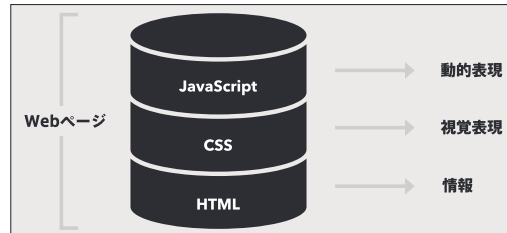


0-1 Webページが表示される仕組み

• Webページはコンピュータ言語で作られている

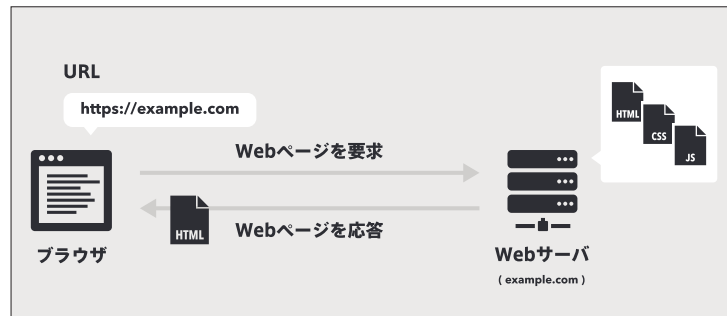
普段、みなさんが見ているWebページは、HTMLやCSS、JavaScriptなどのコンピュータ言語を使って制作されています。文章や表などのデータは、**HTML**という言語を使うことでWebページに表示することができます。そして、そのデータに**CSS**という言語を使って、レイアウトや装飾を工夫することで、いわば服を着せたりお化粧したりするように、華やかに見せることができます。さらに、**JavaScript**というプログラミング言語を使って動きをつけることでWebページが完成します。つまり、現在主流のWebページは、どれか1つの言語で作られているわけではなく、**HTMLやCSS、JavaScriptといったそれぞれ担当部門の異なる言語を、必要に応じて組み合わせることによって制作されています。**



本書ではその中のHTMLにのみフォーカスして解説するよ。

• 世界中に公開するにはWebサーバが必要

制作したWebページは、**Webサーバ**と呼ばれるインターネット上のコンピュータに保存することで、世界中に公開することができます。ユーザがWebページを表示するには、ブラウザと呼ばれるWebページを閲覧するためのソフトを用いて、Webページの保存場所を表すアドレス(URL)を指定します。そして、指定された場所にあるWebページのデータがWebサーバからブラウザに送信されることで、Webページを閲覧できるようになります。



作ったページを公開するにはWebサーバに保存する必要があるんだね。

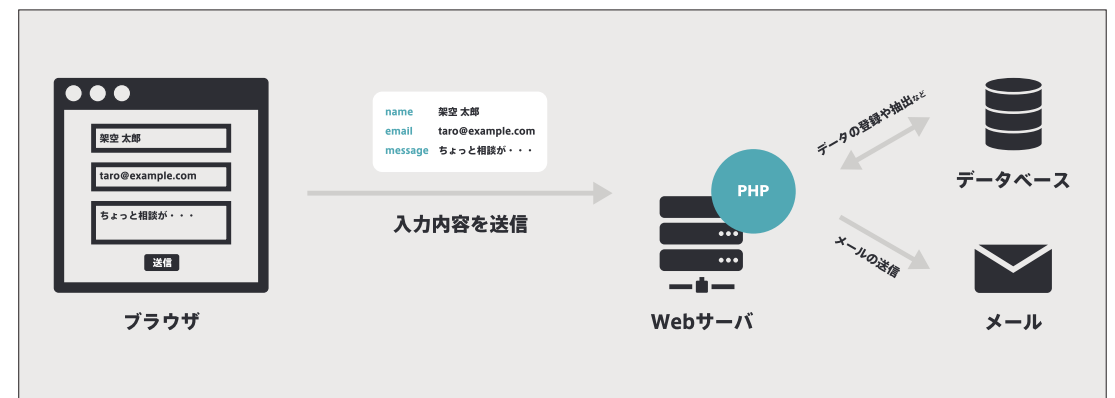
Keyword

Webサーバ

Webページ(HTML、CSS、JavaScriptなどにより制作されたファイル)が保存されているインターネット上の倉庫のようなもの。ユーザが倉庫の場所(URL)を指定すると、保存されているWebページのデータがブラウザに届く。

• ブログやお問い合わせフォームなどのシステムには、ほかの言語も必要

Webサイトによっては、HTMLやCSS、JavaScriptのみでは作れないものもあります。具体的には、ブログやSNSのような、ユーザが記事やメッセージを投稿するタイプのWebサイトなどが挙げられます。こういったWebサイトは、投稿された記事やメッセージをデータベースと呼ばれる場所に保存しておくことで、情報をいつでも閲覧できるようにしています。しかし、**HTMLやCSSではデータベースとの相互のやりとりを行うことはできません。**送信したデータをWebサーバ側で処理するためのPHPなどのプログラミング言語や、SQLといったデータベース言語などが必要になります。また、お問い合わせフォームなどの仕組みも同様で、HTMLで入力フォーム自体を作ることはできますが、送信されたデータをユーザやサイト管理者に届ける際は、PHPなどのWebサーバ側のプログラミング言語が必要になります。**HTMLやCSS、JavaScriptは主にWebページの表示を担当する言語**であり、メールの送信やデータベースを必要とするようなWebサイトを構築する際は、PHPやSQLなどと組み合わせる必要があることを覚えておきましょう。



Keyword

コンピュータ言語

コンピュータ上で用いられる言語の総称。コンピュータ言語の中には、HTMLのようなマークアップ言語やJavaScriptのようなプログラミング言語などがある。

プログラミング言語

コンピュータ言語の中でも、コンピュータに処理を伝えるプログラムを記述するための言語。Webサイトの制作では、JavaScriptやPHP、Rubyなどのプログラミング言語が用いられることが多い。

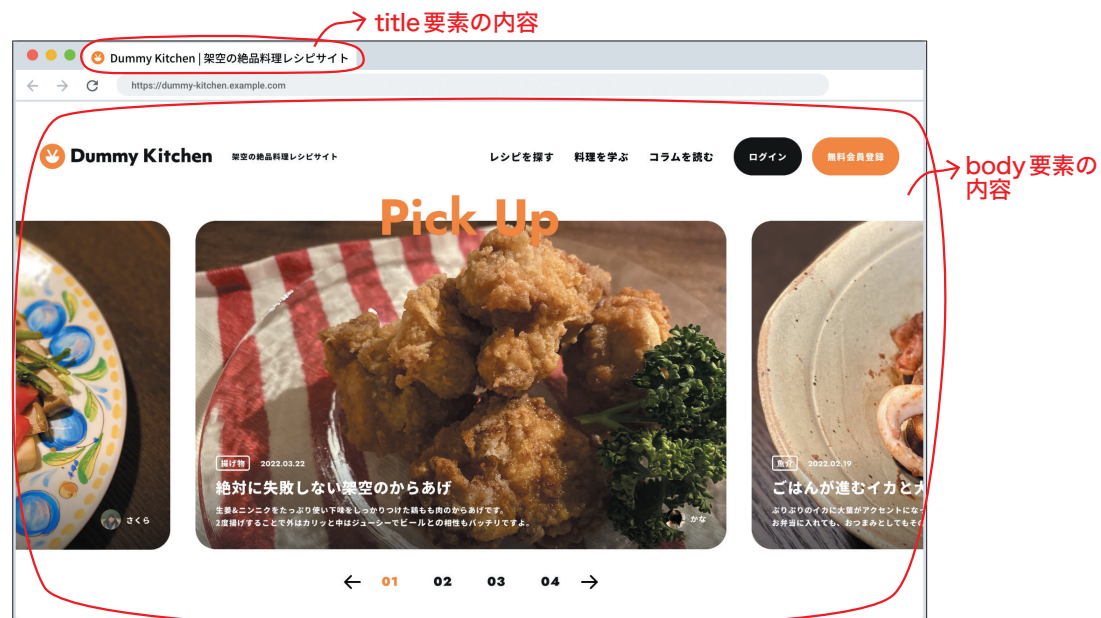
HTMLの全体構造の要素

いよいよ、Webページに表示する文章に対して適切なタグをつけていきます。ここでは、ページの骨格を作る要素を学習し、どのHTMLにも共通して行う基本のマークアップを覚えましょう。

▶ Step 1 全体構造を表す要素を知ろう

HTMLでは、Webページに表示する文章のみを記述するわけではありません。ブラウザに文章構造を伝え、Webページとして表示させるためには、最初に構造の枠組みを作る必要があります。

はじめに、HTMLドキュメントであることを表す **html** 要素を配置し、その中にほかのすべての要素を配置していきます。**html** 要素はすべての要素を包む根源的な要素になるので、「ルート要素」と呼ばれることもあります。**html** 要素の中には、まず、**head** 要素と **body** 要素を配置します。**head** 要素で囲む範囲内は**基本的には表示されない部分**となり、ページ自体の情報や設定に関する要素を配置します。その1つにページタイトルを表す **title** 要素があります。**title** 要素の内容は、ページ内ではなくブラウザのタブや検索エンジンの検索結果画面に表示されます。反対に、ページ内に表示させたい文章や画像などは、すべて **body** 要素の中に配置します。見えない情報は **head** 要素、見せたい内容は **body** 要素で囲むと覚えましょう。



▶ Step 2 全体構造を表す要素の使い方を知ろう

それでは、全体構造を表す要素の使い方を学習していきましょう。最初に、**html** 要素を配置します。続いて、その中に **head** 要素、**body** 要素の順に配置します。さらに **head** 要素の中にページのタイトルを記述する **title** 要素を配置します。最後に、**body** 要素の中にページの内容を記述します。

使い方 HTMLの全体構造

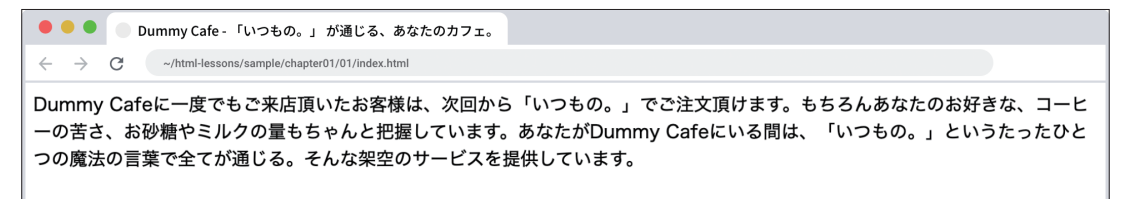
```
<html>
  <head>
    <title>ページのタイトル</title>
  </head>
  <body>
    ページの内容
  </body>
</html>
```

基本的には、この4つの要素がページを構築する上で必要となる基本の要素になります。なお、ページのタイトルは検索エンジンの検索結果ページにも表示されますので、**サイト名をはじめ、各ページの内容を正確に表した固有のページタイトル**をつけるようにしましょう。

サンプル 全体構造の要素を使った例

```
<html>
  <head>
    <title>Dummy Cafe - 「いつもの。」が通じる、あなたのカフェ。</title>
  </head>
  <body>
    Dummy Cafeに一度でもご来店頂いたお客様は、次回から「いつもの。」でご注文頂けます。もちろんあなたの好きな、コーヒーの苦さ、お砂糖やミルクの量もちゃんと把握しています。あなたがDummy Cafeにいる間は、「いつもの。」というたったひとつの魔法の言葉で全てが通じる。そんな架空のサービスを提供しています。
  </body>
</html>
```

上記のHTMLファイルをブラウザで開くと、以下のように表示されます。



このように、HTMLは**要素の中に要素を配置する入れ子構造**になります。そのため、すべてのコードを左端に揃えて書くと、入れ子構造が深くなった時に、どの要素がどの要素に挟まれているのかわかりにくくなります。先ほどのサンプルのように、半角スペースで**インデント**をつけてコードを見やすくするとよいでしょう。

HTMLのバージョンとDOCTYPE

HTMLにはバージョンがあり、さまざまな機能やルールが更新されてきました。ここでは、これまでのバージョンの特徴や、現在の最新の仕様であるHTML Standardについて学習しましょう。

▶ Step① HTMLのバージョンを知ろう

HTMLは、1990年代から幾度もの改訂を経て進化してきました。その改訂の度に、新たな要素や属性が追加・廃止されながら、最新の仕様がバージョンとして公開されてきました。**バージョンによって使える要素や属性、要素の配置ルールなどが異なります。**以下が、主なHTMLのバージョンです。

●主なHTMLバージョン

バージョン	概要
HTML 4.01	1999年に勧告されたバージョン。1997年に勧告されたHTML4.0より、HTMLでの視覚的表現を目的とする要素や属性を非推奨とし、視覚的な表現を行うための要素や属性をCSSという言語に移行した。その後、HTML4.01にマイナーアップデートされた
XHTML 1.0	2000年に勧告されたバージョン。XMLという別の言語の仕様をベースに作られたHTML。HTMLとしての大きな変更はないが、タグは小文字で記述するなど、ルールがより厳格になった。XHTMLの「X」は「エクステンシブル」の略で、拡張可能な意味
HTML5	2014年に勧告されたバージョン。人間にも機械にも、より理解しやすい言語となり、Webアプリケーションを開発するためのさまざまな仕様が新たに盛り込まれている。2016年にはHTML5.1、2017年にはHTML5.2とマイナーアップデートされた
HTML Standard	2021年に勧告された最新のHTML(執筆時点)。これまでのHTMLはW3CというWeb技術の標準化団体が策定した仕様がバージョンとして勧告されてきたが、Apple、Mozilla、Operaというブラウザベンダーの開発者らによって発足したWHATWGという団体によって独自に策定されたHTMLが、2021年1月に公式かつ唯一のバージョンとしてW3Cより勧告された。このHTMLにはバージョンがなく、日々、仕様がアップデートされている

■ Keyword

W3C (World Wide Web Consortium)

1994年に、Web技術の標準化を推進するために創設された団体。

WHATWG (Web Hypertext Application Technology Working Group)

Apple、Mozilla、Operaの開発者たちによって、2004年に結成された団体。現在のHTMLの仕様を策定している。

現在は**HTML Standardが最新のHTML**になります(執筆時点)。これは「HTML Living Standard」とも呼ばれ、従来のようにHTMLをバージョンごとに発表するのではなく、**日々、仕様をアップデートし、常に最新の仕様として公開**されています。本書もこのHTML Standardの仕様に基づいて執筆していますが、HTML Standardの仕様は常にアップデートされているため、時間の経過とともに執筆時点と仕様異なる部分が出てくることが考えられます。そのため、日頃から自身でも最新のHTML Standardを確認するのがよいでしょう。仕様書は英語で書かれていたり、専門用語が登場したりするので、HTMLの勉強を始めたばかりの人にはハードルが高いかもしれませんが、和訳しながら読み進めてみてください。

参考 HTML Standard

<https://html.spec.whatwg.org/>

▶ DOCTYPE宣言の目的

なぜ、ここでHTMLのバージョンの話をしているかというと、HTMLはこれらのバージョンによって利用できる要素や属性、記述のルールなどが異なるからです。これから新しく作るWebページであれば、HTML Standardの仕様に基づいて書けば問題ないのですが、もしそれ以前のバージョンで作られたWebページを修正することになった場合は注意が必要です。なぜなら、どのバージョンにも、段落を表す**p**要素や大見出しを表す**h1**要素などの基本的な要素は同じように存在しているため、一見どのバージョンに基づいて記述されているのか判断できない可能性があるためです。

そこで、**DOCTYPE宣言**が役立ちます。HTML5よりも前のバージョンでは、今記述している(X)HTMLが、どのバージョンの仕様に基づいて記述しているのかを、HTMLファイルの冒頭で宣言していました。具体的には、バージョンごとにDTDと呼ばれるファイルがW3Cによって公開されており、そのファイルには、そのバージョンで利用できる要素や属性の情報、どこに配置できるのかなどの情報が定義されています。このDTDをHTMLファイルの冒頭に指定することで、**どのバージョンの仕様に基づいて記述されたHTMLなのかを示す**ことができるのです。これを、**DOCTYPE宣言**(文書型宣言)といいます。

ですが、**HTML5よりも後のバージョンにはDTDがありません**。したがって理屈上は、HTML5以降ではDOCTYPE宣言は不要です。ただし、ほとんどのブラウザは、DOCTYPE宣言をブラウザの表示モードを切り替える用途に使っており、**DOCTYPE宣言があれば「標準モード」というモードで表示し、DOCTYPE宣言がなければ「互換モード」というモードで表示**します。互換モードの場合、仕様などに準拠していないWebページであると認識されてしまい、CSSが正しく解釈されずにレイアウトが崩れてしまう可能性があります。そのため、**互換モードで表示させないために、HTML5以降でも必要最低限のDOCTYPEの記述を行います**。

■ Keyword

DTD (Document Type Definition : 文書型定義)

HTMLのバージョンごとに、そのバージョンでどの要素や属性が利用できるのか、どこにどの順番で配置できるのか、などが定義されているファイル。

文字エンコーディングと文字化け

文字化けしてしまった、漢字や記号だらけのWebページを見たことがあるでしょうか？文字化けが起きる原因を知り、防げるようになります。ここでは、meta要素を使った適切な文字エンコーディングの指定について学習します。

▶ Step① 文字化けが起きる原因を知ろう

コンピュータは、1つ1つの文字を固有の番号で管理しています。この固有の番号は、簡単にいうと文字の出席番号のようなものです。実際にはこれほどシンプルではありませんが、たとえば1番が「あ」に、2番が「い」に、3番が「う」に対応するようなイメージで文字が管理されています。

1年1組	
1	あ
2	い
3	う

この、どの番号がどの文字になるのかを表す対応表(文字セット)には、さまざまな種類があります。たとえば、ひらがなやカタカナ、漢字を集めた日本語の文字セットや、ハングル文字や漢字を集めた韓国語の文字セット、世界中の文字を扱えるようにした「Unicode」という文字セットなどがあります。

テキストエディタやブラウザは、コンピュータが文字セットを扱いやすいように、データを変換しながら文字を保存したり表示したりしています。この変換方式を「文字エンコーディング」といい、いくつかの種類があります。

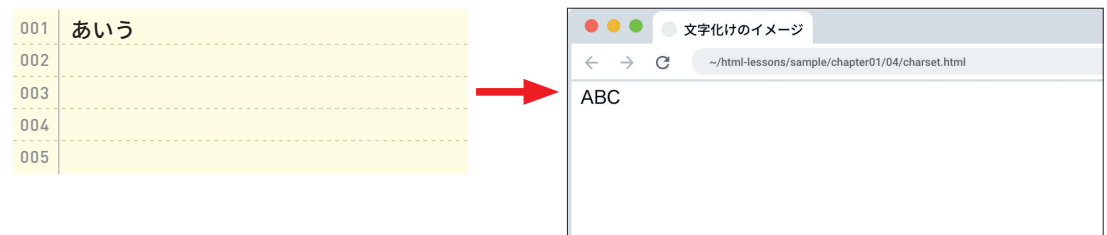
文字エンコーディング	説明
UTF-8	世界中の文字を扱えるようにした「Unicode」という文字セットに利用する
Shift_JIS	主に Windows で使用される。日本語用の文字セットに利用する
EUC-JP	主に UNIX という OS で使用される。日本語用の文字セットに利用する

Webでは、主にこれら3つの文字エンコーディングが利用されていますが、これから新しくHTMLファイルを作る場合は「UTF-8」である必要があります。そのほかの文字エンコーディングに関しては、古いHTML用だと考えてかまいません。

HTMLはテキストエディタなどで記述し、ブラウザで閲覧します。この時にもし、テキストエディタが使う文字エンコーディングと、ブラウザが使う文字エンコーディングが異なれば、テキストエディタとブラウザとで参照される文字セットの文字が異なるため、意図しない文字が表示されてしまいます。仮に「1年1組」という文字エンコーディングが参照する文字セットと、「1年2組」という文字エンコーディングが参照する文字セットが存在し、以下のような対応表だったとします。

1年1組		1年2組	
1	あ	1	A
2	い	2	B
3	う	3	C

もし「1年1組」に設定されたテキストエディタで、「あ」「い」「う」と書いてHTMLファイルを保存すると、コンピュータには「1」「2」「3」という番号で保存されるイメージです。そして、そのHTMLファイルをもし「1年2組」に設定されたブラウザで開くと、「1年2組」が参照する文字セットの対応表を基に、「1」「2」「3」に対応する「A」「B」「C」が表示されてしまうのです。わかりやすく解説するためにかなり簡略化していますが、これが文字化けの原因です。



また、「UTF-8」や「Shift_JIS」などが「文字コード」と呼ばれることもありますが、本来「文字コード」という言葉は「文字に割り当てられた番号」を指すため、本書では「文字エンコーディング」という言葉で解説しました。会話の中では「文字コード」という言葉のほうが多く使われている印象があるため、もし「UTF-8」などの文字エンコーディングについて話す場合は「文字コード」を使ってもよいかもしれません。

Keyword

文字コード

各文字に割り当てられた番号のこと。どの番号がどの文字に対応するかを表した対応表(文字セット)や、文字エンコーディングのことを指す場合もある。

CHAPTER 1 の理解度をチェック!

問 下記の問題を解いてこのチャプターの理解度をチェックしましょう。

1. 「html-lessons」→「chapter01」→「training」フォルダ内に下記のファイル名で新規ファイルを作成する。

ファイル名	index.html
-------	------------

2. HTML ファイルの冒頭にブラウザが互換モードで表示しないように、DOCTYPEを記述する。
3. HTML の全体構造の要素 (html 要素、head 要素、body 要素、title 要素) を記述する。
4. 適切な要素を用いて下記の文字エンコーディングを指定する。

文字エンコーディング	UTF-8
------------	-------

5. ページのタイトルを下記の文字列にする。

タイトルの文字列	Dummy Creations 架空サイトを作る架空のWeb制作会社
----------	--------------------------------------

6. ページの本文を下記の文字列にする。

本文の文字列	Dummy Creations 私たちは架空サイトを作ることに、命を燃やすプロ集団です。
--------	---

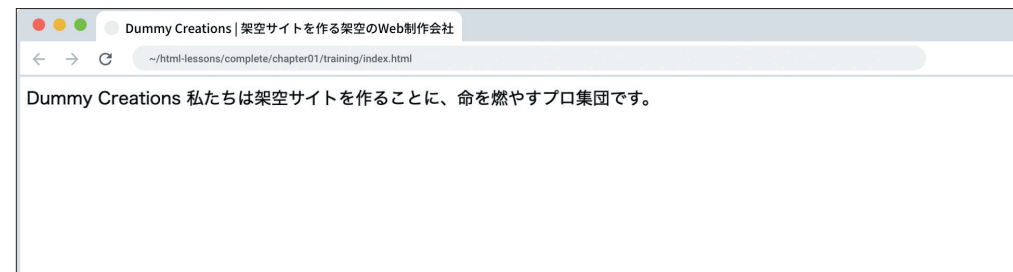


テキストエディタでコードを記述できたら、HTML ファイルを、ブラウザにドラッグ&ドロップしてみよう!

はい!



完成イメージ



解答例 complete/chapter01/training/index.html

```

001 <!DOCTYPE html> ①
002 <html lang="ja"> ②
003 <head> ④
004 <meta charset="UTF-8"> ⑥
005 <title>Dummy Creations | 架空サイトを作る架空のWeb制作会社</title> ⑦
006 </head> ④
007 <body> ⑤
008 Dummy Creations ⑧
009 私たちは架空サイトを作ることに、命を燃やすプロ集団です。
010 </body> ⑤
011 </html> ②
  
```



冒頭には、ブラウザが互換モードで表示しないように<!DOCTYPE html>を記述します①。DOCTYPEの下にhtml要素を配置し②、ほかのすべての要素はこのhtml要素の中に配置します③。html要素の中は、head要素④とbody要素⑤で構成しています。head要素の中にはmeta要素を使ってドキュメントの文字エンコーディングを指定し⑥、title要素を使ってページのタイトルを配置します⑦。ブラウザに表示する文章などのコンテンツは、すべてbody要素の中に配置します⑧。

うまくできました!



よかったね。この、基本的な記述は、どのHTMLファイルにも共通する雛型のようなのなので、忘れずに覚えておこうね。

はい!



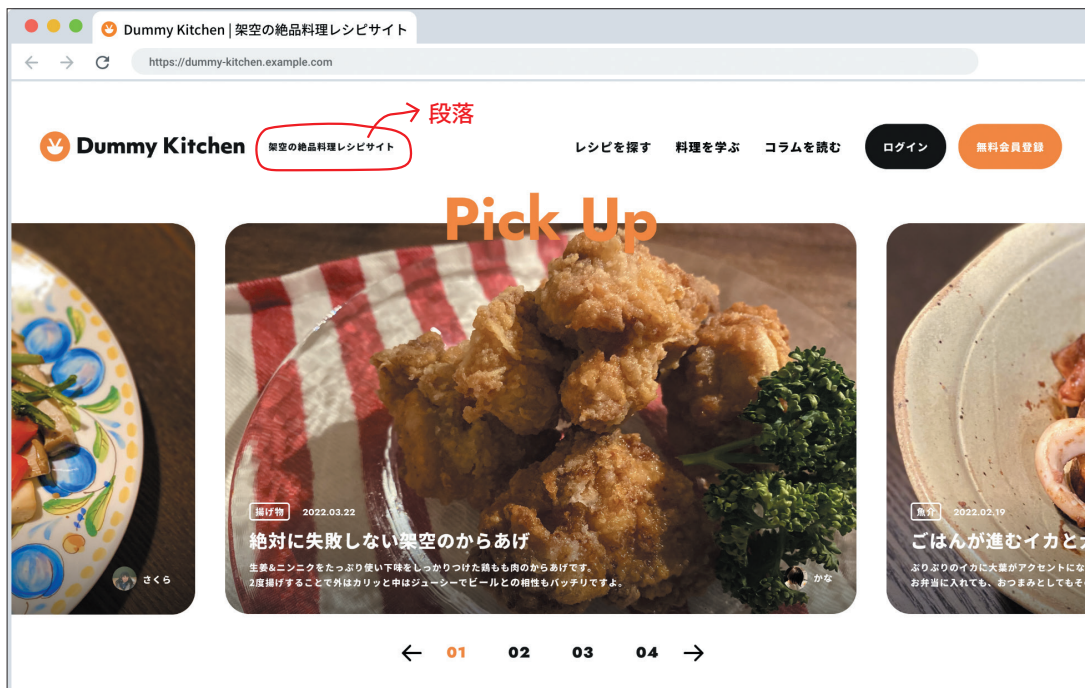
それから、このあとの章末練習問題でも、問題のHTMLを書き終えたら、ブラウザでの表示を確認するようにしてね。

2-2 段落を表す

テキスト関連の要素の中でもっとも使うであろう、段落を表す要素を学習しましょう。段落を表す要素は、論理的な段落だけではなく、構造的な文のまとまりにも使用することができます。

▶ Step 1 段落を表す要素を知ろう

段落は、**パラグラフ**という意味の、**p**要素を使って表します。Webサイト上のほとんどの説明文や概要文は、この**p**要素でマークアップします。段落と聞くと長い文章をイメージするかもしれませんが、HTMLでは一言レベルの文章であっても、この**p**要素でマークアップして問題ありません。



Keyword

p要素

●意味

パラグラフ
paragraphの意味で、段落を表す。

●カテゴリ

フロー・コンテンツ

●コンテンツ・モデル (内包可能な要素)

フレージング・コンテンツ

●利用できる属性

グローバル属性

たった一言の文章でも、**p**要素でマークアップできるんですね。

一般的な段落のイメージとは異なるよね。**p**要素は、もちろん**文章の段落にも使えるし、単語レベルのものにも使えるよ。**

さまざまな種類の文を囲める要素なんですね。

```
007 <body>
008 <h1>Dummy_Kitchen</h1>
009 <p>架空の絶品料理レシピサイト</p>
010
011 <h2>Pick_Up</h2>
012
013 <h3>絶対に失敗しない架空のからあげ</h3>
014 <p>生姜&amp;ニンニクをたっぷり使い下味をしっかりつけた鶏もも肉のからあげです。2度揚げする
    ことで外はカリッと中はジューシーで<em>ビールとの</em>相性もバッチリですよ。</p>
015
016 <h3>ごはんが進むイカと大葉のバター醤油炒め</h3>
017 <p>ぷりぷりのイカに大葉がアクセントになって、ごはんが<em>何杯でも</em>食べられる逸品で
    す！お弁当に入れても、おつまみとしてもその役割を全うします。</p>
018
019 <h2>新着の絶品架空レシピ</h2>
020 <p>最近投稿された絶品レシピをご紹介します。</p>
021
022 <h3>名店の味を再現！自宅で作る究極のTKG</h3>
023 <p>架空のたまごかけ飯専門店「だみい屋」さんのTKGを<em>家庭で</em>再現してみました。隠
    し味のごま油がワンランク上のTKGに仕上げてくれます。</p>
024
025 <p><strong>当サイトのレシピは全て架空のレシピのため味の保証は致しません。</strong></
    p>
026
027 <p>Dummy_Kitchenへのお問い合わせは、お問い合わせフォームか下記の住所宛にご連絡下さい。
    </p>
028
029 <p>
030 〒100-XXXX<br>
031 東京都架空区架空町1-2-3-2001
032 </p>
033
034 <p><small>&copy; 2022 Dummy_Kitchen</small></p>
035 </body>
```

small 要素は、重要度を下げたためや、文字を小さくするためではなく、著作権表記や警告、免責事項などに使うのですね。では、免責事項があったら、難しいことは考えず **small** 要素で囲んでしまっても問題ないですか？

small 要素は短いテキストを対象としているので、なんでも囲めるわけではないよ。たとえば、利用規約を記載するページのテキストは1つのメインコンテンツとして扱うべきで、すべて **small** 要素にするのはふさわしくないよ。

2-8 日時

イベントの開催日やブログ記事の公開日などを、コンピュータでも理解できる日時として表すための要素を学習しましょう。

Step 1 日時を表す要素を知ろう

記事の公開日やイベント開催日などの日付や時刻には、HTML5から登場した **time** 要素を使います。今まで紹介してきた要素と比べると、少し注意が必要です。



Keyword

time 要素

- 意味
日付や時刻、タイムゾーンのオフセットなどを表す。
- カテゴリ
フロー・コンテンツ
フレーズ・コンテンツ

- コンテンツ・モデル (内包可能な要素)
datetime 属性を指定した場合はフレーズ・コンテンツ。そうでなければ、あらかじめ定められたフォーマットに沿った日時のテキスト
- 利用できる属性
グローバル属性
datetime 属性…コンピュータが理解できる日時を指定する

datetime 属性

- 役割
コンピュータが理解できるフォーマットで日時を指定する。
- 属性値
コンピュータが理解できるフォーマットの日時

Web サイト制作の仕事とは？

この本を手にとってくださっている方の中には、これから Web 業界に進もうと思って HTML の勉強をされている方もいらっしゃると思います。Web サイト制作のお仕事は、基本的にクライアントの依頼を基に制作を進めていきますが、いきなり HTML を書いていくわけではありません。

まず、クライアントの要望を基に Web サイトの戦略を考え、閲覧時の導線やページ内の構成など、さまざまな設計を行います。次に、デザインソフトを用いてデザインの完成イメージを作り、そのイメージを基に、HTML や CSS、JavaScript などコーディングを行います。

これらの工程を Web ディレクター、Web デザイナー、コーダーなど、さまざまな職種の方と分業しながら作っていくことが多いです。なお、筆者の場合は、戦略立案からデザイン、コーディングまでのすべてを 1 人で行うことが多いのですが、プロジェクトによっては、フリーランス Web デザイナーの方とタッグを組んで、コーディングだけを担当することもあります。



筆者が戦略立案からデザイン、コーディングまで担当した Web サイト



デザイナーのアシダミヅホさんがデザインを担当し、筆者がコーディングを担当した Web サイト
ASHIM FACTORY アシダミヅホ (<https://ashim-factory.com/>) Twitter : @Ashida_Assy

このように Web サイト制作は、さまざまな工程をさまざまな職種の方と分業しながら制作していくことが多いです。よって、これから Web 業界を目指されるみなさんは、Web 制作の勉強をしていく中で、今後自分がやりたいことは Web サイト制作の流れの中のどの工程なのか、また、こういった職種の人が担当する工程なのか、調べてみるとよいと思います。制作会社によって、それぞれの役割の呼び方や制作のスタイル、ワークフローが異なりますし、それぞれの職種が担当する範囲もさまざまかと思いますので、そのあたりも含めて業界研究することが大切です。

SECTION

3-2 URLの指定方法

前のセクションでは、リンクの設定方法を学びました。ここでは、リンク先や画像など、参照先を指定する際に用いる、URLの指定方法について学習しましょう。

▶ Step 1 URLの指定方法を知ろう

リンクを設定する場合や、画像を表示させたい時は、**URL**とよばれる文字列によって**リンク先のページや画像の保存されている場所を指定する**必要があります。**href**属性の属性値としてURLを指定するには、絶対URLや相対URLといった指定方法があります。絶対URLは目的地の**絶対的な位置**を指定する方法で、相対URLは基準となる場所から目的のファイルを見た時の、**相対的な位置**を指定する方法です。とくに相対URLの指定方法は複雑なため、初学者が最初につまずくポイントになることが多いです。このセクションでは、主に相対URLの指定方法について解説します。

▶ Step 2 絶対URLと相対URLの違いと使い方を知ろう

▶ 絶対URLは「http」や「https」などで始まる

絶対URLは「http」や「https」などから始まるURLのことで、「https://example.com/menu/lunch.html」などの形式となります。郵便物の宛先として記述する住所のようなイメージで、主に**外部のWebサイトへのリンク**に利用されます。すでに公開されているWebページの場合は、ブラウザのアドレスバーで絶対URLを確認することができるため、リンク先として指定したいページをブラウザで表示した上で、アドレスバーにある絶対URLをコピーし、**href**属性の属性値に貼りつけましょう。

サンプル 絶対URLの例

```
<a href="https://gihyo.jp/book">技術評論社</a>
```

絶対URLを使用するのは、たとえばどのような場合でしょうか？

自分のWebサイト内に、外部サイトのリンクを貼る場合だよ。たとえば、関連情報が掲載されている外部サイトや関連会社へのリンク、SNSやYouTubeなどのリンクなどが挙げられるね。

解答例 complete/chapter03/01/index.html

```
007 <body>
008 <h1><a href="index.html">Dummy Kitchen</a></h1>
009 <p>架空の絶品料理レシピサイト</p>
010
011 <h2>Pick Up</h2>
012
013 <a href="recipe/karaage.html">
014 <h3>絶対に失敗しない架空のからあげ</h3>
015 <p><time datetime="2022-03-22">2022.03.22</time></p>
016 <p>生姜&ニンニクをたっぷり使い下味をしっかりつけた鶏もも肉のからあげです。
    2度揚げすることで外はカリッと中はジューシーで<em>ピールとの</em>相性もバッチリですよ</p>
017 <p>かな</p>
018 </a>
019
020 <h2>スポンサー</h2>
021 <p>当架空サイトは下記のスポンサー様のお力で運営しております。</p>
022
023 <ul>
024 <li><a href="https://gihyo.jp/book" target="_blank">技術評論社(新しい
    タブで開く)</a></li>
025 <li><a href="https://shibajuku.net" target="_blank">Shibajuku(新しい
    タブで開く)</a></li>
026 </ul>
027 </body>
```

余裕があれば

●リンク先との関係性を示した場合

解答例 リンク元の情報へのアクセス制限と、広告であることを表した例

```
023 <ul>
024 <li><a href="https://gihyo.jp/book" target="_
025 blank" rel="noopener sponsored">技術評論社(新しいタブで開く)</a></li>
026 <li><a href="https://shibajuku.net" target="_blank" rel="noopener
027 sponsored">Shibajuku(新しいタブで開く)</a></li>
028 </ul>
```

どう？指定したとおりのページにジャンプできたかな？

わあ！すごいです。これがハイパーリンクなのですね。リンク先を新しいタブで展開することもできました。

そうだね。リンク先を指定する方法はよく使うので、**a要素にhref属性を指定する**ということ覚えておこう。

3-4 画像を埋め込む

ここでは、画像の埋め込み方法について学習しましょう。また、高解像度ディスプレイへの対応や読み込みタイミングについてなど、発展的な内容も学びます。

▶ Step① 画像を表す要素を知ろう

Webページを制作していると、ロゴや写真など、画像を使用したい場面が多くあると思います。画像は **image** を意味する **img** 要素を使い、**src** 属性を用いて画像を参照します。なお、**img** 要素は**空要素**となります。



Keyword

img 要素

- 意味
image の意味で、画像を表す。
- カテゴリ
フロー・コンテンツ
フレージング・コンテンツ
エンベディッド・コンテンツ
- コンテンツ・モデル (内包可能な要素)
なし (空要素)
- 利用できる主な属性
グローバル属性
src 属性.....参照する画像ファイルの URL を指定する (必須)

- srcset 属性.....高解像度ディスプレイなど複数の画像ファイルの URL を指定する
- alt 属性.....画像の代替テキストを指定する
- width 属性.....画像の横幅を指定する
- height 属性.....画像の高さを指定する
- decoding 属性.....画像のデコードのヒントを指定する
- loading 属性.....画像の読み込みタイミングを指定する.....など

▶ Step② img 要素の使い方を知ろう

img 要素では、画像を表示したい箇所に **** タグを配置します。**img** 要素には **src** 属性を使って、参照する画像ファイルの URL を指定します。なお、指定した画像が表示されない場合に備え、**alt** 属性を使って **画像の代替テキスト** が表示されるように指定しておきましょう。このようにすることで、ネットワークが不調な場合に対応できるだけでなく、スクリーンリーダーなどを利用しているユーザに対しても、代替テキストによって、画像と同等の情報を提供できるようになります。

Keyword

src 属性 (必須)

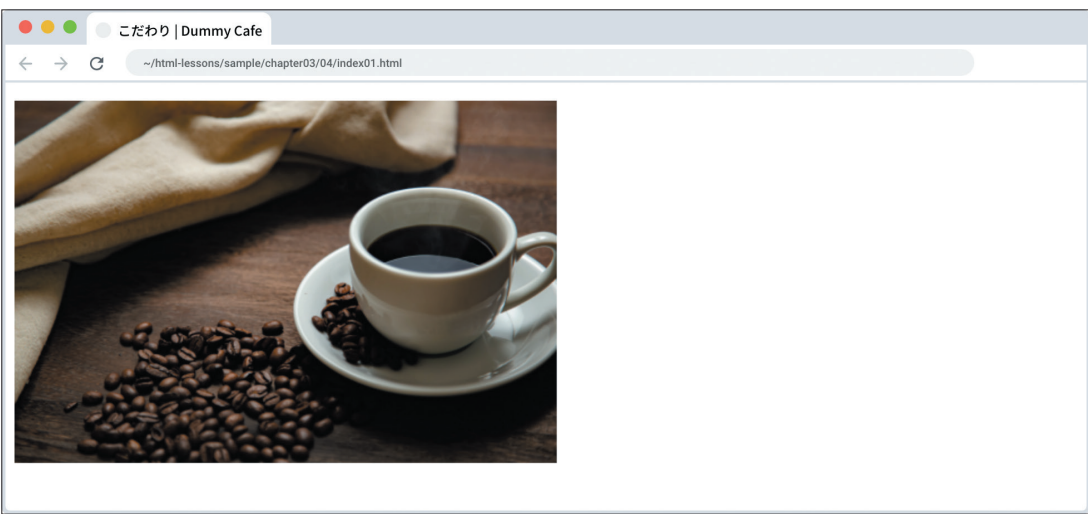
- 役割
ファイルの参照先を指定する。
- 属性値
参照するファイルの URL

alt 属性

- 役割
オルタナティブ テキスト
alternative text の意味で、代替テキストを指定する。ユーザによって投稿された画像などでない限りは指定するようにする。
- 属性値
表示する画像と置き換えてもページの意味が変わらないような代替テキスト。ただし、装飾目的の画像の場合は空にする。

使い方 | img 要素の基本的な使い方

サンプル 「images」フォルダ内の「coffee.jpg」という画像を挿入する例



button	ボタンを表す	フロー・コンテンツ フレージング・コンテンツ インタラクティブ・コンテンツ	フレージング・コンテンツ ただし、子孫にインタラクティブ・コンテンツおよびtabindex属性が指定された要素は配置不可
textarea	複数行の入力フィールドを表す	フロー・コンテンツ フレージング・コンテンツ インタラクティブ・コンテンツ	テキスト
select	セレクトボックス (選択式メニュー) を表す	フロー・コンテンツ フレージング・コンテンツ インタラクティブ・コンテンツ	0個以上のoption要素、optgroup要素、スクリプトサポーティング要素
option	セレクトボックス (選択式メニュー) または、入力を補完する候補の選択肢を表す	なし	・label属性とvalue属性が指定されている場合は「空」 ・label属性は指定されているが、value属性は指定されていない場合は「テキスト」 ・label属性が指定されておらず、datalist要素の子要素でない場合は「空白ではないテキスト」 ・label属性が指定されておらず、datalist要素の子要素である場合は「テキスト」
optgroup	セレクトボックスの選択肢のグループを表す	なし	0個以上のoption要素とスクリプトサポーティング要素
label	コントロール部品のラベルを表す	フロー・コンテンツ フレージング・コンテンツ インタラクティブ・コンテンツ	フレージング・コンテンツ ただし、子孫にlabel要素は配置不可 また、このラベルが示すコントロール部品 (button要素、input要素 (type属性が"hidden"以外)、meter要素、output要素、progress要素、select要素、textarea要素、フォームに関連づけられたカスタム要素) 以外のコントロール部品の要素も配置不可

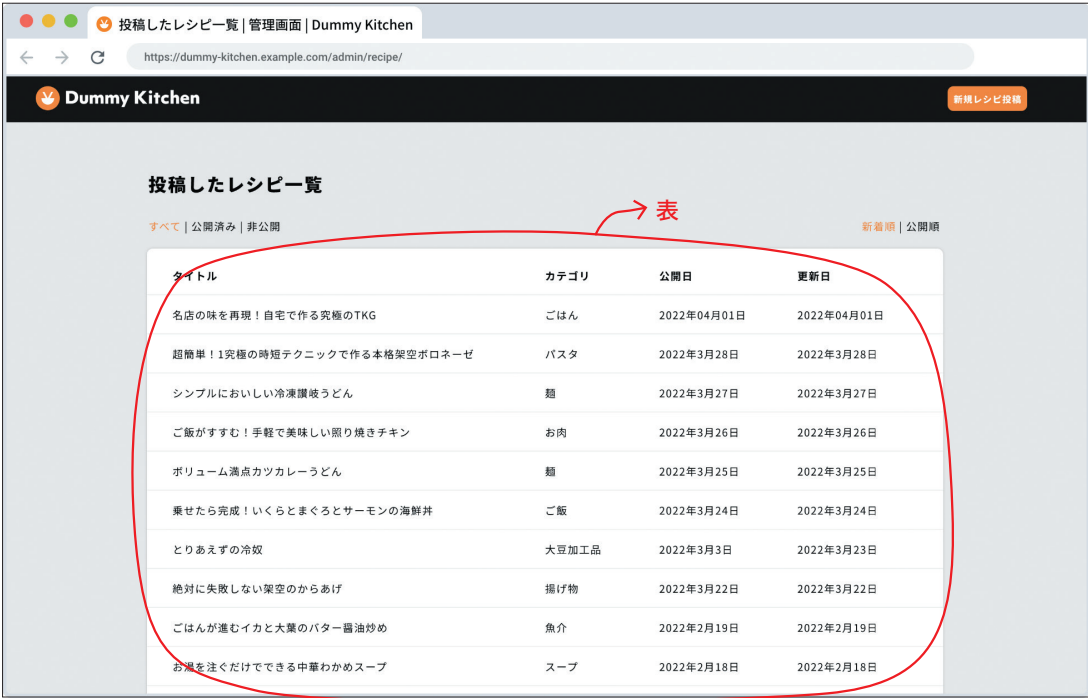
4-1 表 (テーブル) を作る

カレンダーや時間割など、表 (テーブル) の作り方を学習しましょう。表は今まで紹介してきた要素と比べると少々難しいので、まずは基本的な作り方から紹介します。

▶ Step 1 表を表す要素を知ろう

カレンダーや時間割などの表 (テーブル) を作成するには、^{テーブル}table要素を使います。ただし、表は、table要素単体では作成することができません。

まず、table要素で表全体を囲み、各行をtr要素でマークアップします。さらに、その行の中にある各セル (マス) をth要素またはtd要素で挟みます。th要素は^{テーブル}tableの^{ヘッダー}header cellの意味で、見出しとなるセルを挟みます。td要素は^{データ}tableの^{セル}data cellの意味で、見出しの内容を説明するセルを挟みます。また、^{キャプション}caption要素を使うことでテーブルにキャプションを指定することも可能です。



```
024 <li><input type="checkbox" name="keywords[]" value="お弁当"> お弁  
当</li>  
025 <li><input type="checkbox" name="keywords[]" value="子供向け"> 子  
供向け</li>  
026 <li><input type="checkbox" name="keywords[]" value="ダイエット">  
ダイエット</li>  
027 <li><input type="checkbox" name="keywords[]" value="作り置き"> 作  
り置き</li>  
028 <li><input type="checkbox" name="keywords[]" value="節約"> 節約  
</li>  
029 </ul>  
030  
031 <p><button>投稿する</button></p>  
032 </form>  
033 </body>
```

難しかったけれど、なんとかできました。

着実にできることが増えていっているね！

よし！CSSを勉強して「タイトル」という文字を赤文字にしよう。

……どうして？

だって、「必須項目」という文字をわざわざ書くのが面倒くさいんですもん。「赤文字は必須項目です」ってどこかに1回書いておけば、いちいち必須項目と書かなくても伝わりますよね！

う〜ん、それはあまりよくないよ。

どうしてですか！

もし、色覚特性のあるユーザが読んだら、そこが赤文字かどうか判断できないことがあるかもしれないよね。

確かに……それはよくないですね……

Webページを作る時は、色に依存しないように気をつけようね。

SECTION

4-4 複数行の入力フィールド

お問い合わせ内容を書く項目や、ブログ記事の投稿欄などで見かける複数行からなるテキストエリアの作り方を学習しましょう。

▶ Step① 複数行の入力フィールドを表す要素を知ろう

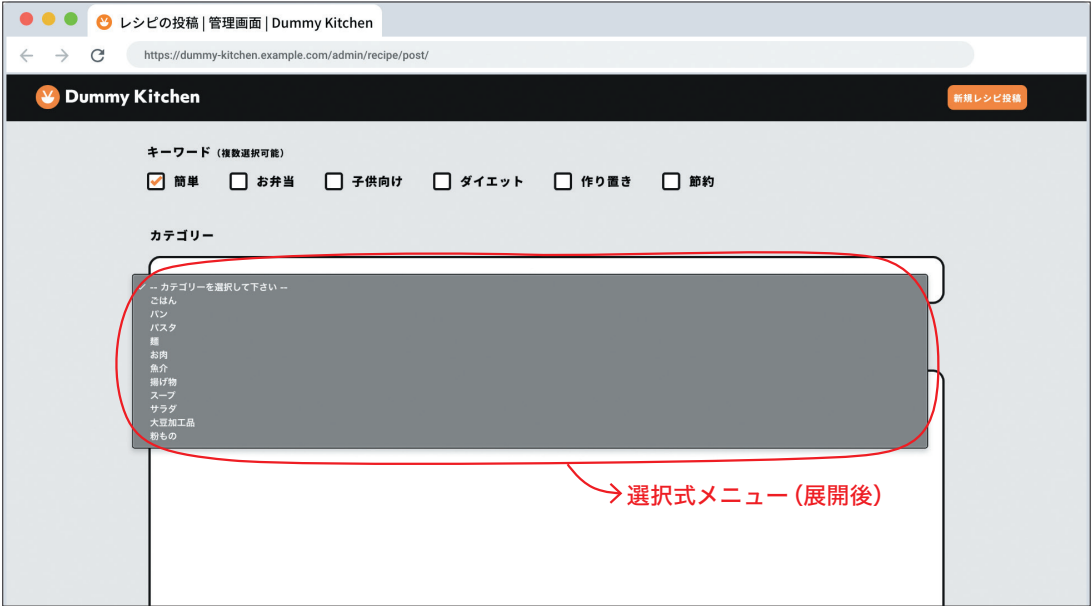
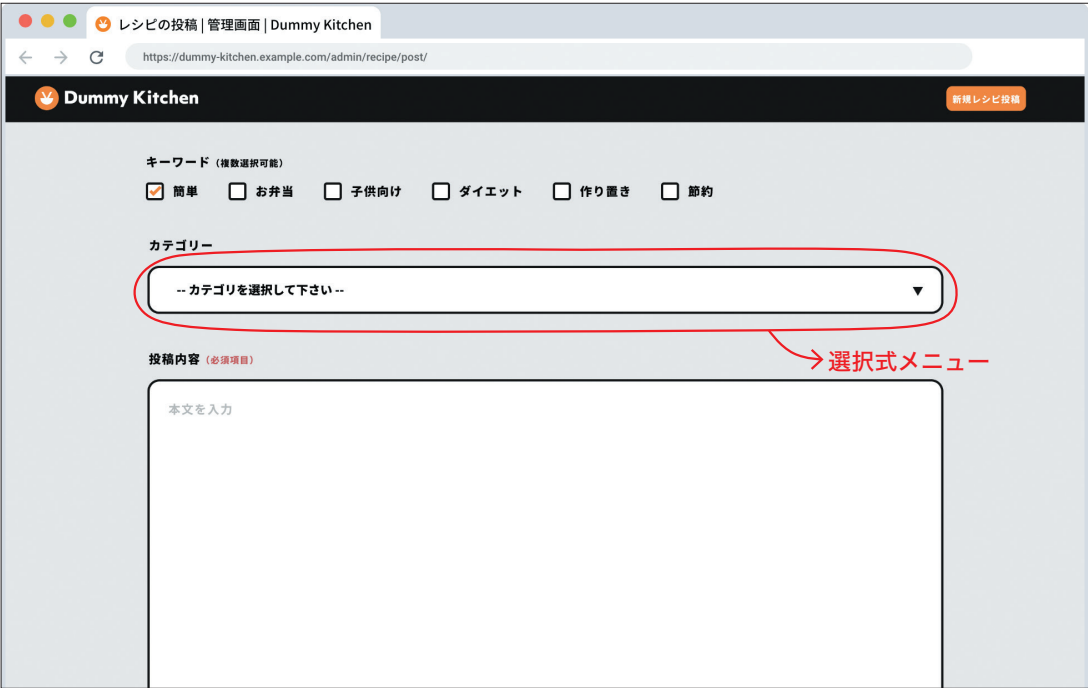
お問い合わせフォームの内容や、SNSなどの投稿を記述するテキストフィールドには、**複数行の記述**ができる**テキストフィールド**が用いられています。これらは、**textarea**要素を使って表すことができます。

4-5 選択式メニュー

プロフィール入力欄でよく見かける「お住まいの都道府県」など、たくさんの選択肢の中から項目を選択する時に使う選択式メニューについて学習しましょう。

▶ Step 1 選択式メニューを表す要素を知ろう

たくさんの**選択肢の中から、項目を選択するような選択式メニュー**を見かけることがあります。このようなメニューは**セレクトボックス**などとも呼ばれ、**select**要素を用いて表すことができます。**select**要素は**ul**要素や**ol**要素のように各選択肢を囲む要素で、各選択肢は**select**要素の中に、**option**要素を用いて配置します。なお、**option**要素は、**optgroup**要素を用いてグループ化することもできます。



Keyword

select 要素

- 意味
セレクトボックス (選択式メニュー) を表す。
- カテゴリ
フロー・コンテンツ
フレージング・コンテンツ
インタラクティブ・コンテンツ
- コンテンツ・モデル (内包可能な要素)
0個以上の option 要素、optgroup 要素、スクリプトサポーティング要素

- 利用できる主な属性
グローバル属性
name 属性.....名前を指定する
multiple 属性.....複数の値を選択できるかを指定する
required 属性.....送信時にこの部品の選択を必須にする
.....など

option 要素

- 意味
セレクトボックス (選択式メニュー) の選択肢、または入力を補完する候補を表す。
- カテゴリ
なし
- コンテンツ・モデル (内包可能な要素)
 - ・ label 属性と value 属性が指定されている場合は「空」
 - ・ label 属性は指定されているが、value 属性は指定されていない場合は「テキスト」
 - ・ label 属性が指定されておらず、datalist 要素の

- 子要素でない場合は「空白ではないテキスト」
- ・ label 属性が指定されておらず、datalist 要素の子要素である場合は「テキスト」
- 利用できる主な属性
グローバル属性
label 属性.....ユーザに表示するラベルを指定する
selected 属性.....選択状態にする
value 属性.....送信される値を指定する
.....など

6-1 ページの紹介文を設定する

検索エンジンの検索結果ページなど、制作したページがほかのページで紹介される際に使われる紹介文を指定する方法を学習しましょう。

▶ Step 1 ページの紹介文を表す要素を知ろう

ページに紹介文を設定しておくと、検索結果の一覧ページなどで紹介される際の表示内容を指定することができます。ページの紹介文は 文字エンコーディングの指定でも紹介した **meta** 要素を使うことで指定できます。**meta** 要素では、**name** 属性の値に **description** を指定することで、**content** 属性に紹介文の内容を指定することができます。

Keyword

meta 要素

- 意味
メタデータを表す。
- カテゴリ
メタデータ・コンテンツ
itemprop 属性を指定した場合は、以下のカテゴリにも属する
フロー・コンテンツ
フレーズ・コンテンツ
- コンテンツ・モデル (内包可能な要素)
なし (空要素)

● 利用できる主な属性

グローバル属性

name 属性.....メタ情報の名前を指定する

http-equiv 属性.....HTTP レスポンスヘッダーの情報を指定する

content 属性.....name 属性または http-equiv 属性で指定した情報の種類に対する内容を指定する

charset 属性.....文字エンコーディングを指定する

.....など

meta 要素を使って、紹介文を記述するのですね。

そうだね。meta 要素を使って、各ページ固有の紹介文を指定するといいよ。

すべてのページで同じ紹介文にするのは駄目なんですね。ページごとに考えるのが大変そうですね。あ、本文からそのままコピーしてしまえばいいんですかね？

ページの紹介文のためにちゃんと考えたほうがいいよ。この文だけ見ても、ユーザがサイトにアクセスしたくなるような紹介文にしてね！

▶ Step 2 ページの紹介文の指定方法を知ろう

meta 要素は、基本的に **head** 要素内に配置する要素になります。紹介文を指定する場合は、**name** 属性の属性値に **description** を指定し、続けて **content** 属性に紹介文を記述します。紹介文は、ページの内容を表した各ページ固有の内容を指定しましょう。

Keyword

name 属性

- 役割
情報の名前を指定する。
- 主な属性値
author.....ページの制作者
description.....ページの紹介文
generator.....ページ制作に使用したソフト
keywords.....ページのキーワード
robots.....検索エンジンのクローラーの制御
.....など

content 属性

- 役割
name 属性または http-equiv 属性で指定した情報の種類に対する内容を指定する。
- 主な属性値
情報の種類に対する内容
.....など

使い方

ページの紹介文を指定する meta 要素の使い方

<meta name="description" content="ページの紹介文">

サンプル

ページの紹介文を meta 要素で指定した例

<meta name="description" content="Dummy Cafe は「いつもの。」の一言ですべてが通じる架空のカフェです。一度でもご来店頂いたお客様は、次回から「いつもの。」というたったひとつの言葉で、最高のひとときをお過ごし頂けます。">

カフェ - SEARCH 検索

https://search.example.com

SEARCH カフェ

https://dummy-cafe.example.com

Dummy Cafe - 「いつもの。」 が通じる、あなたのカフェ。

Dummy Cafe は「いつもの。」の一言で全てが通じる架空のカフェです。一度でもご来店頂いたお客様は、次回から「いつもの。」というたったひとつの言葉で、最高のひとときをお過ごし...

6-2 SNSの共有情報を設定する

OGP (Open Graph Protocol) について知り、SNSでページをシェアする際に表示される画像や紹介文などの設定方法を学習しましょう。

▶ Step① SNSの共有情報を表す要素を知ろう

SNSのタイムラインにWebページのURLを貼り付けると、そのWebページのタイトルやサムネイル画像が表示されることがあります。これは、**OGP (Open Graph Protocol)** という仕組みを使って、WebページがSNSでシェアされた際に表示するサムネイル画像やページの紹介文をあらかじめ設定しているためです。このOGPも **meta** 要素を使って設定することができます。

▶ Step② OGPの使い方を知ろう

HTML Standardでは定義されていませんが、**meta** 要素に **property** 属性を使ってOGPの情報の種類を指定することで、**content** 属性にその内容を記述できます。なお、OGPで利用される一般的なサムネイル画像のサイズは幅1200px × 高さ630pxです。

Keyword

property 属性

●役割

OGPなどの情報の種類を指定する。

●主な属性値

og:url.....ページのURL

og:title.....ページのタイトル

og:description.....ページの紹介文

og:image.....ページのサムネイル画像のURL

og:type.....ページの種類

og:locale.....ページの言語

og:site_name.....サイト名

fb:app_id.....FacebookのApp ID

.....など

OGPというのは、SNSにURLを貼りつけた時に、プレビュー情報のように表示される画像や文のことですか？

そうそう！SNSでシェアした時に表示されるサムネイル画像やページの情報も、HTMLで指定することができるんだ。

使い方 OGPの設定方法

```
<meta property="og:url" content="ページのURL">
<meta property="og:title" content="ページのタイトル">
<meta property="og:description" content="ページの紹介文">
<meta property="og:image" content="ページのサムネイル画像のURL">
<meta property="og:type" content="ページの種類(website、article、profile など)">
<meta property="og:locale" content="ページの言語(日本語: ja_JP)">
<meta property="og:site_name" content="サイト名">
```

サンプル OGPを設定した例

```
<meta property="og:url" content="https://dummy-cafe.example.com">
<meta property="og:title" content="Dummy Cafe - 「いつもの。」が通じる、あなたのカフェ。">
<meta property="og:description" content="Dummy Cafeは「いつもの。」の一言で全てが通じる架空のカフェです。一度でもご来店頂いたお客様は、次回から「いつもの。」というたったひとつの言葉で、最高のひとときをお過ごし頂けます。">
<meta property="og:image" content="https://dummy-cafe.example.com/images/ogp.png">
<meta property="og:type" content="website">
<meta property="og:locale" content="ja_JP">
<meta property="og:site_name" content="Dummy Cafe">
```

ページの種類(og:type)の値には、「website」や「article」や「profile」など、いくつかの種類があります。「website」以外の値には、追加で複数の値を設定できます。たとえば「article」の場合は、記事の公開日時や更新日時、著者などの情報を追加することができます。

使い方 ページの種類を「article」としてOGPを設定する方法

```
<meta property="og:type" content="article">
<meta property="article:published_time" content="記事の公開日時">
<meta property="article:modified_time" content="記事の更新日時">
<meta property="article:expiration_time" content="記事の期限切れ日時">
<meta property="article:author" content="記事の著者">
<meta property="article:section" content="記事のカテゴリ">
<meta property="article:tag" content="記事のタグ">
```

▶ Step③ OGP を設定してみよう

それでは、実際にOGPやTwitterの情報を設定してみましょう。

- ①「html-lessons」→「chapter06」フォルダ内にある「index.html」をテキストエディタで開く。
- ②下記のようにOGPとTwitterの情報を設定する。

ページのURL	https://dummy-kitchen.example.com
ページのタイトル	Dummy Kitchen 架空の絶品料理レシピサイト
ページの概要文	Dummy Kitchenは架空の絶品料理のレシピを公開しています。手軽に作れてとっても美味しい架空のレシピを多数公開していますので、今夜の献立に悩んだらぜひご活用下さい。
ページのサムネイル画像のURL	https://dummy-kitchen.example.com/images/ogp.png
ページの種類	website
ページの言語	日本語
サイト名	Dummy Kitchen
Twitterのカードのタイプ	大きなサムネイル画像とタイトル、説明文が表示される
【Twitterアカウントをお持ちの場合】 サイト管理者のTwitterアカウント	ご自身のTwitterアカウント

③上書き保存する。

解答例complete/chapter06/02/index.html

```
003 <head>
004   <meta charset="UTF-8">
005   <meta name="description" content="Dummy Kitchenは架空の絶品料理のレシピを
公開しています。手軽に作れてとっても美味しい架空のレシピを多数公開していますので、今夜の献立に悩
んだらぜひご活用下さい。">
006   <meta property="og:url" content="https://dummy-kitchen.example.com">
007   <meta property="og:title" content="Dummy Kitchen | 架空の絶品料理レシピ
サイト">
008   <meta property="og:description" content="Dummy Kitchenは架空の絶品料理
のレシピを公開しています。手軽に作れてとっても美味しい架空のレシピを多数公開していますので、今夜
の献立に悩んだらぜひご活用下さい。">
009   <meta property="og:image" content="https://dummy-kitchen.example.
com/images/ogp.png">
010   <meta property="og:type" content="website">
011   <meta property="og:locale" content="ja_JP">
012   <meta property="og:site_name" content="Dummy Kitchen">
013   <meta name="twitter:card" content="summary_large_image">
014   <meta name="twitter:site" content="@hilosiva">
015   <title>Dummy Kitchen | 架空の絶品料理レシピサイト</title>
016 </head>
```

6-3 ファビコンを設定する

ここでは、ブラウザのタブなどに表示され、サイトの識別に役立つファビコンの設定方法を学習しまし
ょう。

▶ Step① ファビコンを設定する要素を知ろう

Webサイトを閲覧していると、ブラウザのタブにアイコンが表示されているサイトをよく見かけると
思います。あの画像は「**ファビコン**」と呼ばれ、複数の
タブを同時に開く際に、サイトを識別する上で便利な
画像になります。ファビコンは、一般的に**ICO**とい
う形式で作成します。拡張子は「**.ico**」となり、
「**favicon.ico**」というファイル名で保存されることが
多いです。通常はこの「**favicon.ico**」というファイ
ル名で保存されたファビコンを、Webサイトのルート
ディレクトリ(サイトを公開するWebサーバの公開
ディレクトリの直下)にアップロードするだけで表示
されます。また、**link**要素を使って直接ファビコンを
指定することも可能です。**link**要素は主にCSSファ
イルとのリンクに用いられるのですが、こういった設
定にも利用できます。



スマートフォンやタブレットでの表示例

Keyword

link 要素

- 意味
外部のリソースとのリンクを表す。
- カテゴリ
メタデータ・コンテンツ
要素がbody要素内で許可されている場合は、
以下のカテゴリにも属する
フロー・コンテンツ
フレージング・コンテンツ
- コンテンツ・モデル(内包可能な要素)
なし(空要素)

●利用できる主な属性

グローバル属性

href 属性 … リンク先を指定する

rel 属性 …… リンクするリソースとの関係を指
定する

sizes 属性 … rel 属性に icon が指定されている際
の、アイコンのサイズを指定する
……など

検索エンジンの クローラーを制御する

インターネット上を巡回している検索エンジンのクローラーにサイトの情報をインデックスされないよう、HTMLで制御する方法を学習しましょう。

▶ Step① 検索エンジンに関する設定を知ろう

Googleなどの検索エンジンでは、クローラーと呼ばれるロボットプログラムがインターネット上のWebサイトを巡回しています。巡回したWebサイトの内容は検索エンジンに登録され、検索しやすいように情報が整理され、索引が作られます。これを「**インデックス**」と呼びます。したがって、制限をしない限り、クローラーはあなたのWebページを見つけ、コンテンツが適切かどうか判断し、誰かが検索した時のために、インデックスします。そして、いつか誰かが検索し、ヒットした際に、その内容が検索結果として表示される仕組みです。会員専用のWebページなど、検索エンジンに登録されたくないWebページの場合は、**meta**要素を使って**クローラーを制御**することができます。

▶ Step② クローラーを制御する方法を知ろう

HTMLを使ってクローラーを制御するには、**meta**要素の**name**属性に**robots**と指定し、**content**属性に制御内容を示したキーワードを記述します。複数の制御を行う場合は、「,(半角カンマ)」区切りで指定します。

使い方 クローラーを制御するmeta要素の使い方

```
<meta name="robots" content="クローラーの制御を表すキーワード">
```

●主なキーワード

キーワード	説明
all	制限なし
noindex	インデックスを拒否
nofollow	ページ内のリンクの追跡を拒否
none	noindex, nofollowと同じ意味

このほかにも、Googleの「robots メタタグの指定」ページにはさまざまなキーワードが掲載されていますので、必要に応じて確認してみてください。

参考 robots メタタグの指定 | Google 検索セントラル | ドキュメント | Google Developers
https://developers.google.com/search/docs/advanced/robots/robots_meta_tag?hl=ja

サンプル meta要素を使ってインデックスとリンクの追跡を拒否する例

```
<meta name="robots" content="noindex, nofollow">
```

上記のサンプルでは、クローラーに対してインデックスとリンクの追跡を拒否しています。そのため検索エンジンには登録されず、検索結果にも表示されません。

Memo

特定の検索エンジンのクローラーだけ制御する

name属性のrobotsは、すべてのクローラーに対しての制御になりますが、指定を変更することで特定の検索エンジンのクローラーだけを制御することもできます。たとえばGoogleのクローラーのみを制御する場合は、name属性をgooglebotにします。

サンプル Googleのクローラーに対してインデックスを拒否する指定をした例

```
<meta name="googlebot" content="noindex">
```

また、目的ごとにクローラーが用意されているので、必要に応じてご確認ください。

リンク Google クローラの概要(ユーザー エージェント) | Google 検索セントラル | ドキュメント | Google Developers
<https://developers.google.com/search/docs/advanced/crawling/overview-google-crawlers?hl=ja>

こうやって、検索エンジンに登録しないようにすることもできるんですね。

一般的なサイトでは、あまりそういうことはしないと思うけれど、会員サイトやエラー用のページや、検索結果に出てきてほしくないページがあれば対策することができるよ。

検索エンジンに登録してほしくないページを公開する機会があれば、忘れずに設定したいと思います！

そうしてね。それから、name属性に指定する「robots」の「s」をつけ忘れてる人をよく見かけるので、「s」のつけ忘れに気をつけてね。

うわあ……つけ忘れてしまったら、検索されたくない場合でも、検索結果に表示されてしまうかもしれないということですよね。気をつけます！

7-1

架空のお店の Web ページを
マークアップしてみよう!

CHAPTER1～6で学習した内容を使って、WebページのHTMLを実際にマークアップしてみましょう。わからない時は、〈ヒント〉のページに戻って復習しましょう。

▶ Step 1 完成デザインを確認しよう

①「html-lessons」→「practice」フォルダ内にある「design.png」を開いてデザインを確認する。

