

# 作って学ぶ HTML+CSS グリッドレイアウト

エビスコム 著

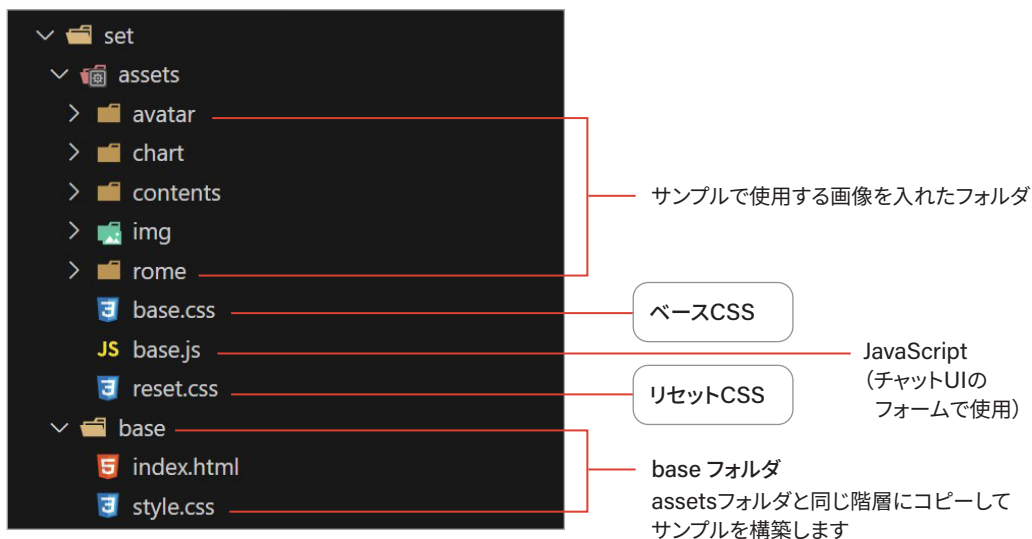
副読本

サンプルの構築に使用するセット



CSS グリッドを使ったレイアウト構築に集中して作業できるようにするため、本書ではサンプル作成時に使用するセットを用意しています。各サンプルは `assets` フォルダと同じ階層に `base` フォルダをコピーし、`base` フォルダ内の `index.html` と `style.css` にコードを追加して構築していきます。

`assets` フォルダ内にはサンプルで使用する画像と JavaScript (`base.js`) に加えて、CSS の初期設定を行う「リセット CSS (`reset.css`)」と「ベース CSS (`base.css`)」を用意してあります。本 PDF ではこれらで設定した CSS についてまとめます。



```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="../assets/reset.css" />
    <link rel="stylesheet" href="../assets/base.css" />
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <!-- ここから -->

    <!-- ここまで -->
  </body>

  <script src="../assets/base.js"></script>
</html>
```

baseフォルダ内のindex.html  
にはリセットCSSとベースCSS  
を適用してあります

index.html

# リセットCSS (reset.css)

assets フォルダに収録したリセット CSS (reset.css) は、書籍 P.51 で紹介した「Tailwind CSS」で使用されているリセット CSS を元に、サンプルの構築に必要な設定を取り出して構成したものです。Tailwind CSS のリセット CSS をそのまま利用してサンプルを構築していくこともできます。

## Tailwind CSSのリセットCSS

<https://github.com/tailwindlabs/tailwindcss/blob/master/src/css/preflight.css>

## assetsフォルダに収録したリセットCSS (reset.css)

```
@charset "utf-8";

/* =====

リセット CSS

Tailwind CSS のリセット CSS (preflight.css) を元にサンプルの構築に必要な設定で構成

Tailwind CSS
- https://github.com/tailwindlabs/tailwindcss
- https://github.com/tailwindlabs/tailwindcss/blob/master/LICENSE

===== */

*,
::before,
::after {
  box-sizing: border-box;
  border-color: currentcolor;
  border-style: solid;
  border-width: 0;
}

html {
  font-family: ui-sans-serif, system-ui, -apple-system, BlinkMacSystemFont,
    "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif,
    "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "Noto Color Emoji";
  line-height: 1.5;
  -moz-tab-size: 4;
  tab-size: 4;
  -webkit-text-size-adjust: 100%;
}
```

すべての要素の横幅 (width) と高さ (height) にパディングやボーダーを含めて処理するように指定。ボーダーは非表示にされています。

フォント、行の高さ、タブサイズを指定。フォントはシステムフォントを使用し、iOSのフォントサイズの自動調整は無効化するように指定されています

```
body {  
  margin: 0;  
  line-height: inherit;  
}
```

ページまわりの余白を削除し、行の高さは<html>の設定を継承するように指定

```
h1,  
h2,  
h3,  
h4,  
h5,  
h6 {  
  font-size: inherit;  
  font-weight: inherit;  
}
```

見出し<h1>～<h6>のフォントサイズと太さは親要素の設定を継承するように指定

```
a {  
  color: inherit;  
  text-decoration: inherit;  
}
```

リンク<a>の色と装飾は親要素の設定を継承するように指定

```
code,  
kbd,  
samp,  
pre {  
  font-family: ui-monospace, SFMono-Regular, Menlo, Monaco, Consolas,  
    "Liberation Mono", "Courier New", monospace;  
  font-size: 1em;  
}
```

コード関連<code>や<pre>などのフォントとフォントサイズを指定

```
table {  
  text-indent: 0;  
  border-collapse: collapse;  
  border-color: inherit;  
}
```

テーブル関連のスタイルをリセット

```
/* フォーム */  
button,  
input,  
optgroup,  
select,  
textarea {  
  margin: 0;  
  padding: 0;  
  font-family: inherit;  
  font-size: 100%;  
  font-variation-settings: inherit;  
  font-weight: inherit;  
  font-feature-settings: inherit;  
  line-height: inherit;  
  color: inherit;  
}
```

フォーム関連のマージン、パディング、フォント、色のスタイルをリセット

```
button,  
select {  
  text-transform: none;  
}
```

<button>と<select>でtext-transformが継承されないように指定

```
button,  
[type="button"],  
[type="reset"],  
[type="submit"] {  
  -webkit-appearance: button;  
  background-color: transparent;  
  background-image: none;  
}
```

<button>や<input>が構成するボタンのスタイルをリセット

```
button,  
[role="button"] {  
  cursor: pointer;  
}
```

ボタンのカーソルの形状を指定

```
textarea {  
  resize: vertical;  
}
```

<textarea>は縦方向のリサイズのみを許可

```
/* デフォルトのマージンやパディングを削除 */  
blockquote,  
dl,  
dd,  
h1,  
h2,  
h3,  
h4,  
h5,  
h6,  
hr,  
figure,  
p,  
pre {  
  margin: 0;  
}  
  
fieldset {  
  margin: 0;  
  padding: 0;  
}  
  
legend {  
  padding: 0;  
}
```

標準で挿入されるマージンやパディングを削除

```
ol,  
ul,  
menu {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}
```

標準で挿入されるマージン、パディング、リストマークを削除

```
dialog {  
  padding: 0;  
}
```

/\* 埋め込み要素をブロック要素に指定（要素の下に小さなスペースが入るのを防ぐ） \*/

```
img,  
svg,  
video,  
canvas,  
audio,  
iframe,  
embed,  
object {  
  display: block;  
  vertical-align: middle;  
}
```

画像やボタンなどの下に入る余白を削除  
(余白が入るロジックや対処方法について詳しくは書籍のP.175~180を参照してください)

/\* 画像とビデオを可変サイズに指定 \*/

```
img,  
video {  
  max-width: 100%;  
  height: auto;  
}
```

<img>と<video>の横幅を可変サイズにし、高さは横幅に対してオリジナルの縦横比を維持したサイズにするように指定

## ベースCSS (base.css)

assets フォルダに収録したベース CSS (base.css) は、サンプルで使用するフォントや色などの基本的なスタイルを用意したものです。クラスを指定することで利用できるようにしてあります。

### assetsフォルダに収録したベースCSS (base.css)

```
@charset "utf-8";

/* =====

ベース CSS

===== */

/* フォント */
@import url("https://fonts.googleapis.com/css2?family=Noto+Sans+JP:wght@400;700&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Gloria+Hallelujah&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700&display=swap");
```

Googleフォント (<https://fonts.google.com/>) から  
下記フォントを使用するための設定をインポート

- Noto Sans JP
- Gloria Hallelujah
- Inter

```
body {
  font-family: Inter, "Noto Sans JP", sans-serif;
}
```

使用フォントを欧文はInter、  
日本語はNoto Sans JPに指定

```
/* カラー */
:root {
  --color-accent: #ffcd6d;
  --blue-smoke: #b4d0d9;
  --dark-blue-gray: #374151;
  --light-blue-gray: #e2e8f0;
}
```

繰り返し使用する色     を指定


```
/* 背景色 */
.background-ivory {
  background-color: #f4f0e9;
}
```







 背景色を象牙色にするクラス

```
.background-whitesmoke {
  background-color: #f5f5f5;
}
```

 背景色をホワイトスモークにするクラス

```
.background-bluesmoke {
  background-color: var(--blue-smoke);
}
```

 背景色をブルースモークにするクラス

|   |   |
|---|---|
| <pre>.background-bluegray {<br/>  background-color: var(--light-blue-gray);<br/>}</pre>   |  背景色を淡い青灰色にするクラス |
| <pre>.background-white {<br/>  background-color: #fff;<br/>}</pre>  |  背景色を白色にするクラス    |
| <pre>.background-lightgreen {<br/>  background-color: lightgreen;<br/>}</pre>   |  背景色を明るい緑色にするクラス |
| <pre>.background-green {<br/>  background-color: #eefde;<br/>}</pre>  |  背景色を緑色にするクラス    |
| <pre>.background-red {<br/>  background-color: #fff0e5;<br/>}</pre>   |  背景色を赤色にするクラス    |
| <pre>.background-yellow {<br/>  background-color: #feffc;<br/>}</pre>   |  背景色を黄色にするクラス    |
| <pre>/* パディングを挿入するクラス */<br/>.padding-small {<br/>  padding: 16px;<br/>}</pre>  | パディングを挿入するクラス   |
| <pre>/* パディングを削除するクラス */<br/>.no-padding {<br/>  padding: 0 !important;<br/>}</pre>   | パディングを削除するクラス   |
| <pre>/* 区切り */<br/>hr {<br/>  margin: 0;<br/>  padding-block: 40px;<br/>  border-width: 0;<br/>}</pre>  | サンプルの間に入れる区切り<hr>用のスタイル   |
| <pre>/* スクリーンリーダー用（画面に表示しない） */<br/>.sr-only {<br/>  position: absolute;<br/>  overflow: hidden;<br/>  width: 1px;<br/>  height: 1px;<br/>  margin: -1px;<br/>  padding: 0;<br/>  clip: rect(0, 0, 0, 0);<br/>  white-space: nowrap;<br/>  border-width: 0;<br/>}</pre> | スクリーンリーダー用のコンテンツを画面に表示しないようにするクラス<br>※Tailwind CSSの.sr-onlyクラスと同じ設定を使用                            |



/\* 配置先に合わせたサイズにする \*/

```
.fill {
  width: 100%;
  height: 100%;
}
```

横幅と高さを100%にするクラス

/\* 画像を配置先に合わせたサイズにして切り抜く \*/

```
.img-fill {
  width: 100%;
  height: 100%;
  object-fit: cover;
}
```

横幅と高さを100%にして  
画像を切り抜くクラス

/\* 角丸にする \*/

```
.radius {
  border-radius: 8px;
}
```

角丸(半径8px)にするクラス

/\* ブーツアイコンを付けるクラス \*/

```
[class*="boots-logo"] {
  padding: 32px 10px;
  font-size: clamp(10px, 0.0286rem + 2.5445vw, 20px); /* 画面幅 375px ~ 768px で 10px ~ 20px に変更 */
  font-weight: bold;
  color: #fff;
  text-align: center;
  background-color: #511524;
}
```

```
[class*="boots-logo"]::before {
  content: url("img/boots.svg");
  display: block;
}
```

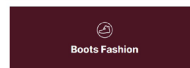
```
.boots-logo-large {
  font-size: 40px;
}
```

/\* コードのスタイル \*/

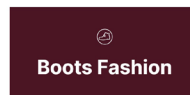
```
pre {
  overflow-x: auto;
  padding: 16px;
  font-size: 85%;
  line-height: 1.45;
  color: white;
  background-color: #222;
  border-radius: 6px;
}
```

#### ブーツアイコンを付けるクラス

class属性の値に「boots-logo」を含む要素にブーツアイコン(img/boots.svg)を付加し、背景を赤褐色にします。テキストは白色の中央揃えで、フォントサイズを画面幅に合わせて10px~20pxに変化させます。boots-logo-largeクラスの場合はフォントサイズを40pxにします。



class="boots-logo"  
を指定したもの



class="boots-logo-large"  
を指定したもの

#### <pre>でマークアップしたコードのスタイル

overflow-xで必要に応じて横スクロールバーを表示するように指定。テキストはひとまわり小さくして白色に、背景は黒色にして角丸のスタイルにしています。

```
<button aria-label="オプション" class="action show-on-mobile">
  <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" fill="none" viewBox="0 0 24 24">
    <path
      stroke-linecap="round"
      stroke-linejoin="round"
      d="M12 3v3m0 18v3m9-3h3m-3 9h3M4.5 12h3m3 0h3m-6 0h3" />
  </svg>
</button>
```

```
/* =====  
 * テキスト  
 * ===== */
```

```
.text-xsmall {  
  font-size: 12px;  
}
```

フォントサイズを12pxにするクラス

```
.text-small {  
  font-size: 14px;  
}
```

フォントサイズを14pxにするクラス

```
.text {  
  line-height: 1.8;  
}
```

行の高さを1.8にするクラス

```
@media (width < 500px) {  
  .text-xsmall-mobile, .text-xsmall-mobile * {  
    font-size: 10px !important;  
  }  
}
```

モバイルでのフォントサイズを10pxにするクラス

```
.heading-center {  
  margin-block: 40px;  
  font-size: 24px;  
  font-weight: bold;  
  text-align: center;  
}
```

上下マージンで余白を確保し、24pxの太字で中央揃えにする見出し用のクラス

```
.heading-large {  
  margin-block: 0 20px;  
  font-size: 28px;  
  color: #475569;  
}
```

下マージンで余白を確保し、28pxのグレーの色にする見出し用のクラス

```
/* =====  
 * アイコン  
 * ===== */
```

```
:where(.vertical, .horizontal, .search-01, .search-02, .header) svg {  
  width: 24px;  
  height: 24px;  
}
```

```
:where(.with-icon-vertical, .with-icon-horizontal, .with-icon-double, .with-icon-auto) svg {  
  width: 32px;  
  height: 32px;  
}
```

5.2、5.4、5.5で作成するパーツ内のSVGアイコンを24pxに、5.3で作成するパーツ内のSVGアイコンを32pxに指定

```
/* =====
 * フォーム
 * ===== */
```

```
[type="text"],
[type="email"],
[type="url"],
[type="tel"],
[type="search"],
[type="password"],
select,
textarea {
  padding: 12px;
  background-color: white;
  background-image: none;
  border: solid 1px #888;
  border-radius: 4px;
}
```

```
[type="submit"],
[type="reset"] {
  padding: 12px;
  border-radius: 4px;
}
```

```
::placeholder {
  color: #999;
}
```

```
:where(input, select, textarea, button):focus-visible {
  outline: 2px solid black;
}
```

```
/* =====
 * ボタン
 * ===== */
```

```
.btn {
  padding: 16px 40px;
  color: white;
  background-color: black;
  border-radius: 4px;
}
```

```
.btn-accent {
  margin-block-end: 16px;
  padding: 16px 40px;
  color: black;
  background-color: var(--color-accent);
  border-radius: 8px;
  box-shadow: 0 4px 4px rgb(0 0 0 / 20%);
}
```

### フォーム関連のスタイル

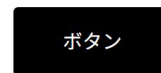
フォームの入力欄はリセットCSSでスタイルがリセットされているので、背景を白色にしてグレーの角丸ボーダーで囲むように指定。パディングと角丸のサイズは送信・リセットボタンにも適用し、入力欄とサイズを揃えています。さらに、プレースホルダのテキストは薄いグレーに、フォーカス時のアウトラインは黒色に指定しています。



フォーカス時の表示

ボタンには必要に応じてボタン関連のクラスを追加してスタイリングします

### 黒いボタンの形にするクラス



### 影付きの黄色いボタンの形にするクラス

下マージンでは影のスペースを確保し、オーバーフローで影がカットされるのを防ぎます



```
.btn-icon-black {
  color: white;
  background: black;
}

.btn-small {
  padding: 4px 12px;
  font-size: 14px;
  color: white;
  background-color: black;
  border: solid 1px black;
  border-radius: 4px;

  @media (width <=600px) {
    font-size: 10px;
  }
}

.btn-small-outline {
  padding: 4px 12px;
  font-size: 14px;
  color: black;
  background-color: transparent;
  border: solid 1px black;
  border-radius: 4px;

  @media (width <=600px) {
    font-size: 10px;
  }
}

/* =====
 * スクロールバー
 * ===== */

@media (pointer: fine) {
  .scrollbar {
    /* スクロールバーの太さと背景色 */
    &::-webkit-scrollbar,
    &::-webkit-scrollbar-corner {
      width: 8px;
      height: 8px;
      background: transparent;
    }

    /* 矢印ボタンのスペースを削除 */
    &::-webkit-scrollbar-button {
      width: 0;
      height: 0;
    }

    /* ハンドルの最小の長さ、色、角丸半径 */
    &::-webkit-scrollbar-thumb {
      min-width: 48px;
      min-height: 48px;
      background: lightgray;
      border-radius: 4px;
    }
  }
}
```

検索フォームの送信ボタンを黒色にするクラス

テキストやアイコンを白色に、背景を黒色にするクラスです。検索フォームの送信ボタンのスタイリングに使用します



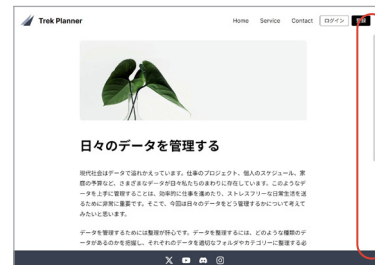
小さいボタンの形にするクラス

パディングやフォントサイズを小さくしたボタンで、背景を黒くするクラスと、アウトラインにするクラスを用意。モバイルではさらにフォントサイズを小さくしています



スクロールバーの形状を指定するクラス

スクロールバーを太さ8pxのグレーの形状にするクラスです。@mediaでデスクトップ環境のみに適用しています



```

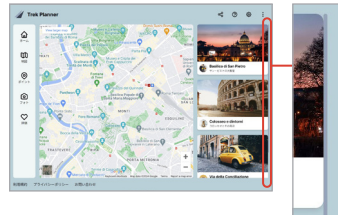
    }
  }

  .holy .scrollbar {
    /* スクロールバーの太さ */
    &::-webkit-scrollbar,
    &::-webkit-scrollbar-corner {
      width: 4px;
      height: 4px;
    }

    /* ハンドルの色 */
    &::-webkit-scrollbar-thumb {
      background: #94a3b8;
    }
  }
}

```

聖杯レイアウト内の  
スクロールバーの形状を指定するクラス  
「5.14 聖杯レイアウト」ではスクロールバー  
を太さ4pxの青灰色にしています



```

/* =====
 * ヘッダー関連
 * ===== */

```

```

/* ロゴ画像 */
.logo img {
  width: 48px;
  height: 48px;

  @media (width <=700px) {
    width: 32px;
    height: 32px;
  }
}

```

```

/* サイト名 */
.site {
  font-size: 20px;
  font-weight: bold;

  @media (width <=700px) {
    font-size: 16px;
  }
}

```

```

/* アクションボタン */
.action > svg {
  width: 24px;
  height: 24px;

  @media (width <=700px) {
    width: 20px;
    height: 20px;
  }
}

```

画像やテキストのサイズを調整するクラス  
ロゴ画像、サイト名、アクションボタンのサ  
イズを調整するクラスで、「5.5 ヘッダー」で  
使用しています。モバイルではひとまわり  
小さいサイズにしています



```

/* ナビゲーションメニュー */
.nav {
  display: grid;
  grid-auto-flow: column;
  gap: 8px;
  justify-content: start;
}

/* 検索フォーム */
.search {
  display: grid;
  grid-template-columns: auto 1fr;
  gap: 8px;
  align-items: center;

  > button {
    grid-column: 1;
    grid-row: 1;
  }

  > input {
    grid-column: 1 / -1;
    grid-row: 1;
    max-width: 240px;
    padding-block: 8px;
    padding-left: 48px;
    background-color: var(--light-blue-gray);
    border: none;
  }
}

/* モバイルで表示 */
.show-on-mobile {
  @media (width > 980px) {
    display: none !important;
  }
}

/* モバイルで隠す */
.hide-on-mobile {
  @media (width <= 980px) {
    display: none !important;
  }
}

```

#### ナビゲーションメニューのクラス

リンクを横に並べてメニューの形にするクラス。「5.2 スタックレイアウト」の横並びのCSSを使用し、ガター (gap) を8pxにしたものです

Home Service Contact

Home Service Contact

#### 検索フォームのクラス

検索フォームの形にするクラス。「5.4 検索フォーム」の入力欄の中にアイコンボタンを入れたパターンのCSSを使用したものです。ここでは<input>の背景を淡い青灰色にして、ボーダーを削除しています

Q Search

Q Search

画面幅が980pxより大きい場合は非表示にするクラス

画面幅が980px以下の場合は非表示にするクラス

```
/* =====
 * セクション
 * ===== */
```

「5.6 セクション」や「5.13 フルブリードレイアウト」で使用するクラス  
class属性の値が「section-」で始まる要素に適用

```
[class^="section-"] {
```

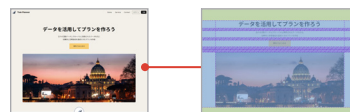
```
/* セクション内の上下の余白 */
padding-block: 96px;
```

セクション内の上下の余白

各セクション内には96pxの上下パディングを挿入

```
&[class^="section-01"] + [class^="section-02"] {
  padding-block-start: 112px;
}
```

```
&[class^="section-02"] + [class^="section-02"] {
  padding-block-start: 0;
}
```



section-01に続くsection-02では上パディングを112pxに変更

section-02に続くsection-02では上パディングを削除

```
/* 横並びセクションのカスタマイズ
   (ランディングページ用) */
&[class^="section-02"].lp-style {
  grid-template-rows: 1fr repeat(3, auto) 2fr;
```

```
@media (width <= 768px) {
  grid-template-rows: repeat(4, auto);
}
```

横並びセクションのカスタマイズ

.lp-styleクラスがある場合、横並びセクション(書籍P.240)のグリッドの1行目と5行目の高さをカスタマイズ



```
/* 重ね合わせセクションのカスタマイズ
   (ランディングページ用) */
&[class^="section-03"].lp-style {
  grid-template-rows: 160px repeat(3, auto) 112px;
  padding-block: 0;
  color: black;
```

```
.photo {
  filter: none;
  object-position: right center;
  border-radius: 0;
}
```

重ね合わせセクションのカスタマイズ

.lp-styleクラスがある場合、重ね合わせセクション(書籍P.242)のグリッドの1行目と5行目の高さ、画像の色合いなどをカスタマイズ



```
/* セクション内の見出し、文章、ボタン、画像のスタイル */
.heading {
  font-size: min(32px, 6vw);
  font-weight: bold;
  line-height: 1.4;
}

& h1.heading {
  font-size: min(48px, 6vw);
}

.text {
  max-width: 26em;
  font-size: min(16px, 3.5vw);
}

.button {
  max-width: 300px;
  padding: 16px 40px;
  font-size: min(16px, 3.5vw);
  color: black;
  background-color: var(--color-accent);
  border-radius: 4px;
}

.photo {
  border-radius: 8px;
}

}
```

セクション内のアイテムをスタイリングするクラス  
セクション内の見出し、文章、ボタン、画像をスタイリングするクラスです。フォントサイズ、最大幅、角丸の半径などを指定しています。  
たとえば、見出しはmin()で最大フォントサイズを32pxにした上で、<h1>でマークアップした場合は48pxまで大きくなるようにしています

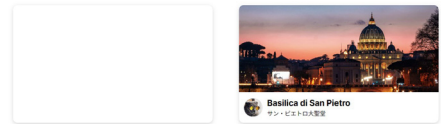


```
/* =====
 * カード
 * ===== */
```

```
[class^="card-"] {
  /* カードの形状 */
  overflow: hidden;
  background-color: white;
  border-radius: 8px;
  box-shadow:
    0 2px 8px -1px rgb(0 0 0 / 20%);
}
```

「5.7 カード」で使用するクラス  
class属性の値が「card-」で始まる要素に適用

カードの形状を指定するスタイル  
カードの形状は白色の角丸で、影をつけたスタイルにしています





```
/* カード内のタイトル、サブタイトル、アバターのスタイル */
.title {
  font-size: 14px;
  font-weight: bold;
}

.subtitle {
  font-size: 10px;
}

.avatar img {
  width: 32px;
  height: 32px;
  background-color: #f1f5f9;
  border: solid 1px #cbd5e1;
  border-radius: 50%;
  object-fit: cover;
}
```

#### カード内のアイテムをスタイリングするクラス

カード内のタイトル、サブタイトル、アバターをスタイリングするクラスです。フォントサイズを調整し、アバター画像は円形に切り抜きます



アバター .avatar

サブタイトル .subtitle

タイトル .title

```
/* =====
 * データカード
 * ===== */
```

```
[class^="datacard-"] {
  /* カードの形状 */
  overflow: hidden;
  border-radius: 8px;
  box-shadow: 0 2px 8px -1px rgb(0 0 0 / 20%);

  /* カード内のタイトル、サブタイトル、データ、アイコンのスタイル */
  .title {
    font-size: 14px;
    font-weight: bold;
    color: #727272;
  }

  .subtitle {
    font-size: 12px;
  }

  .data {
    font-size: 28px;
    font-weight: 500;

    &.text-small {
      font-size: 16px;
    }
  }

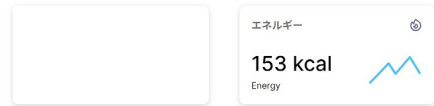
  .icon {
    color: #64748b;
    & svg {
      width: 20px;
      height: 20px;
    }
  }
}
```

#### 「5.8 データカード」で使用するクラス

class属性の値が「datacard-」で始まる要素に適用

#### データカードの形状を指定するスタイル

データカードの形状は角丸で、影をつけたスタイルにしています



#### データカード内のアイテムをスタイリングするクラス

データカード内のタイトル、サブタイトル、データ、アイコンをスタイリングするクラスです。フォントサイズや色を調整しています

タイトル .title

エネルギー

データ .data

153 kcal

Energy

アイコン .icon

サブタイトル .subtitle

```
/* =====
 * 記事
 * ===== */
```

```
.post {
  & h1 {
    margin-block: 1.25em 1em;
    font-size: min(36px, 7vw);
    font-weight: bold;
  }

  & p {
    margin-block: 1.5em;
    line-height: 1.8;
  }
}
```

### 記事のスタイルを指定するクラス

記事を構成する見出しと段落をスタイリングするクラスです。フォントのサイズと太さ、行の高さ、上下マージンを指定しています。「5.12 パンケーキ・スタック・レイアウト」で使います

#### 日々のデータを管理する

現状は日々でデータを管理しています。従来のプロジェクト、個人のスケジュール、得意の料理など、さまざまなデータが日々私たちのまわりに存在しています。このようなデータを上手に管理することは、効率的に仕事を進めたり、ストレスフリーな生活を営むために非常に重要です。そこで、今回は日々のデータをもっと管理する方法について考えてみたいと思います。

見出し <h1>

段落 <p>

```
/* =====
 * フッターバー
 * ===== */
```

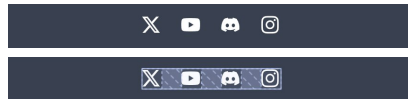
```
.bar {
  padding: 16px;
  color: white;
  background-color: var(--dark-blue-gray);

  /* 横並びメニューをカスタマイズ */
  &.horizontal {
    justify-content: center;

    & svg {
      fill: white;
    }
  }
}
```

### フッターをバーの形にするクラス

フッターを暗い青灰色のバーの形にするクラスです。フッター内の横並びメニューは横方向中央に配置し、メニュー内のSVGアイコンは白色にしています



```

/* =====
 * ダッシュボード
 * ===== */

.border-bottom {
  border-bottom:
    solid 1px rgb(0 0 0 / 20%);
}

.dark-bar {
  color: white;
  background-color: var(--dark-blue-gray);

  & h2 {
    margin: 20px 12px;
    font-size: 18px;
    color: #cbd5e1;
  }

  .menu-vertical {
    .current {
      background-color:
        rgb(255 255 255 / 10%);

      :hover {
        background-color:
          rgb(255 255 255 / 20%);
      }
    }

    .profile {
      padding: 8px 4px;

      & img {
        width: 48px;
        height: 48px;
        background-color: #f1f5f9;
        border-radius: 50%;
        object-fit: cover;
      }
    }
  }
}

```

### 要素の下に区切り線を入れるクラス

「5.15 ダッシュボードUI」でヘッダーの下に薄いグレーのボーダーを入れるのに使用します

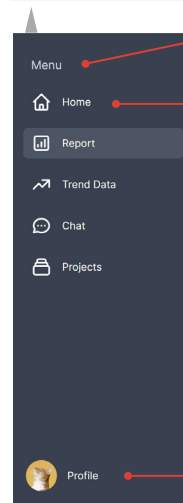


≡ Trek Planner

区切り線(下辺ボーダー)

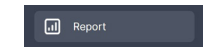
### サイドバーをスタイリングするクラス

サイドバーを暗い青灰色にします。サイドバー内  
の見出しやメニューの色を整え、アイコン画像は  
円形に切り抜きます

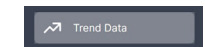


見出し <h2>

縦並びメニュー  
.menu-vertical



.currentクラスを付けた  
リンクの表示



カーソルを重ねたときの  
リンクの表示

アイコンとテキストを  
横に並べたリンク  
.profile

```
/* =====
 * チャット
 * ===== */

.chat {

  .avatar {
    display: grid;
    place-items: center;
    width: 48px;
    height: 48px;
    background: #f1f5f9;
    border: solid 2px #cbd5e1;
    border-radius: 50%;

    & svg {
      width: 24px;
      height: 24px;
    }
  }

  .form {
    & svg {
      width: 16px;
      height: 16px;
    }
  }
}
```

### チャットUIをスタイリングするクラス

「5.16 チャットUI」で使用するアバターの表示と、SVG画像のサイズを整えます。アバターはSVG画像であることを前提に、CSSグリッドで直径48pxの円の中心に24pxのSVG画像を配置するように指定しています

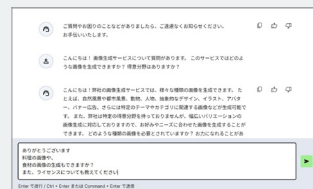
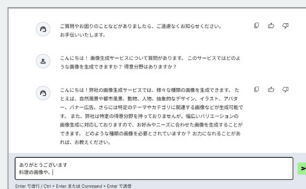
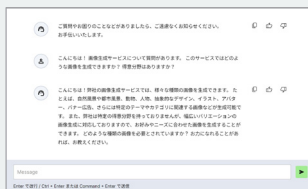


.avatar

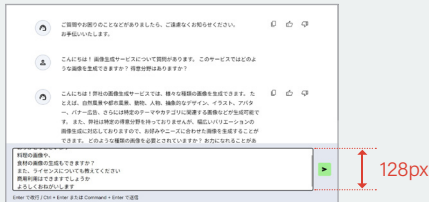


## 入力欄の高さを自動でリサイズするスクリプト (base.js)

assets フォルダに収録した JavaScript (base.js) は、「5.16 チャットUI」のフォームで入力欄 `<textarea id="message">` の高さを入力内容に合わせてリサイズします。リサイズしても崩れないように CSS グリッドでレイアウトする設定については書籍 P.306 ~を参照してください。



なお、入力欄の高さの最大値は 128px に指定し、入力内容が増えてもそれ以上大きくならないようにしています。



### assetsフォルダに収録したbase.js

```
// -----
// textarea 自動リサイズ
// -----

// ID が message の textarea 要素を処理対象に指定
const textarea = document.querySelector("textarea#message")

if (textarea !== null) {
  // スタイルの初期状態を設定
  textarea.style.overflowY = "hidden" // スクロールバーを非表示
  textarea.style.resize = "none" // 標準のリサイズ機能を無効化
  textarea.style.height = "auto" // 高さを auto に指定

  // textarea を入力内容に合わせた高さにセット
  textarea.addEventListener("input", autoResize)
  function autoResize() {
    this.style.height = "auto" // 高さを初期状態 (auto) にリセット
    this.style.height = this.scrollHeight + "px" // 入力内容の高さ (scrollHeight) を取得してセット

    this.style.maxHeight = "128px" // textarea の高さの最大値を 128px にセット
    if (this.scrollHeight > 128) {
      this.style.overflowY = "auto" // 入力内容の高さが 128px より大きいときはスクロールバーを表示
    } else {
      this.style.overflowY = "hidden" // 入力内容の高さが 128px 以下のときはスクロールバーを非表示
    }
  }

  // Ctrl + Enter または Command + Enter で送信
  textarea.addEventListener("keydown", sendByEnter)
  function sendByEnter(event) {
    if ((event.ctrlKey || event.metaKey) && event.key === "Enter") {
      textarea.form.submit() // フォームを送信する
    }
  }
}
```

## ■著者紹介

### エビスコム

<https://ebisu.com/>

Web と出版を中心にフロントエンド開発・制作・デザインを行っています。

HTML/CSS、WordPress、GatsbyJS、Next.js、Astro、Docusaurus、Figma、etc.

主な編著書：

- 『作って学ぶ WordPress ブロックテーマ』マイナビ出版刊
- 『作って学ぶ Next.js/React Web サイト構築』マイナビ出版刊
- 『作って学ぶ HTML & CSS モダンコーディング』同上
- 『HTML5 & CSS3 デザイン 現場の新標準ガイド【第2版】』同上
- 『Web サイト高速化のための 静的サイトジェネレーター活用入門』同上
- 『CSS グリッドレイアウト デザインブック』同上
- 『WordPress ノート クラシックテーマにおける theme.json の影響と対策 2023』エビスコム電子書籍出版部刊
- 『Astro v2 と TinaCMS でシンプルに作るブログサイト』同上
- 『HTML&CSS コーディング・プラクティスブック 1～8』同上

ほか多数

# 作って学ぶ HTML + CSS グリッドレイアウト サンプルの構築に使用するセット

---

2024 年 2 月 26 日 ver.1.0 発行

---

© 2024 EBISUCOM. All rights reserved.