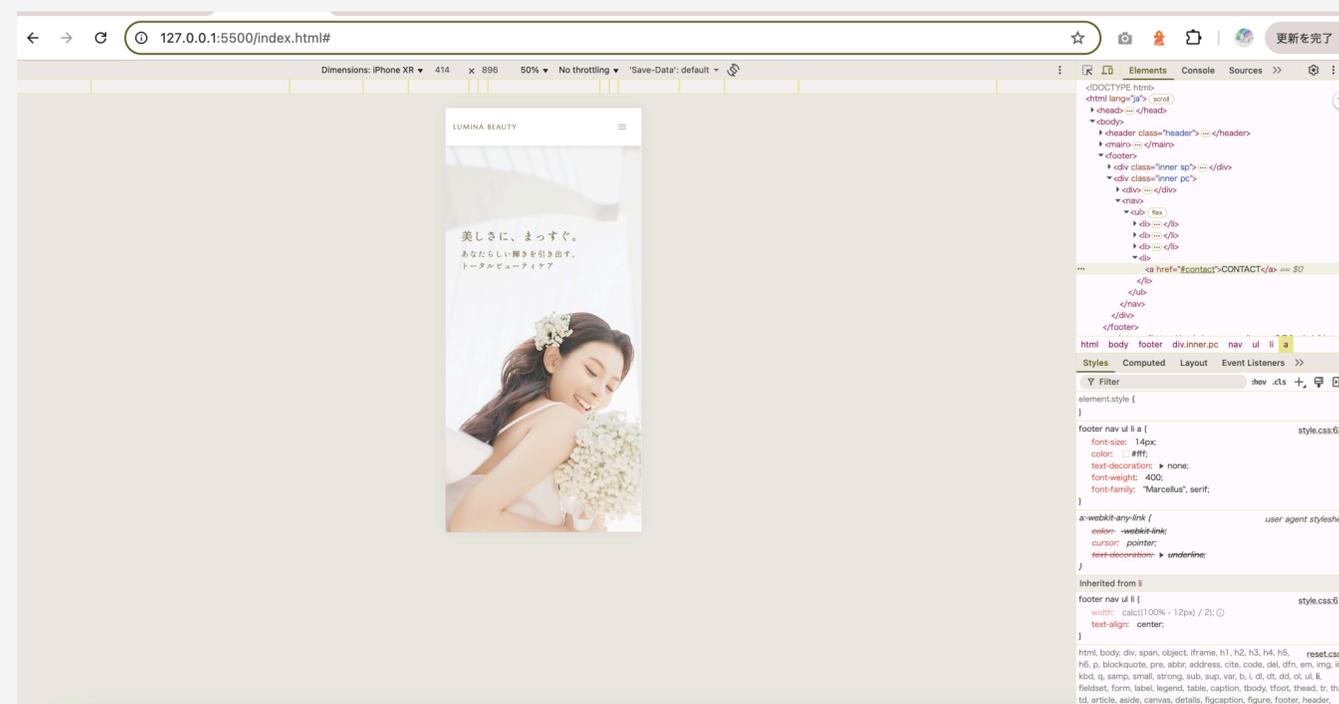
```
1 <!doctype html>
2 <html lang="ja">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width,initial-scale=1.0" />
7   <meta name="description" content="このページの説明が入ります" />
8   <link rel="stylesheet" href="css/reset.css" type="text/css" media="all" />
9   <link rel="stylesheet" href="css/style.css" type="text/css" media="all" />
10  <link rel="preconnect" href="https://fonts.googleapis.com">
11  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin">
12  <link
13    href="https://fonts.googleapis.com/css2?family=Marcellus&family=Noto+Serif+JP:wght@200..900&family=ShipporiMincho&B1&display=swap"
14    rel="stylesheet">
15  <title>LUMINA BEAUTY</title>
16 </head>
17
18 <body>
19   <header class="header">
20     <div class="header__inner">
21       <a class="header__logo" href="#">
22         <h1>LUMINA BEAUTY</h1>
23       </a>
24       <button class="header__hamburger hamburger" id="js-hamburger">
25         <div class="span">
26           <span></span>
27           <span></span>
28           <span></span>
29         </div>
30       </button>
31       <nav class="header__nav nav pe" id="js-nav">
32         <ul class="nav__items nav-items">
33           <li class="nav-items__item"><a href="#">TOP</a></li>
34           <li class="nav-items__item"><a href="#service">SERVICE</a></li>
35           <li class="nav-items__item"><a href="#company">COMPANY</a></li>
36           <li class="nav-items__item"><a href="#contact">CONTACT</a></li>
37         </ul>
38       </nav>
39     </div>
40     <nav class="header__nav nav sp" id="js-nav-sp">
41       <ul class="nav__items nav-items">
42         <li class="nav-items__item"><a href="#">TOP</a></li>
43         <li class="nav-items__item"><a href="#service">SERVICE</a></li>
44         <li class="nav-items__item"><a href="#company">COMPANY</a></li>
45       </ul>
46     </nav>
47   </header>
48
```

## ④ デザインデータを見ながらCSSを書く

①のデザインデータを見ながらstyle.cssにCSSを書いていきます  
ブラウザでプレビューや検証ツールを見ながらやるのがおすすめです

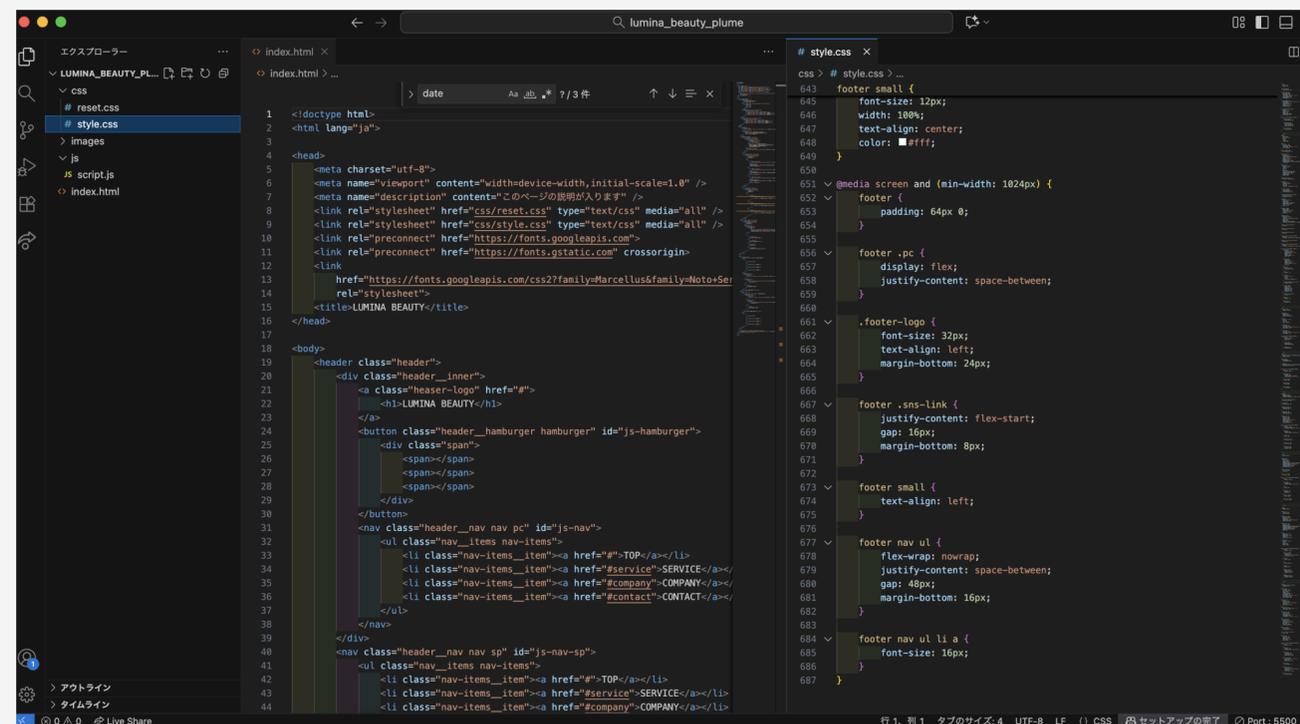


The screenshot shows a code editor with two files open: index.html and style.css. The index.html file contains HTML code for a website header, including a logo, a hamburger menu button, and a navigation menu with links for 'TOP', 'SERVICE', 'COMPANY', and 'CONTACT'. The style.css file contains CSS code for the header, including a media query for screens with a minimum width of 1024px, and styles for the body, header, and navigation items.



## ⑤ レスポンシブ

スマホか版からコーディングして、その後にPC版に変換していきます  
CSSのメディアクエリ **@media screen and (min-width: 1024px) {}** に  
PCのCSSを記載します



```
1 <!doctype html>
2 <html lang="ja">
3
4 <head>
5
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width,initial-scale=1.0" />
8 <meta name="description" content="このページの説明が入ります" />
9 <link rel="stylesheet" href="css/reset.css" type="text/css" media="all" />
10 <link rel="stylesheet" href="css/style.css" type="text/css" media="all" />
11 <link rel="preconnect" href="https://fonts.googleapis.com">
12 <link
13 href="https://fonts.googleapis.com/css2?family=Marcellus&family=NotoSerif"
14 rel="stylesheet">
15 <title>LUMINA BEAUTY</title>
16 </head>
17
18 <body>
19 <div class="header">
20 <div class="header_inner">
21 <a class="header-logo" href="#">
22 <h1>LUMINA BEAUTY</h1>
23 </a>
24 <button class="header_hamburger hamburger" id="js-hamburger">
25 <div class="span">
26 <span></span>
27 <span></span>
28 <span></span>
29 </div>
30 </button>
31 <nav class="header__nav nav pc" id="js-nav">
32 <ul class="nav_items nav-items">
33 <li class="nav-items_item"><a href="#">TOP</a></li>
34 <li class="nav-items_item"><a href="#service">SERVICE</a></li>
35 <li class="nav-items_item"><a href="#company">COMPANY</a></li>
36 <li class="nav-items_item"><a href="#contact">CONTACT</a></li>
37 </ul>
38 </nav>
39 </div>
40 <nav class="header__nav nav sp" id="js-nav-sp">
41 <ul class="nav_items nav-items">
42 <li class="nav-items_item"><a href="#">TOP</a></li>
43 <li class="nav-items_item"><a href="#service">SERVICE</a></li>
44 <li class="nav-items_item"><a href="#company">COMPANY</a></li>
45 </ul>
46 </nav>
47 </div>
48 </body>
49 </html>
```

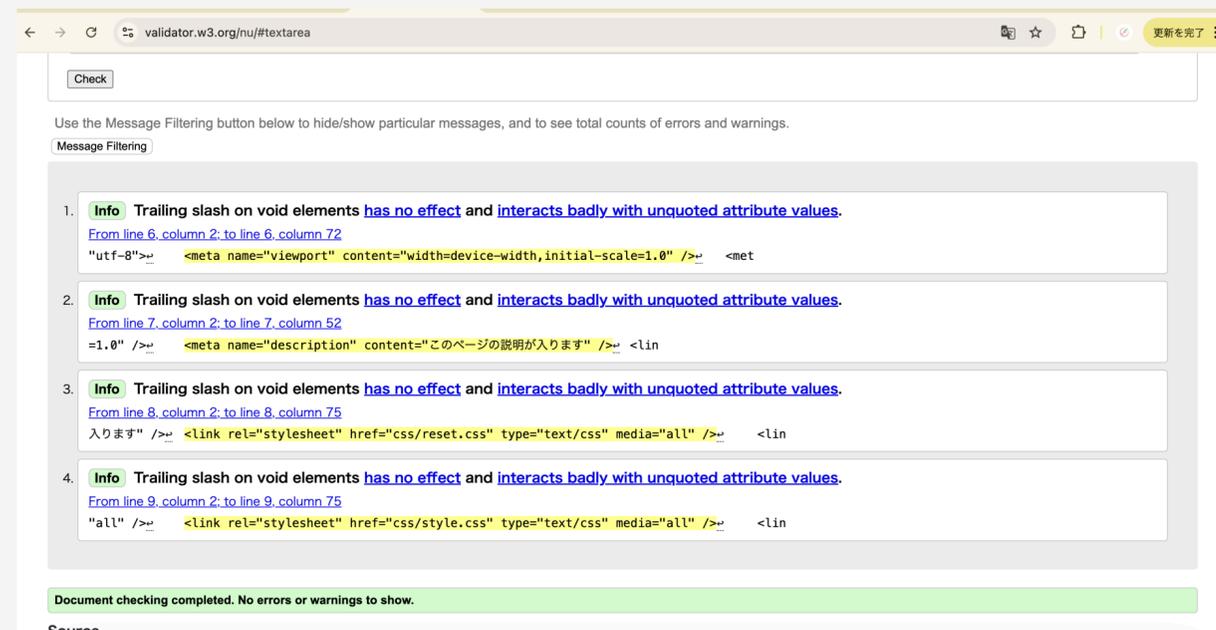
```
643 footer small {
644 font-size: 12px;
645 width: 100%;
646 text-align: center;
647 color: #fff;
648 }
649
650
651 @media screen and (min-width: 1024px) {
652 footer {
653 padding: 64px 0;
654 }
655
656 footer_pc {
657 display: flex;
658 justify-content: space-between;
659 }
660
661 footer-logo {
662 font-size: 32px;
663 text-align: left;
664 margin-bottom: 24px;
665 }
666
667 footer_sns-link {
668 justify-content: flex-start;
669 gap: 16px;
670 margin-bottom: 8px;
671 }
672
673 footer_small {
674 text-align: left;
675 }
676
677 footer_nav ul {
678 flex-wrap: nowrap;
679 justify-content: space-between;
680 gap: 48px;
681 margin-bottom: 16px;
682 }
683
684 footer_nav ul li a {
685 font-size: 16px;
686 }
687 }
```

## ⑥ エラーチェック

全てのコードが書けたら、コードのエラーチェックをします。

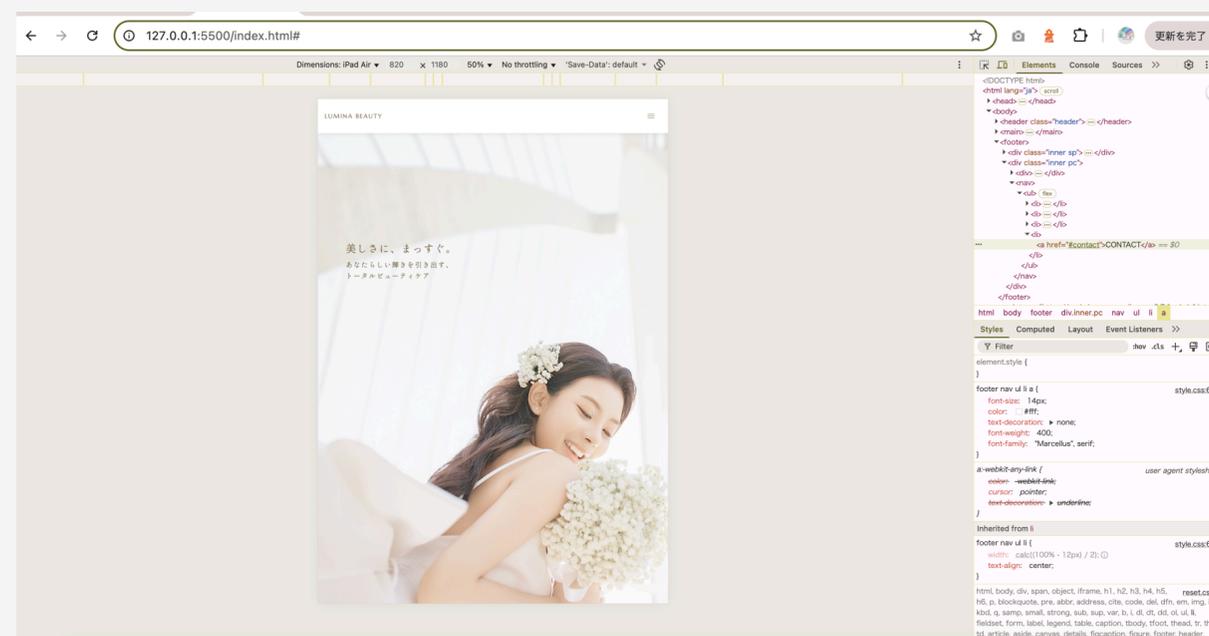
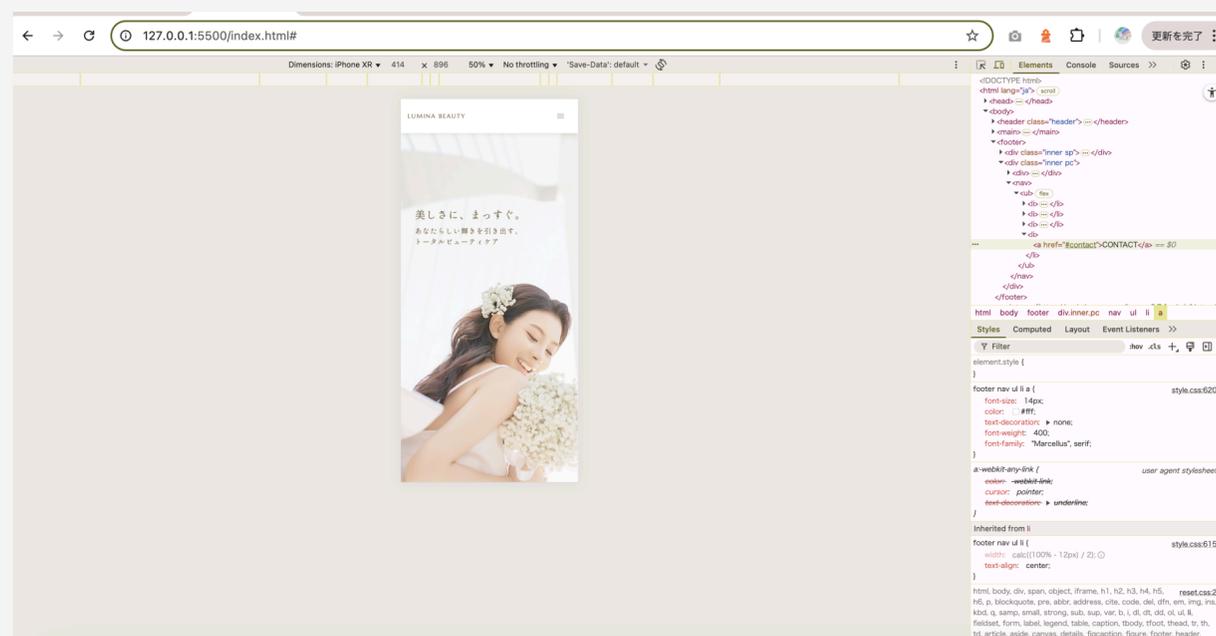
HTML : [https://validator.w3.org/#validate\\_by\\_input](https://validator.w3.org/#validate_by_input)

CSS : [https://jigsaw.w3.org/css-validator/#validate\\_by\\_input](https://jigsaw.w3.org/css-validator/#validate_by_input)



## ⑦ ブラウザごとの表示チェック

各ブラウザ（Chrome,Safari,FireFox）ごとに表示エラーがないかを確認します。画面幅ごとのエラーも確認しましょう。  
この時mac,windows,iPhone,Andoroidでも確認しましょう。



## ⑧納品

エラー・表示に問題がなければ初回の納品！  
この後修正があれば修正対応をしましょう。

### 納品方法

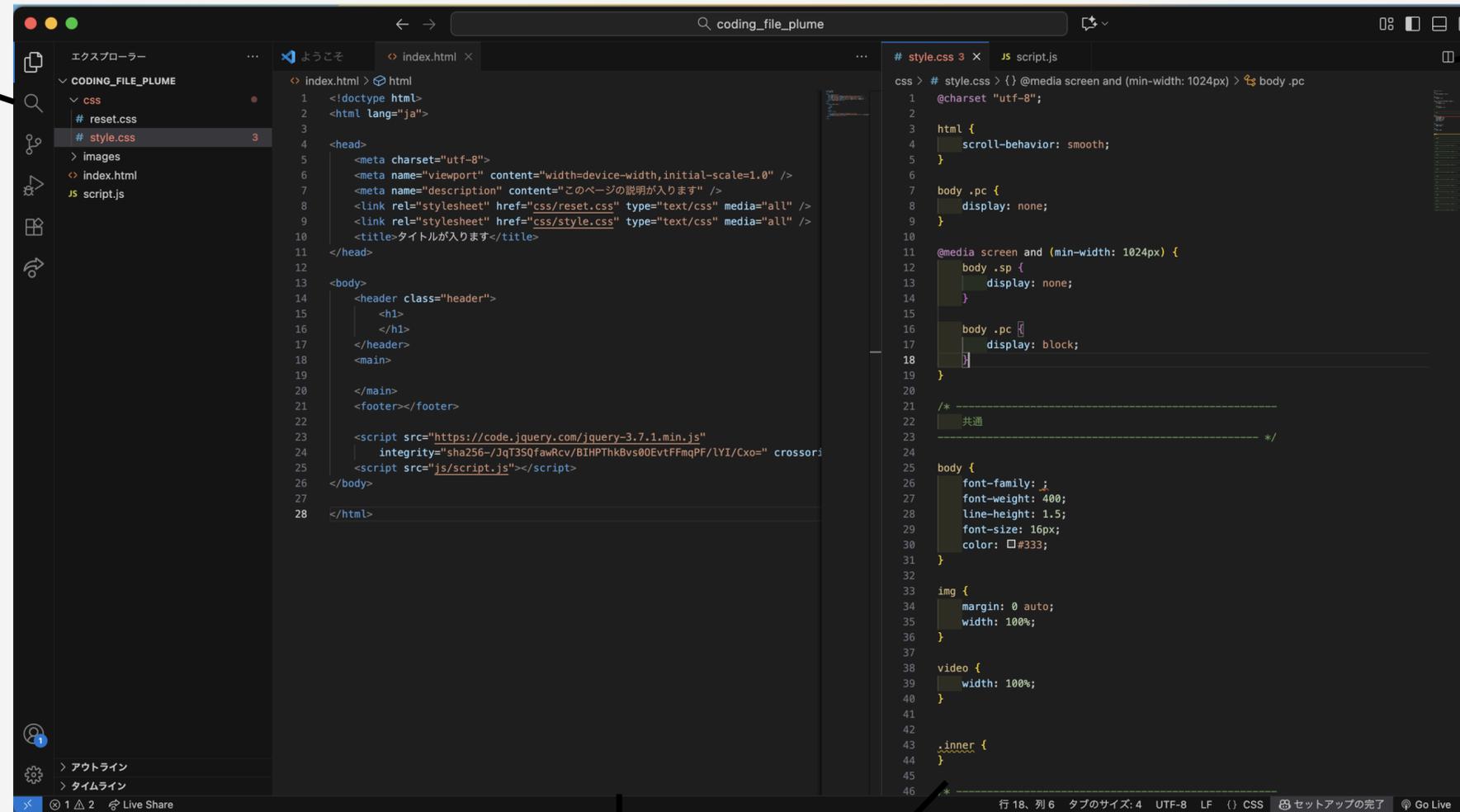
- ファイル納品
- テストサーバーにアップ

1. コーディングとは
2. HTML/CSS
3. JSとjQuery
4. 動的/静的サイト
5. コーディングの手順
- 6. VSコードの使い方**
7. コーディングのコツ
8. コーディング実況



# VSコードの基本表示

左側にファイルの中身が表示されます  
選択すると右側に表示されます



このアイコンを選択すると  
コードを横並びにできます

左にindex.html、右にstyle.cssやscript.jsを配置すると  
コードが書きやすいです

Go Liveを押すと  
ブラウザ上でのプレビューを  
見れます

# 自動インデント設定

**Command + S (WindowsはCtrl + S)** でインデントを整えることができる

【設定方法】

**Mac** : Code → 基本設定 → 設定

**Windows** : ファイル → ユーザー設定 → 設定

設定画面で、検索欄に下記の項目をコピーで入力し、出てきたそれぞれの項目にチェックを入れておきましょう。

- Editor : Format On Paste
- Editor : Format On Save
- Editor: Format On Type

1. コーディングとは
2. HTML/CSS
3. JSとjQuery
4. 動的/静的サイト
5. コーディングの手順
6. VSコードの使い方
- 7. コーディングのコツ**
8. コーディング実況



# Emmet

VSコードではEmmetという機能が初めから使えるようになっていています。Emmetを活用すると、HTML・CSSのコーディングを**素早く行うことができます**。

## HTML

- h2 + tabキー ⇒ <h2></h2>
- p + tabキー ⇒ <p></p>
- a + tabキー ⇒ <a href="">
- div + tabキー ⇒ <div></div>
- ul + tabキー ⇒ <ul></ul>
- li + tabキー ⇒ <li></li>
- img + tabキー ⇒ <img src="">

**タグの<は入力しなくて大丈夫！**

**spanなど文字の途中に入れるもののはうまくいかないこともあります。**

## CSS

- tac + tabキー ⇒ text-align: center
- tar + tabキー ⇒ text-align: right
- fz16 + tabキー ⇒ font-size: 16px
- fw700 + tabキー ⇒ font-weight: 700
- lh1.5 + tabキー ⇒ line-height: 1.5
- c + tabキー ⇒ color: #000
- m0 + tabキー ⇒ margin: 0
- mr10 + tabキー ⇒ margin-right: 10px
- pl10 + tabキー ⇒ padding-left: 10px
- df + tabキー ⇒ display: flex
- jcc + tabキー ⇒ justify-content: center
- ali + tabキー ⇒ align-items: center

# 検証ツール

Google Chromeデベロッパーツールの起動方法は、Google Chromeで対象のWebサイトを開いた状態で**右クリック**をして「**検証**」をクリックするだけです。

デバイスごとの表示を確認したり、エラーを確認したりすることができます。



←検証ツール

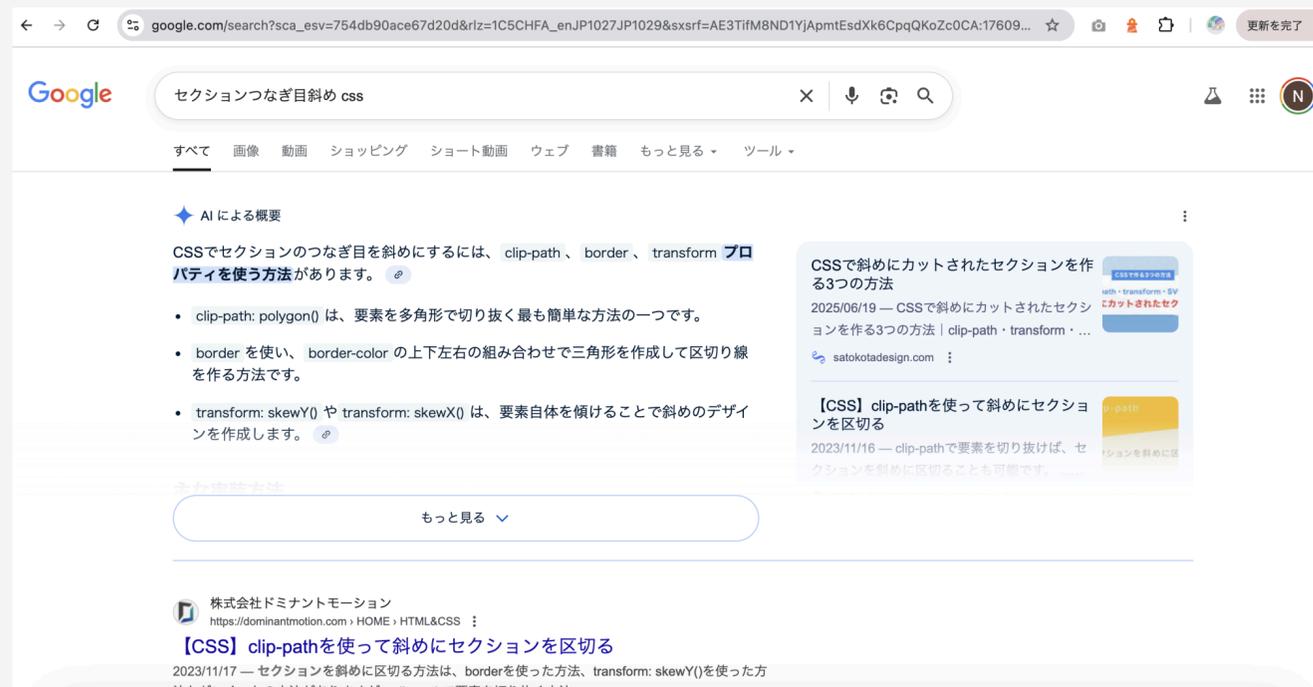
# 調べる

例えば...右のデザイン、セクションの繋ぎ目が斜めになっています。これってどうやってコーディングするのか？と最初はわかりませんよね。

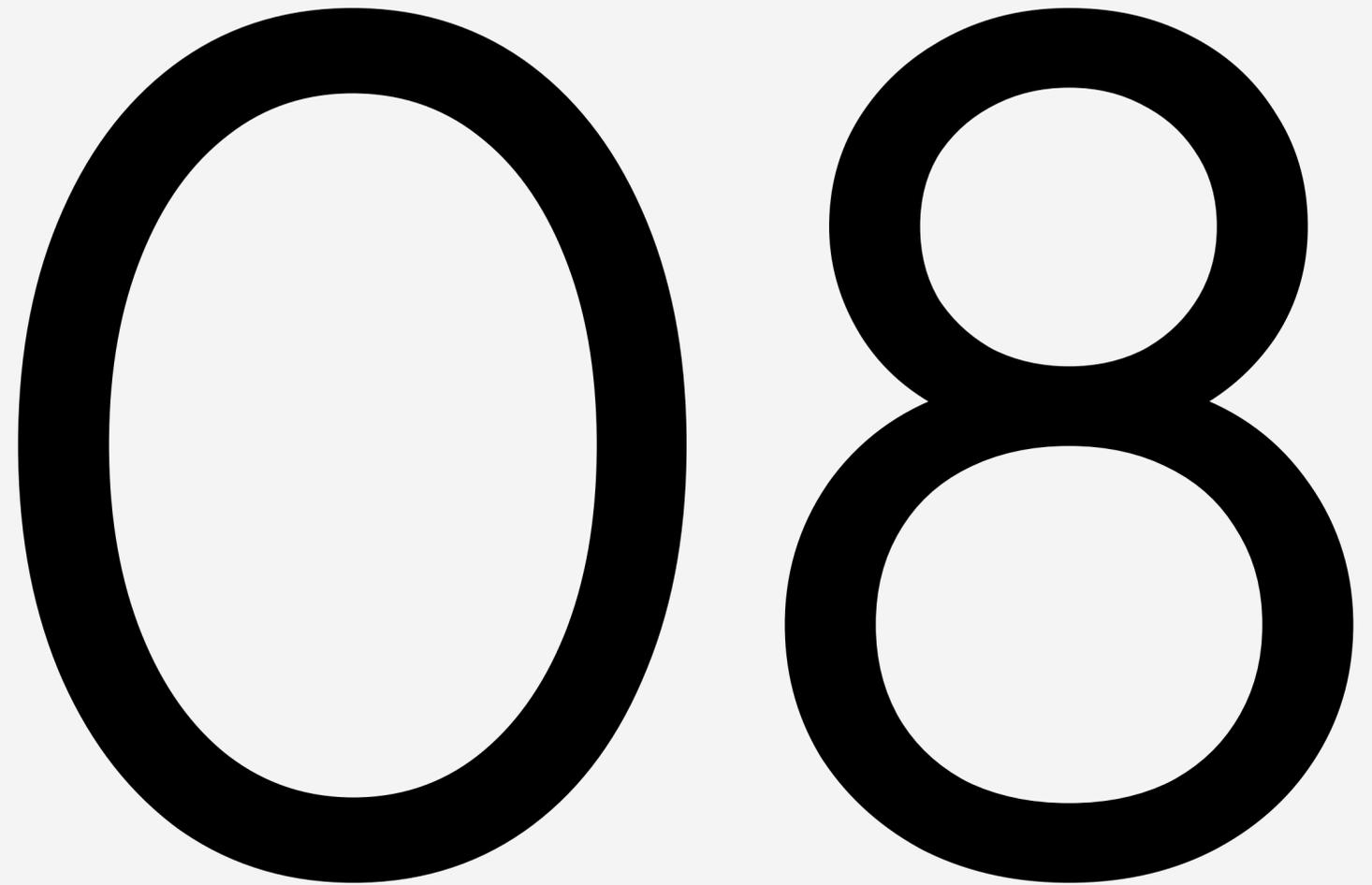


# 調べる

『セクションつなぎ目斜め css』と検索すると、  
答えが書かれてるサイトが出てきました。



1. コーディングとは
2. HTML/CSS
3. JSとjQuery
4. 動的/静的サイト
5. コーディングの手順
6. VSコードの使い方
7. コーディングのコツ
8. コーディング実況



**実際にコーディングするところを見てみましょう**