

# CONTENTS

## Lesson 01

はじめに	003
本書の使い方	004
レッスンファイルのダウンロード	006

## www や web サイト制作の 基本を理解しよう 011

1-1 web サイトが表示される仕組み	012
1-2 web サイト制作のキーワード	014
1-3 web サイトに使用される言語	016
1-4 コーディングに必要なアプリケーション	019
1-5 web サイトに使用される画像	021
1-6 エディタの使い方を学ぶ	025
Q 練習問題	030

## Lesson 02

## HTML コーディングの基本を学ぼう 031

2-1 HTML とは	032
2-2 マークアップとは	034
2-3 タグと要素と属性	036
2-4 head 要素と body 要素	040
2-5 head 要素の中に入る要素	042
2-6 web サイトを構成するファイルの管理	046
2-7 HTML ファイルを作成して保存する	048
2-8 web サイトの全体像をしてみる	050
Q 練習問題	052

## Lesson 03

## 画像表示とリンクをマークアップしよう 053

3-1 画像を表示する	054
3-2 リンクを設定する	059
3-3 パスを指定する	067
Q 練習問題	076

## Lesson 04

## リストとナビゲーションを マークアップしよう 077

4-1 順不同リストを作る	078
4-2 順序付きリストを作る	081
4-3 定義リストを作る	084
4-4 リストでナビゲーションを作る	087
Q 練習問題	092

## 表組みをマークアップしよう 093

5-1 シンプルな表組みを作る	094
5-2 セルを結合する	098
5-3 より高度な表組みを作る	104
Q 練習問題	110

## フォームをマークアップしよう 111

6-1 フォームの基本を身につける	112
6-2 入力形式に合わせたフォームパーツを作る	120
6-3 お問い合わせフォームを作る	128
Q 練習問題	134

## CSS コーディングの基本を学ぼう 135

7-1 CSS とは	136
7-2 CSS の基本ルールを身につける	137
7-3 HTML に CSS を組み込む3つの方法	139
7-4 よく使うセレクタを知る	143
7-5 特殊なセレクタ「疑似クラス」と「疑似要素」	148
Q 練習問題	152

## 見出しや段落をスタイリングしよう 153

8-1 文字サイズを調整する	154
8-2 行間を調整する	158
8-3 書体を指定する	160
8-4 文字の色を指定する	164
8-5 見出しを背景色で装飾する	169
8-6 見出しを線で装飾する	172
8-7 見出しの先頭を装飾する	176
8-8 段落のスタイルを整える	179
Q 練習問題	184

## CSS レイアウトの基本を学ぼう 185

9-1 ボックスモデルを理解する	186
9-2 余白を調整する	190
9-3 ボックスサイズ(幅・高さ・余白)を計算する	194
9-4 web サイトの基本レイアウト	198
Q 練習問題	200

## ページ全体をレイアウトしてみよう 201

10-1 4つのwebレイアウトパターン	202
10-2 段組みでレイアウトする	204
10-3 自由に要素を配置する	213
10-4 要素を回り込ませる	221
10-5 要素の横・縦方向の位置を指定してレイアウトする	226
Q 練習問題	232

## Lesson 05

## Lesson 06

## Lesson 07

## Lesson 08

## Lesson 09

## Lesson 10

## Lesson 11

### リストとナビゲーションをスタイリングしよう .. 233

11-1 リストの装飾について学ぶ .....	234
11-2 ナビゲーションを装飾する .....	239
11-3 グローバルナビゲーションを作る .....	246
11-4 パンくずリストを作る .....	252
Q 練習問題 .....	254

## Lesson 12

### 表をスタイリングしよう ..... 255

12-1 表の装飾について学ぶ .....	256
12-2 横線・縦線のための表を作る .....	260
12-3 さまざまな装飾の表を作る .....	263
Q 練習問題 .....	272

## Lesson 13

### フォームをスタイリングしよう ..... 273

13-1 フォームの装飾について学ぶ .....	274
13-2 使いやすい装飾を施す .....	280
13-3 ボタンをデザインする .....	285
Q 練習問題 .....	290

## Lesson 14

### web サイト制作を実践してみよう ..... 291

14-1 コーディングの準備をする .....	292
14-2 トップページのHTMLを記述する .....	296
14-3 トップページのCSSを記述する .....	302
14-4 コース紹介ページを作る .....	309
14-5 アクセスページを作る .....	315
14-6 レスポンシブデザインにする .....	319

## Lesson 15

### web サイトを公開しよう ..... 327

15-1 サーバーとドメインを準備する .....	328
15-2 FTPソフトでアップロードする .....	333
15-3 webサイトのデータをチェックする .....	337
15-4 webサイトを管理する .....	343

索引 .....	348
----------	-----

# 1-3 webサイトに使用される言語

webサイトの基本となる言語はHTMLです。HTMLで文書に構造的な意味を与え、CSSで装飾やレイアウトを施します。このほかにもwebサイトを構成するための言語がいくつも存在します。代表的なものを見ていきましょう。

## HTMLとバージョン

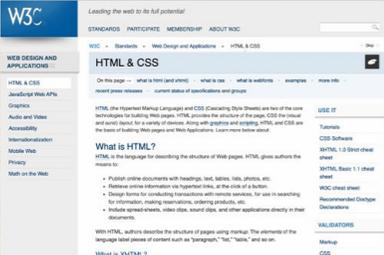
HTML (HyperText Markup Language) とは、文書をwebページとして表示させるために用いる言語です。HTMLはwebデザイナーが習得する基本の言語です。Lesson02より詳しく解説します。

HTMLにはいくつかのバージョンがあります。そのバージョンによって利用できる要素や属性に差があり、記述のルールなどが異なります。2017年12月14日に勧告されたバージョンが、「HTML 5.2」です。草案からはじまり、勧告に至るまで新たな要素や属性が追加されたり、廃止されたり、または意味や役割が変更されたりしながらバージョンアップを繰り返してきました。2021年5月以降は、WHATWGが策定しているHTMLの規格である「HTML

Living Standard」が標準となりました。

このほかにも「HTML 4.01」や「XHTML 1.0」など、いくつかのバージョンが存在しますが、すでに過去のものであり、これから学習すべきは「HTML 5」や「HTML Living Standard」をベースとしたバージョンとなります。しかし世の中には、HTML 4.01やXHTMLで作られたwebサイトが数多く存在します。webデザイナーは、新しくwebサイトを作るだけでなく、過去に作られたものを編集することも多いため、過去のバージョンについても少なからず知見を持つ必要があります。

1993年	HTML 1.0
1995年	HTML 2.0
1997年	HTML 3.2, 4.0
1999年	HTML 4.01
2014年	HTML 5.0
2016年	HTML 5.1
2017年	HTML 5.2
2021年	HTML Living Standard



**HTMLの記述ルールは誰が決める?**

HTMLのバージョンや仕様は、web技術の標準化団体であるWorld Wide Web Consortium (W3C) とWeb Hypertext Application Technology Working Group (WHATWG) によって策定されています。W3Cのwebサイトは全編が英語で書かれていますが、HTMLに関する仕様やルールなどが記載されています。

<https://www.w3.org/standards/webdesign/htmlcss.html>

COLUMN

## CSSとバージョン

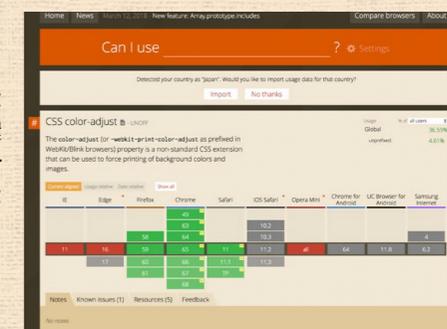
CSS (Cascading Style Sheets) とは、webページの見目や構造などのスタイルを指定するための言語です。HTMLだけでは、どうしても簡素な表示になってしまう情報に対して、色をつけたりレイアウトを施して、使い勝手を向上させるために用います。世の中のwebサイトは、ほぼすべてがCSSによって何らかの装飾がなされています。CSSには、「CSS1」「CSS2」「CSS3」という3つのバージョンが存在します。数字が大きいほど新しいバージョンを表します。

webブラウザの種類やバージョンによっては、新しいCSSが有効にならないなどのバラツキがありました。現在はブラウザのサポート状況も向上したため、積極的に新しいバージョンのCSSを利用できるようになりました。主流のバージョンは「CSS3」です。

## COLUMN

### ブラウザごとの対応状況をチェックする

webサイトを閲覧する環境は、誰しもが最新の状態であるとは限りません。それどころか、WindowsやMac、スマートフォンなど閲覧環境は多岐にわたります。特に、古いバージョンのInternet Explorerにおいては、新しいCSSの記述がうまく効かなかったりと、なにかとweb制作者の頭を悩ませるものです。海外のwebサイト「Can I Use (<https://caniuse.com/>)」では、CSSの記述がブラウザのどのバージョンから対応しているかなど、細かなチェックを行うことができます。

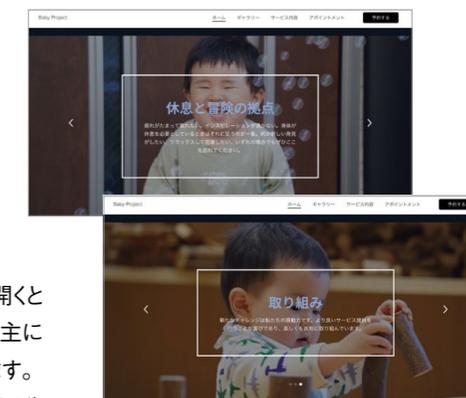


## JavaScript

JavaScriptはプログラミング言語のひとつです。HTMLやCSSと並んでとてもよく使われます。私たちが普段利用するwebサイトには、何かしらのJavaScriptが使われていることがほとんどです。JavaScriptができることは大きささまざまなのですが、ひとこと言えば、「webサイトにふるまいを与える」のが役割です。

### サイトに動的な機能を与える

たとえば複数枚の画像がスライドして切り替わったり、メニューを開くときになめらかなアニメーション効果を与えたりすることができます。主にデザインをリッチにしたり、使い心地を向上させる目的で利用されます。JavaScriptはクライアントサイド (Webサイトを閲覧しているブラウザ) がコードを受け取り、そのうえで動作します。



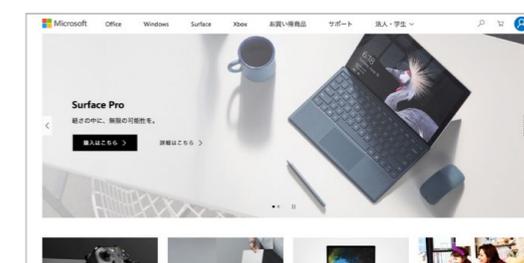
複数の画像がスライドショーで切り替わるJavaScriptの例

## jQuery

JavaScriptはとても身近な言語ですが、高度で複雑な処理をするためには、高いレベルで技術を習得しなければいけません。ですが「ゼロから書く」のではなく「すでに用意されたものを利用する」ことで、特に深い知識を得なくても取り入れることが可能です。

代表的なものが「jQuery」というライブラリです。jQueryはそもそもJavaScriptで書かれており、すでにあらゆる処理の記述がなされている「スターターキット」のようなものです。プログラムの基盤としてjQueryを利用することにより、複雑なアニメーションや表示エリアの切り替えなどが、ほとんど手間をかけずに導入できます。

多くのwebサイトでは、jQueryが導入されています。



jQueryを利用すると、画像をスライドして表示させることも簡単にできる

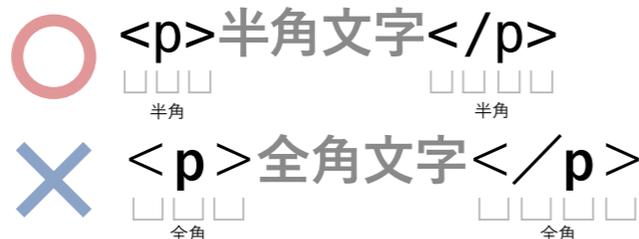
# 2-3 タグと要素と属性

タグは、HTML 中のテキストにつけられる目印です。  
タグの記述ルールと、あわせて使用する属性について学んでいきましょう。

## タグの記述ルール

### タグはすべて半角文字で記述する

HTML の中身はタグと内容で構成されますが、タグ部分の記号と英数字は、すべて半角文字で記述するのが決まりです。タグ部分に全角文字が1つでも入った場合、エラーが起きてしまうので注意しましょう。また、タグ名は大文字で記述しても認識しますが、小文字で統一するのが通例です。



### 開始タグと終了タグのセット

タグは「開始タグ」と「終了タグ」の2つをセットで記述します。開始タグの形式は、あらかじめ決められているタグ名を「<」と「>」で囲みます。終了タグは、タグ名の直前に「/」を記述します。たとえば、「<p>内容</p>」といったように開始と終了を

セットで扱います。タグという箱の中に、内容が収められていると考えるとわかりやすいでしょう。タグは入れ物のようなものです。



## 要素とは

### webページは要素で構成されている

HTML の開始タグと終了タグはセットで扱われ、そしてその中には何かしらの内容が入ります。このセットを「要素」と言います。タグが目印なのに対して、要素は内容物を含んだひとまとまりのことです。右図の例では、段落を表す p (Paragraph・段落) タグを使った要素です。

また、タグは入れ子で記述することができます。入れ子とは、要素の中にまた要素を構成することです。web ページの中身は、すべて何らかの要素が並んで、さらには入れ子になって構成されている必要があります (P.041 参照)。



### 終了タグが必要のない要素もある

一部のタグ (meta や img など) では、空 (から) 要素と呼ばれる終了タグを持たないものも存在します。通常の要素は「ここからここまで」といったように範囲指定がなされますが、空要素は範囲で扱うものではなく、ピンポイントにその場に記述するものが対象です。たとえば img 要素 (画像の挿入) は、ここからここまでといったような範囲での挿入ではなく、この場所に、ピンポイントに挿入します (P.054 参照)。

### p 要素

web ページの大部分は段落で構成されています。リストや表組みなどに当てはまらないテキストに対しては、p タグを使用します。

### CHECK!

## 属性とは

### タグに性質を与える

HTML のタグに対して、特定の「属性」をつけることができます。属性とは、個々のタグに対してなんらかの性質を与えるものです。そして属性を使用する場合、それに対する「値」もセットで記述します。

属性はそれぞれのタグに対して専用のものがあったり、すべてのタグに対して使える汎用的なものであったりさまざまです。代表的な例を紹介しましょう。

たとえば a タグは指定した範囲に対してリンクを設定するもので、「<a>詳しくはこちら</a>」のように使用します。「詳しくはこちら」という文字列に対してリンクを張るわけですが、この状態ではリンク先がどこになるのかわかりません。そこで属性と値を記述し、リンクを張ることを明示してリンク先を指定します。href という属性に対して、属性の値にリンクしたいアドレスやファイル名を記述します。



### 複数の属性を設定することも

1 つのタグに対して複数の属性を設定することが可能です。先ほどの href 属性は「a タグからリンク指定と、その値」でしたが、これにブラウザでの開き方の指定を加えてみましょう。半角スペースで区切ったあと、「target 属性」に対して属性の値に「\_blank」を指定します。

これによりリンクをクリックした際に、ブラウザは新しいタブまたは新しいウィンドウでリンク先のページを開くようになります。属性はすべてのタグに必ず使用するものではなく、それぞれのタグのルールに則って使用します。



# 3-1 画像を表示する

web ページに掲載できる情報は、テキストだけではありません。テキストだけでは表現しきれない内容や、感覚的に伝えたい情報などをweb ページに掲載したい場合には、画像を使用するのが効果的です。多くのweb サイトでは、テキストの補足情報として画像が使用されています。

## 画像を挿入する「img 要素」

画像を挿入するには「img 要素」を使用します。img 要素は、src 属性と alt 属性をセットで記述します。src 属性は、画像ファイルが置いてある場所とファイル名を指定します。alt 属性は、画像の説明を記述します。

書式	img 要素の書き方
	<pre>&lt;img src="画像の場所・ファイル名" alt="画像の説明"&gt;</pre>
	<p>src 属性      src 属性の値      alt 属性      alt 属性の値</p>

### img 要素は内容を持たない空要素

img 要素は終了タグがなく、内容を持ちません。たとえば、h1 要素だと開始タグ<h1>と終了タグ</h1>のあいだに内容(テキストや画像)が入りますが、img 要素は<img>というタグ単体です。このように、**内容を持たない要素を「空要素」と言います。**

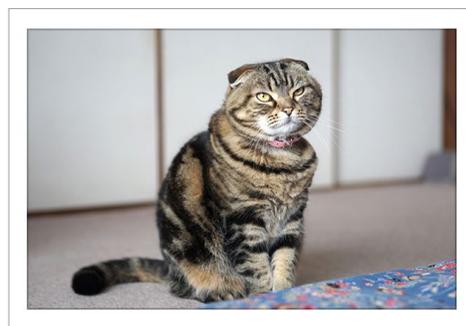
	開始タグ	内容	終了タグ
h1 要素の場合	<h1>	見出しの文言	</h1>
img 要素の場合	<img>		

## STEP 01 画像をマークアップする

Lesson 03 ▶ 3-1 ▶ 3-1-1

実際に web ページに画像を表示させてみましょう。今回は猫の画像を表示させます。学習用に用意された HTML ファイルに img 要素を記述していきます。

- [3-1-1] フォルダ内には [images] フォルダがあり、その中に画像ファイル「ring.jpg」が入っています。



画像ファイル「ring.jpg」

- [3-1-1] フォルダの HTML ファイル「3-1-1.html」をエディタで開きます。

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>画像を挿入する</title>
6 </head>
7 <body>
8
9 </body>
10 </html>

```

- <body>~</body>のあいだに、以下を記述しましょう。

```

<body>

</body>

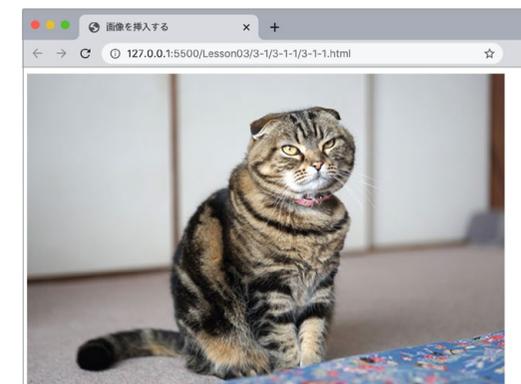
```

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>画像を挿入する</title>
6 </head>
7 <body>
8   
9 </body>
10 </html>

```

- 「images」フォルダに置かれた画像「ring.jpg」がブラウザで表示されました。



## img 要素の属性

img 要素の属性について、詳しく見ていきましょう。まずは基本の2つの属性から解説していきます。

### 画像ファイルを指定する「src 属性」

**表示する画像が保存されている場所とファイル名を指定**します。この属性は必ず記述します。画像ファイルの場所とファイル名を示す文字列(パス)を記述することで、画像ファイルを呼び出すことができます。

### パスとは?

ファイルの場所とファイル名を示す文字列を「パス」と言います。src 属性の値として記述する文字列もパスです。「/」(スラッシュ)を入れることで、フォルダの区切りを表します。たとえば「images/ring.jpg」というパスは、「images」という名前のフォルダに入っている画像「ring.jpg」を示します。パスについてはP.067から詳しく解説します。

### COLUMN

# 5-3 より高度な表組みを作る

web ページで表組みを作るときの基本を学んできました。ここまで学んだことだけでも表組みは作成できます。しかし、情報の量が多かったり複雑な内容の表組みは、より厳密にマークアップすることが必要な場合があります。表組みの応用編を学んでいきましょう。

## 行をグループ分けする要素

tableの子要素には、行をグループ分けする役割の要素があります。表組みの先頭で見出しなどに使用する「thead要素」、メインとなる内容に使用する「tbody要素」、表組みの下部でまとめの意味合いの内容などに使用する「tfoot要素」です。グループ分けをしても見た目の変化はありませんが、表組みの構造がより明確になります。また、CSSを使って装飾を行う際に利用できます。

書式 **thead要素・tbody要素・tfoot要素の書き方**

```

表の始まり
<table>
  <thead>
    <tr>
      <th>見出しセル</th>
      <th>見出しセル</th>
      <th>見出しセル</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>見出しセル</th>
      <td>通常のセル</td>
      <td>通常のセル</td>
    </tr>
    <tr>
      <th>見出しセル</th>
      <td>通常のセル</td>
      <td>通常のセル</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>見出しセル</th>
      <td>通常のセル</td>
      <td>通常のセル</td>
    </tr>
  </tfoot>
</table>
表の終わり
  
```

表の始まり

表の終わり

thead要素  
ヘッダー部分のグループ

tbody要素  
メインの内容部分のグループ

tfoot要素  
フッター部分のグループ

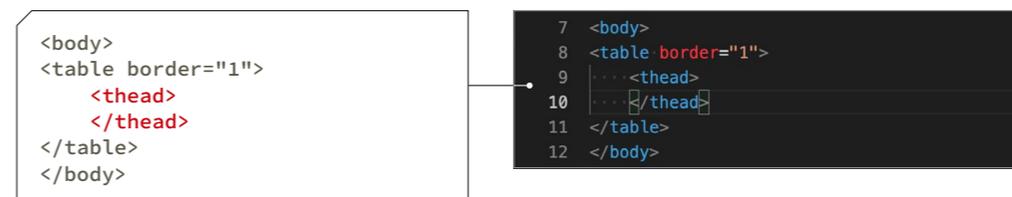
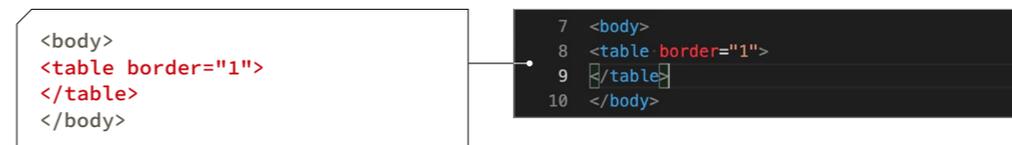
## STEP 01 行をグループ分けする

スポーツジムの料金表を作りながら、実際に表組みの行をグループ分けしてみましょう。ヘッダー部分は「料金表の項目」、メイン部分には「2つのプランの入会費と月会費の金額情報」、フッター部分には「費用の合計金額」を表示するテーブルを作成していきます。

table要素 表の枠組み



- 1 [5-3-1] フォルダのHTMLファイル「5-3-1.html」をエディタで開きます。



## Lesson 07 練習問題

## Q1 CSSとは

CSSはwebサイトに対して【 】を施すための言語です。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① レイアウトや装飾
- ② 内容の構築
- ③ 検索ロボットに対しての施策

## Q2 HTMLにCSSを組み込む方法「インライン」

HTMLにCSSを組み込む方法のひとつ「インライン」。空欄【 】に当てはまる要素は何でしょうか。

```
<h1【 】="color:#ff0000;">猫カフェ</h1>
```

## Q3 HTMLにCSSを組み込む方法「エンベッド」

HTMLにCSSを組み込む方法のひとつ「エンベッド」。「<style>」と「</style>」のあいだにCSSの指定を行うものですが、これらを記述する場所は【 】です。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① head要素内
- ② main要素内
- ③ footer要素内

## Q4 HTMLにCSSを組み込む方法「リンク」

HTMLにCSSを組み込む方法のひとつ「リンク」。CSSを記述した専用のファイルをHTMLから参照する方法です。下記は「リンク」を使用する例です。HTMLから参照しているCSSのファイル名を①～③の中から選んでください。

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

- ① stylesheet
- ② style.css
- ③ text/css

## Q5 呼び方

CSSで装飾する見た目のことを【 】と呼びます。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① カスケーディング
- ② レイアウト
- ③ スタイル

## Q6 セレクタ

スタイルを適用する箇所を指定することを「セレクタ」、どんなスタイルを施すのか指定することを【 】、装飾する値を「値」と呼びます。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① プロパティ
- ② システム
- ③ スタイル

## Q7 記述のルール

CSSを記述するには、セレクタの後ろに【 }】(中括弧)と記述します。【 }】のあいだにプロパティと値を: (コロン)で区切り、値の後ろに【 ;】を記述します。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① . (ドット)
- ② ; (セミコロン)
- ③ , (カンマ)

## Q8 タイプセレクタ

スタイルを適用する箇所を指定する「セレクタ」。段落を指定する場合、セレクタは【 p】となります。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① table
- ② h1
- ③ p

## Q9 IDセレクタ

HTMLタグに追加できるオリジナルのセレクタのひとつ「IDセレクタ」。HTMLでは開始タグの内側、「ID="名称(半角文字)"」と記述します。CSSでIDセレクタを記述する場合、「【 #】名称 {スタイル;}」となります。

空欄【 】に当てはまる言葉を答えてください。

## Q10 classセレクタ

HTMLタグに追加できるオリジナルのセレクタのひとつ「classセレクタ」。HTMLでは開始タグの内側、「class="名称(半角文字)"」と記述します。CSSでclassセレクタを記述する場合、「【 .】名称 {スタイル;}」となります。

空欄【 】に当てはまる言葉を答えてください。

## Q11 IDセレクタ・classセレクタの違い

IDセレクタとclassセレクタ、どちらもHTMLで好きな名称をつけることができますが、ひとつのHTML内で同一の名称を複数回登場させてはいけないのは【 ID】です。

空欄【 】に当てはまる言葉を答えてください。

## Q12 CSSが適用される優先度①

CSSで記述したスタイルは上から順番に読み込まれて、下に書いたものが優先されます。同じセレクタを複数記述している場合、【 最初】です。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① 最初
- ② 最後

## Q13 CSSが適用される優先度②

CSSが適用される優先順位は、セレクタによっても変わります。セレクタの中で最も優先度が高いものは【 ID】です。

下記の候補から空欄【 】に当てはまる言葉を選んでください。

- ① IDセレクタ
- ② classセレクタ
- ③ タイプセレクタ



Q1:① Q2:style Q3:① Q4:② Q5:③ Q6:① Q7:② Q8:③  
Q9:#(ハッシュ) Q10:.(ドット) Q11:IDセレクタ Q12:② Q13:①

# 8-4 文字の色を指定する

CSSを使って、さまざまな要素に色をつけたり、変更したりすることができます。webサイトにおける色の指定方法を、文字の色を変える作業を通してマスターしてみましょう。

## 文字色の基礎知識

### 文字の色の選び方

色はwebサイトの印象を決める重要な要素のひとつなので、配色は好みや思いつきで決めるのではなく、色を設定する部分(情報)の意味を考えて色を決めましょう。

### 派手だけを意識した配色

当店のスタッフ



OG3 (オジサン)    ちっち太郎    シェリー    みらく

アメリカンカールの男の子。おらかなみんなのお父さん。    セルカークレックスの男の子。運動神経抜群の甘えん坊。    メインクーンの女の子。クールで美人なお姉さま。    ミヌエットの男の子。少しぬけている天然のかわいい子。

注意事項

- お客様にははじめに、ハンドソープで手を洗いアルコールで消毒していただきます。
- 撮影は自由ですが、フラッシュなどの強い光は失明の恐れがあるので必ず消してください。
- 無理やり抱っこ、寝ているのを起こすなど猫たちの嫌がる行為はご遠慮ください。
- 店内では大声で話す、走り回るなど他のお客様や、猫たちの迷惑になる行為はご遠慮ください。

「赤色は目立つから使う」ではなく「注意を促したい・強調したいので赤色を使う」といったように、見た目で装飾を行うのではなく、機能を意識した装飾を行うことが大切です。

### 強調や注意を促すことを意識した配色

当店のスタッフ



OG3 (オジサン)    ちっち太郎    シェリー    みらく

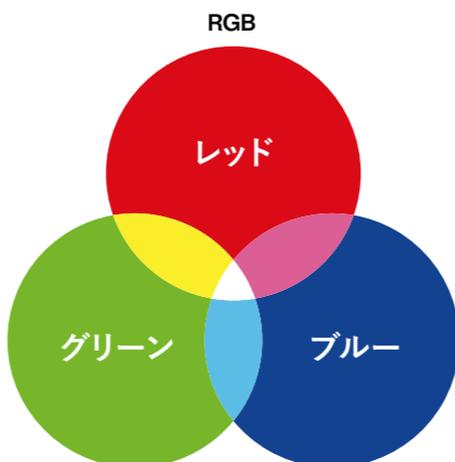
アメリカンカールの男の子。おらかなみんなのお父さん。    セルカークレックスの男の子。運動神経抜群の甘えん坊。    メインクーンの女の子。クールで美人なお姉さま。    ミヌエットの男の子。少しぬけている天然のかわいい子。

注意事項

- お客様にははじめに、ハンドソープで手を洗いアルコールで消毒していただきます。
- 撮影は自由ですが、フラッシュなどの強い光は失明の恐れがあるので必ず消してください。
- 無理やり抱っこ、寝ているのを起こすなど猫たちの嫌がる行為はご遠慮ください。
- 店内では大声で話す、走り回るなど他のお客様や、猫たちの迷惑になる行為はご遠慮ください。

### webサイトにおける色の指定

端末上で取り扱う色の表現方式(カラーモード)はいくつか種類がありますが、代表的なものは「RGB」です。RGBとは光の三原色であるレッド(Red:赤)、グリーン(Green:緑)、ブルー(Blue:青)の頭文字による略称です。パソコンやスマートフォンなどのディスプレイ機器で利用されるほか、デジタルカメラで撮影した写真データなどにもRGBが利用されています。RGBはディスプレイの発光を利用して色を表現(加法混色)します。



webサイトで色を指定する代表的な方法には「16進数」「RGB」「カラーネーム」などが挙げられます。中でも最もポピュラーな方法が16進数です。

16進数とはRGBの表現手法のひとつで、6桁の数値を1セットとして扱い、0から9までの数字とAからFまでのアルファベットの組み合わせで表現します。6桁の数値のうち、最初の2桁がR(赤)、次の2桁がG(緑)、最後の2桁がB(青)を指しており、CSSで使用する際には先頭に「#」(ハッシュ)をつけて記述します。

16進数	色
# ff 00 00	赤
# 00 00 00	黒
# ff ff ff	白
# 88 88 88	灰

### 加法混色

#### COLUMN

加法混色とは、光の三原色であるレッド、グリーン、ブルーを混合させて色を表現することを指します。それぞれの色の混合具合によって、さまざまな色を表現します。

## STEP 01 文字色を指定する

Lesson 08 ▶ 8-4 ▶ 8-4-1

文字の色を指定するには、CSSの「color」プロパティを使います。colorプロパティは、ブロックレベル、インライン(P.187で解説)どちらの性質も備えた要素にも適用できるので、見出しや段落、段落内の一部分だけに色をつけるということが可能です。

色を指定するときは、①「color: #ff0000;」と「#」で始まる6桁のカラーコードで指定する、②「color: rgb{255,0,0;）」とRGBを数値で指定する、③「color: red;」とカラーネームで指定することが可能です。ここでは、「#」で始まる6桁のカラーコードで指定する方法を試してみましょう。

書式	文字色の指定方法
セレクトタ	{ color: #16進数; }
もしくは	
セレクトタ	{ color: rgb(赤の値, 緑の値, 青の値); }
もしくは	
セレクトタ	{ color: カラーネーム; }

記述例	文字色を赤に指定する
h1	{ color: #ff0000; }
もしくは	
h1	{ color: rgb(255,0,0); }
もしくは	
h1	{ color: red; }

# 9-1 ボックスモデルを理解する

webサイトらしい見た目にするには「ボックスモデル」という概念を知っておく必要があります。この概念を理解することで、CSSのプロパティがどの部分に適用されるかがわかるようになります。意図するレイアウトを行うためにもしっかりと覚えましょう。

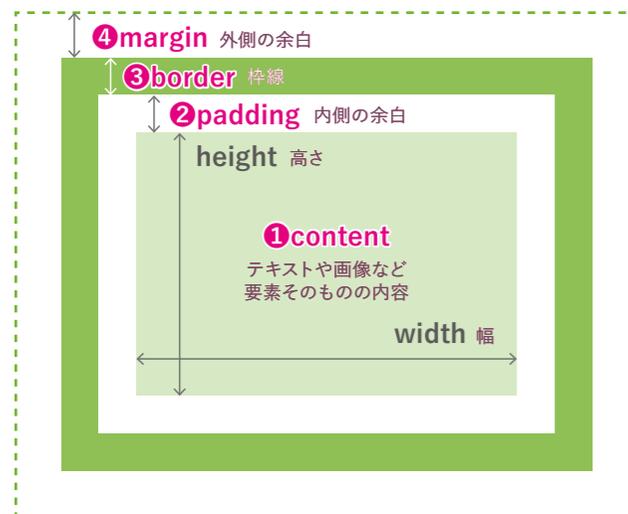
## ボックスモデルとは？

HTMLの要素は「ボックス」と呼ばれる四角形の領域を生成します。この概念をボックスモデルと呼びます。一つひとつの要素を「箱のようなもの」とイメージすると、ボックスモデルを理解しやすいでしょう。CSSでレイアウトする

ということは、ボックスの大きさを変えたり、左右に並べて配置したり、ボックス同士の間隔を調整するなどの作業になります。

### ボックスの構造

まずは、ボックスの構造から見ていきましょう。1つのボックスは次の4つの領域から成り立っています。内側の領域から順番に説明します。



#### ① content (コンテンツ)

テキストや画像など要素そのものの内容が表示される領域です。この領域のサイズは、初期値では「width」(幅)と「height」(高さ)プロパティで指定できます。

#### ② padding (パディング)

contentと枠線(border)のあいだにある余白の領域です。要素の内側の余白を取るために使います。「padding」プロパティでサイズを指定できます。

#### ③ border (ボーダー)

枠線です。paddingの外側の領域です。「border」または「border-width」プロパティでサイズを指定できます。

#### ④ margin (マージン)

要素の外側の余白の領域です。「margin」プロパティでサイズを指定できます。paddingとは対になる領域です。

「外側の余白はmargin」「内側の余白はpadding」とセットで覚えましょう。

これら各領域の境界線を「辺(edge)」と呼びます。それぞれの領域は上下左右の4辺に分けられます。それぞれにCSSでスタイルを適用できます。

## ブロックレベルとインライン

ボックスは大きく分けて「ブロックレベル」と「インライン」のいずれかの性質に分類されます。各要素は、どちらに分類されるかで初期値が決まっています。

### ブロックレベルとインラインのイメージ

ブロックレベル要素は、初期では親要素と同じ幅になります。幅や高さを指定することもできます。

インラインは幅や高さを指定できません。文字数やコンテンツの大きさでサイズが決まります。

ブロックレベル

インライン

### ブロックレベル

ブロックレベルは、上から下へ縦に並びます。これは幅のサイズに関係なく、上から順に連なっていきます。ブロックレベルには次のような特徴があります。

- width (幅) や height (高さ) を指定できる
- 幅を指定しない場合、親要素(要素自身を囲う1段階上位の要素)の content (コンテンツ) と同じ幅になる
- 上下左右の margin、padding を指定できる
- 前後に改行が入る

### インライン

インラインは、左から右へ横に並びます。基本的にブロックレベルの中で使用します。主に文章の一部として使用します。ブロックレベルより小さい部品というイメージです。インラインには次のような特徴があります。

- width (幅) や height (高さ) を指定できない(文字数やコンテンツの大きさで決まる)
- 上下の margin、padding を指定できない
- 左右の margin、padding は指定できる

### 主なブロックレベル

要素名	役割
p要素	段落
h1～h6要素	見出し
div要素	要素の囲い
ul要素・ol要素	リスト
li要素	リストの項目

### 主なインライン

要素名	役割
a要素	リンク
img要素	画像
span要素	範囲の定義
strong要素	重要性を表す

### COLUMN

#### HTML5では分類は廃止されたが、要素の性質は変わらない

HTML4までのバージョンでは、要素は「ブロックレベル要素」と「インライン要素」という、2つの要素カテゴリーのいずれかに分類されていました。HTML5では廃止こそされましたが、性質は変わりません。CSSを使う際の挙動に大きな差があるため、その要素がどの性質を持っているかを把握する必要があります。CSSでレイアウトをするうえで、この性質の理解は欠かせません。

## STEP 01 content部分を作ってみる

Lesson 09 ▶ 9-1 ▶ 9-1-1

HTML要素のボックスモデルを理解したところで、ボックスモデルのcontentに当たる部分を実際につけてみましょう。幅の指定にはwidthプロパティ、高さの指定にはheightプロパティを使用します。HTML要素の幅と高さ

をCSSで指定する基本をマスターしてみましょう。

HTML要素の幅を指定する形式は「セクタ { width: 数値と単位; }」となり、高さの指定する形式は「セクタ { height: 数値と単位; }」となります。

書式	幅の指定方法
	セクタ { width: 数値と単位; }

書式	高さの指定方法
	セクタ { height: 数値と単位; }

記述例	main要素の幅を指定する
	main { width: 980px; }

記述例	header要素の高さを指定する
	header { height: 200px; }

# 10-1 4つのwebレイアウトパターン

あなたが普段よく閲覧するwebサイトは、どのような見た目でしょうか。文章や写真はどのように配置されているでしょうか。webサイトはそれぞれ異なる見目をしていますが、よく使われるレイアウトというものが存在します。まずはwebサイトのレイアウトの「型」について学んでいきましょう。

## 1カラムのレイアウト

**カラムとは「段組み」**を指します。誌面で文章を読みやすいように段組みにして整えるように、webページにおいても段組みを行います。1カラムのレイアウトの場合、段組みは1つです。つまり、段組みをしない状態です。

### マルチデバイス対応ページに向いている

1カラムのメリットは、内容を大きなサイズで掲載できる点です。幅を広く取れるために文字サイズを大きくしたり、画像などの各パーツのサイズも大きめに扱えます。シンプルな構造のため、スマートフォンで見てもパソコンとあまり変わらない状態で閲覧することができます。逆に言うと、作り手側ではスマートフォンでの閲覧を意識してwebページを作る場合に、レイアウトしやすい手法です。

1カラムは業種を問わず人気のあるレイアウトです。情報を大きなサイズで表現できるので、コンテンツを印象的に表現することができます。余白を活かした高級感のあるデザインにしたり、空いているスペースにイラストや写真などを配置して華やかさを演出するなどの装飾が行いやすい、カスタマイズ性に富んだレイアウトです。

## 2カラムのレイアウト

2カラムは、2列の段組みレイアウトです。主役となる情報を掲載する部分である「メインコンテンツ」と、補足情報を掲載する部分の「サイドバー」を並べて配置します。昔から国内のwebサイトでよく使われている手法で、まだまだ見かけることも多いレイアウトです。右サイドバーもしくは左サイドバーに、ナビゲーションやバナーを配置するケースが多く見られます。情報を並列に並べることにより、たくさんの情報を一度に見せることができます。

### 特徴的なサイドバー

近年のwebサイトは「情報の質」が重要視されています。補足情報を掲載するサイドバーは、雑音となり得てしまうこともあります。そのため、2カラムのレイアウトは現在の主流からは少し外れています。しかし「情報の種類が複数ある」「情報が多岐にわたる」など、コンテンツの見せ方によっては有効なレイアウト手法です。

## 3カラムのレイアウト

3カラムは3列の段組みレイアウトです。メインコンテンツに加えてサイドバーを2つ有するレイアウトを指すことが多いです。多種多様なコンテンツを含んだwebサイトに適しています。

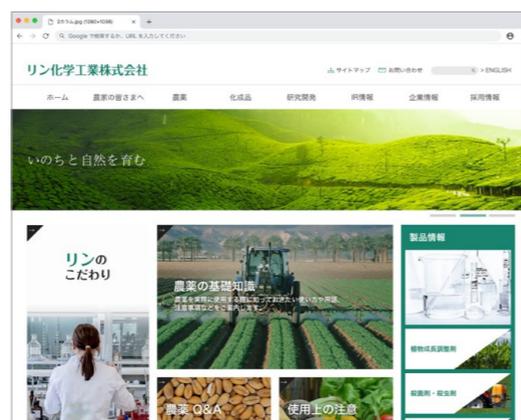
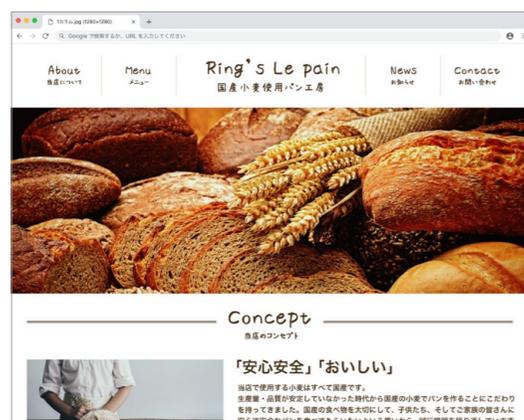
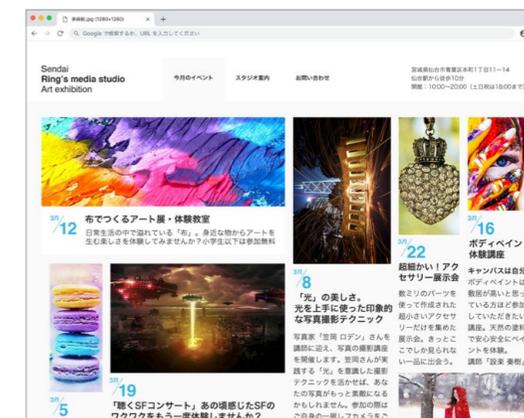
### 多くの情報を伝達するサイト向き

たとえば「Yahoo! JAPAN」をはじめとする検索エンジンやポータルサイト、「CNN」などのニュースサイトによく用いられる手法です。主役となる内容の左右にサイドバーを持つレイアウトは、たくさんの情報を伝達することを目的としたサイトに適しています。さまざまな情報を扱うニュース系のサイトをはじめ、オンラインストアでも多様な商品ジャンルを扱っていたり、ブログサイトなど記事の一覧や広告を多く掲載する必要があるサイトでも採用されています。



## 多段組みのレイアウト

ここまでカラム（段組み）レイアウトを紹介してきましたが、必ずしも型にはめて作らないといけないわけではありません。デザイナーのアイデアや情報の種類によっては自由なレイアウトも行います。3カラムよりも段組みが多い「多段組みのレイアウト」と呼ばれるデザイン手法もあります。多段組みは、ひとつの範囲の中で内容を複数のブロックとして表現することを指します。下に掲載した事例のwebサイトは、サイトの1ページをひとつの範囲と捉え、内容を大小さまざまなブロックで分けて表現しています。各コンテンツの見た目の大きさが違うことでデザインにメリハリが生まれ、それぞれに注目しやすくなるメリットがあります。



### COLUMN

#### トレンドのレイアウト

毎年、世界的にデザインやレイアウトのトレンドが存在します。2021年は「パララックス・アニメーション」というスクロール動作に応じてさまざまな要素を動かし、コンテンツに動きや立体感を出す、視覚に訴える演出手法を取り入れたレイアウトがよく見られました。このような情報は、海外のwebサイトや海外のブログをまとめた記事でよく紹介されています。そのようなデザイン系のブログをこまめにチェックすると、最新のトレンドをつかめることができます。



# 14-2 トップページHTMLを記述する

一般的にwebサイトのデータ制作で最初に取りかかるのは、トップページの作成です。トップページが完成したら、それをフォーマットとして他のページのデータも作っていきます。まずはトップページのHTMLを記述していきましょう。

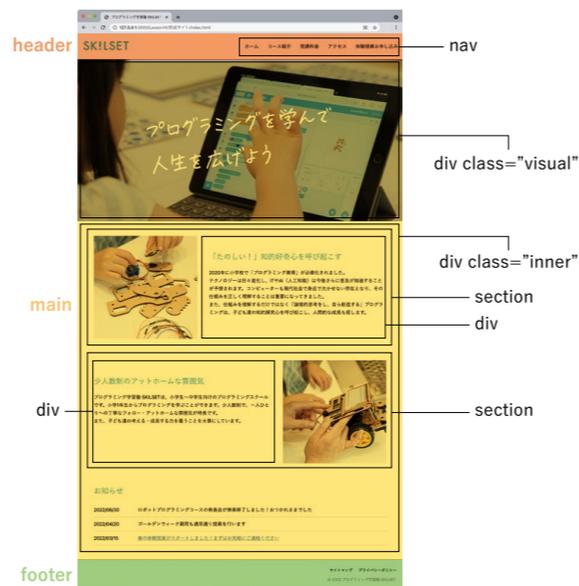
## トップイメージを確認する

Lesson14 ▶ 完成サイト

[完成サイト]フォルダの中にあるindex.htmlをブラウザで表示しましょう。これがトップページが完成した状態です。トップページではwebページの全体のレイアウトをします。webページをレイアウトするための主要な要素やCSSの指定方法をおさらいしていきます。のちほど、他ページに

も同様の全体レイアウトの設定を適用させていきます。webサイトのレイアウトの骨格を作っていくイメージです。トップページはheader要素・main要素・footer要素を使用して大きく3分割します。その後header要素内にサイト見出しやナビゲーションを記述したり、main要素内に文章や画像を配置してページのコンテンツを作っていきます。

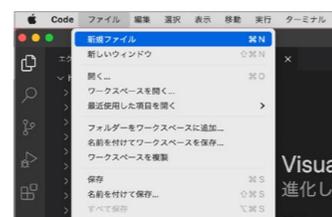
完成イメージ



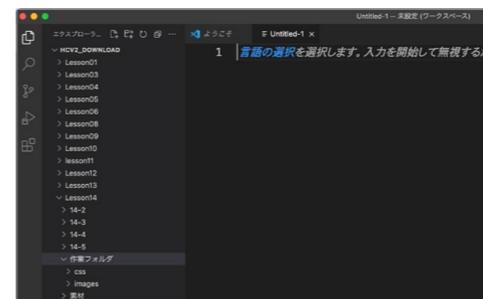
## STEP 01 head要素をマークアップする

Lesson14 ▶ 素材 / Lesson14 ▶ 14-2

1 HTMLファイルを作成します。Visual Studio Codeのメニューから[ファイル]>[新規ファイル]をクリックします。



2 名称未設定ファイルが開きました。Visual Studio Codeのメニューから[ファイル]>[保存]をクリックし、14-1で作成した[作業フォルダ]の中に[index.html]という名前で保存します。



3 まずはhead要素を記述していきましょう。1行目にDOCTYPE宣言を記述し、HTML要素を記述します。

```
<!DOCTYPE html>
<html lang="ja">
</html>
```

```
1 <!DOCTYPE html>
2 <html lang="ja">
3
4 </html>
```

4 HTML要素の内側にhead要素を記述します。

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4
5 </head>
6 </html>
```

5 head要素の中にmeta要素を書いていきましょう。まずは文字コードを指定するmeta要素を記述します。charset属性には「UTF-8」を指定します。

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5 </head>
6 </html>
```

6 webサイトの概要を示すmeta要素を記述します。name属性の値には「description」を指定し、content属性の値にはページの説明文を指定します。テキストは[素材]>[index.txt]にあります。

```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="description"
6     content="東京都新宿区のプログラミング学校「プログラミング学習塾 SKILSET」。少人数制で、一人ひとりへの丁寧なフォロー・アットホームな雰囲気が特長です。子ども達の考える・成長する力を養います。">
7 </head>
```

7 title要素を記述します。テキストは[素材]>[index.txt]を利用してOKです。

```
3 <head>
4   <meta charset="UTF-8">
5   <meta name="description"
6     content="東京都新宿区のプログラミング学校「プログラミング学習塾 SKILSET」。少人数制で、一人ひとりへの丁寧なフォロー・アットホームな雰囲気が特長です。子ども達の考える・成長する力を養います。">
7   <title>プログラミング学習塾 SKILSET | 東京都新宿区のプログラミング学校</title>
8 </head>
```