

```
# index.css
# menu.css
# shop.css
> images
> js
< common.html
< concept.html
< index.html
< menu.html
< sample.html
< shop-detail.html
< shop.html
```

```
106
107 .item-list dt {
108   margin-top: 20px;
109
110
111 .item-list dt {
112   font-weight: bold;
113
114
115 .item-list dd {
116   font-size: 13px;
117   line-height: 20px;
118   margin-top: 10px;
119
```

index.cssの107～119行をコピー

```
# shop.css
> images
> js
< common.html
< concept.html
< index.html
< menu.html
< sample.html
< shop-detail.html
< shop.html
```

```
64
65 .item-list a:hover {
66   transform: scale(1.05);
67 }
68
69 .item-list dt {
70   margin-top: 20px;
71 }
72
73 .item-list dt {
74   font-weight: bold;
75 }
76
77 .item-list dd {
78   font-size: 13px;
79   line-height: 20px;
80   margin-top: 10px;
81
```

shop.cssの69～81行へペースト

アイテム内のテキストがスタイリングされました。これでメインコンテンツエリアは完成です。



7 サイドバーをスタイリングする

続いてサイドバーのスタイリングを進めます。

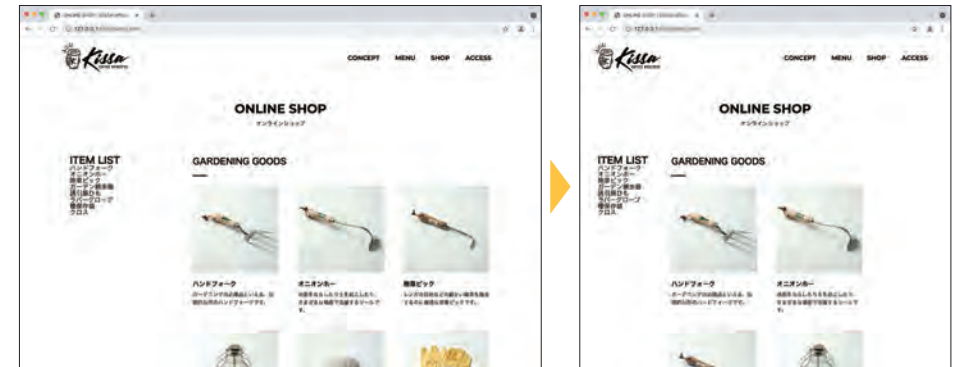
STEP 1 サイドバーの縮小率と余白を指定する

aside要素につけてあるクラス名「shop-menu」をセレクトにして、サイドバーのスタイリングを進めます。まず縮小率と余白を指定します。ブラウザの幅が狭くなっても、サイドバーは縮小せずに、大きさを保ったままにしたいので、flex-shrinkプロパティに「0」を指定します。そして、右側に60pxの余白を作り、メインコンテンツエリアとの間隔を保つようにします。

```
83 .shop-menu {
84   flex-shrink: 0;
85   margin-right: 60px;
86 }
```

```
.shop-menu {
  flex-shrink: 0;
  margin-right: 60px;
}
```

ブラウザの幅を縮めて、サイドバーの幅と右側の余白が保たれることを確認します。



STEP 2 スクロールしてもサイドバーが上部に留まるようにする

画面をスクロールしていくと、当然サイドバーも一緒にスクロールしていきますが、ブラウザの上部まで来たときにスクロールせず留まるようにします。「.shop-menu」の中にあるdiv要素「.shop-menu-inner」をセレクトにして、positionプロパティで「sticky」を指定します。こうすることで、指定の位置までスクロールしたときに、要素がスクロールに追従せず留まるようになります。位置の指定は他のpositionの値と同様に、top、left、right、bottomなどのプロパティで指定します。上部から30pxの位置に固定しましょう。

```
.shop-menu-inner {
  position: sticky;
  top: 30px;
  left: 0;
  right: 0;
}
```

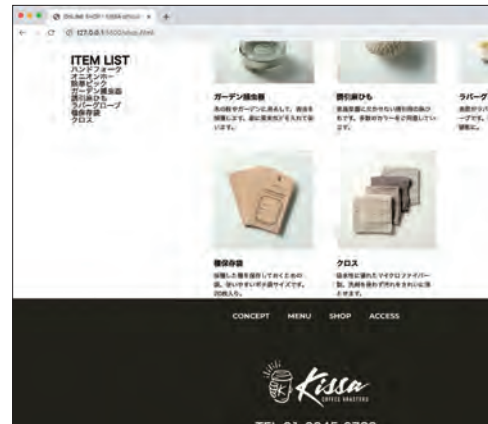


ページのスクロールにあわせてメニューも移動

ページの上部まできたら位置を留める

```
88 .shop-menu-inner {
89   position: sticky;
90   top: 30px;
91   left: 0;
92   right: 0;
93 }
```

ページをスクロールしても、サイドバーだけは追従せず、上から30pxの位置に留まるようになりました。



STEP 3 中見出しをスタイリングする

続いて、サイドバーの中身をスタイリングします。まずは中見出しです。「.shop-menu-inner内のh2要素」をセレクトにして、テキストのスタイルを入力します。

```
.shop-menu-inner h2 {
  font-size: 18px;
  font-weight: bold;
}
```

「ITEM LIST」という中見出しのスタイルが変更されました。

STEP 4 リストのスタイルを指定する

続いてul要素のスタイルを指定します。「.shop-menu-inner内のul要素」をセレクトにして、list-style-typeプロパティで行頭アイコンを「disc」に指定し、marginプロパティで余白を調整します。

```
.shop-menu-inner ul {
  list-style-type: disc;
  margin-top: 20px;
  margin-left: 20px;
}
```

```
95 .shop-menu-inner h2 {
96   font-size: 18px;
97   font-weight: bold;
98 }
```



CSS プロパティ list-style-type

リスト項目の行頭アイコンの形状を指定する。
【書式】 list-style-type: キーワード
【値】 none、disc、circle、square、decimal、upper-roman、lower-roman など

```
100 .shop-menu-inner ul {
101   list-style: disc;
102   margin-top: 20px;
103   margin-left: 20px;
104 }
```

リストの行頭にアイコンが表示され、それに合わせて余白も調整されました。

STEP 5 リストの項目のスタイルを調整する

リストの各項目のスタイルを調整します。「.shop-menu-inner内のli要素」をセレクトにして、文字サイズと上部の余白を指定します。

```
.shop-menu-inner li {
  font-size: 14px;
  margin-top: 15px;
}
```

文字サイズと余白が調整されました。これでサイドバーのスタイリングは完了です。

フッター上部の余白が調整できていませんが、商品詳細ページの完成後に行いますので、商品一覧ページはこれで一旦完成となります。



```
106 .shop-menu-inner li {
107   font-size: 14px;
108   margin-top: 15px;
109 }
```



学習ポイント

リストの行頭アイコンを指定する

リスト項目の行頭には、デフォルトでは黒い丸 (disc) が表示されますが、このアイコンはCSSで変更することが可能です。アイコンの種類は、中空円 (circle)、四角 (square) などいくつかの記号が用意されています。

また、順序付きリスト (ol要素) では、1から始まる数値 (decimal)、ローマ数字 (upper-roman、lower-roman) や漢数字 (CJK-ideographic)、日本語のひらがな (hiragana)、カタカナ (katakana)などを指定することもできます。

• disc	○ circle	■ square	1. decimal	I. upper-roman	i. lower-roman	一、 cjk-ideographic	あ、 hiragana	ア、 katakana
• disc	○ circle	■ square	2. decimal	II. upper-roman	ii. lower-roman	二、 cjk-ideographic	い、 hiragana	イ、 katakana
• disc	○ circle	■ square	3. decimal	III. upper-roman	iii. lower-roman	三、 cjk-ideographic	う、 hiragana	ウ、 katakana

8 商品詳細ページをスタイリングする

続いて商品詳細ページのスタイリングを進めていきます。

STEP 1 商品詳細ページの表示を確認する

shop-detail.htmlを開き、ブラウザでプレビューしてみましょう。

商品一覧ページと共通する部分が多いため、メインの商品の箇所以外はすでにスタイリングされた状態で表示されています。



STEP 2 画像と詳細情報を横並びにする

商品の画像と詳細情報を包んでいるdiv要素「.item-area」をセレクトにして、スタイルを記述していきます。まずは上部に20pxの余白を作り、次にdisplayプロパティで「flex」を指定して子要素を横並びにします。

```
.item-area {
  margin-top: 20px;
  display: flex;
}
```

商品画像と、詳細情報のテキストが横並びになりました。

STEP 3 商品画像の大きさを指定する

続いて、商品画像の大きさを指定します。

```
111 .item-area {
112   margin-top: 20px;
113   display: flex;
114 }
```



```
116 .item-area img {
117   width: 50%;
118   max-width: 380px;
119 }
```

「.item-area内のimg要素」をセレクトにして、幅を「50%」、最大幅を「380px」に指定します。

```
.item-area img {
  width: 50%;
  max-width: 380px;
}
```

画像の大きさが調整されました。

STEP 4 商品説明エリアの余白を調整する

画像と商品説明エリアがくっついているので、あいだに余白を作ります。div要素「.about-item」をセレクトにして、左側に30pxの余白を指定します。

```
.about-item {
  margin-left: 30px;
}
```

画像と商品説明エリアのあいだに余白ができました。

STEP 5 商品説明文と商品価格のスタイルを指定する

商品の説明文と商品価格のスタイルを指定します。説明文はp要素「.item-text」に、商品価格はp要素の「.item-price」をセレクトにして、それぞれ指定を入力します。

```
.about-item .item-text {
  font-size: 14px;
  line-height: 26px;
}
```

```
.about-item .item-price {
  font-weight: bold;
  margin-top: 20px;
}
```



```
121 .about-item {
122   margin-left: 30px;
123 }
```



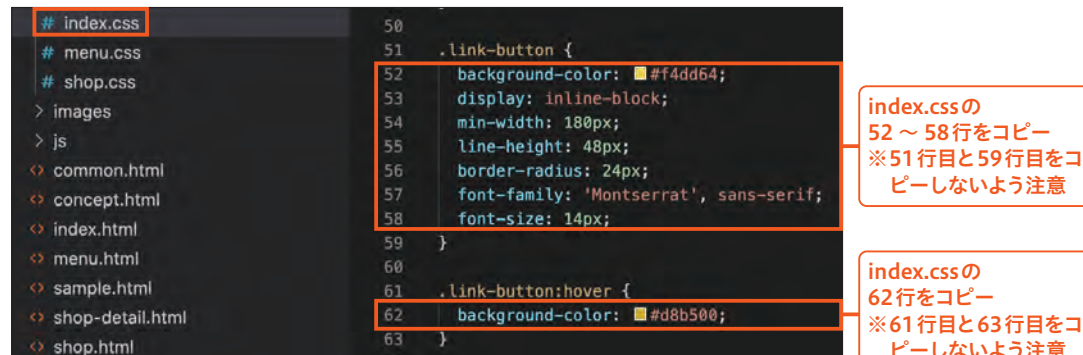
```
125 .about-item .item-text {
126   font-size: 14px;
127   line-height: 26px;
128 }
129
130 .about-item .item-price {
131   font-weight: bold;
132   margin-top: 20px;
133 }
```

テキストのスタイルが調整されました。



STEP 6 ボタンのスタイルを トップページから流用する

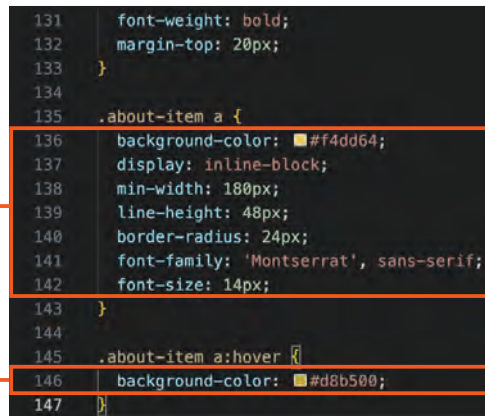
続いて「BUY NOW」ボタンを作成します。「.about-item内のa要素」をセレクトにして、スタイルを記述します。スタイルはindex.cssの52～58行目をコピーして流用します。そして、疑似クラス「:hover」でマウスを置いたときのスタイルを指定します。疑似クラスのスタイルも、同じくindex.cssの62行目のスタイルを複製します。ここでもコピーするのはスタイルだけで、セレクトはコピーしないように注意してください。



```
.about-item a {
  background-color: #f4dd64;
  display: inline-block;
  min-width: 180px;
  line-height: 48px;
  border-radius: 24px;
  font-family: 'Montserrat', sans-serif;
  font-size: 14px;
}

.about-item a:hover {
  background-color: #d8b500;
}
```

コピーしたindex.cssの52～58行を、shop.cssの136～142行にペースト



コピーしたindex.cssの62行を、shop.cssの146行にペースト

a要素の表示が変更されました。

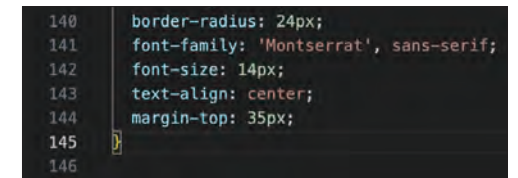


STEP 7 ボタンのスタイルを調整する

流用したスタイルのままでは表示が崩れているのでスタイルを調整します。テキストの揃えを中央揃えにし、上部に35pxの余白を作ります。

```
.about-item a {
  ... (略) ...
  font-size: 14px;
  text-align: center;
  margin-top: 35px;
}
```

ボタンのスタイルが完成しました。



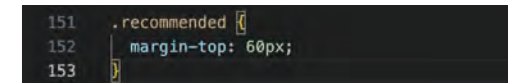
9 コンテンツ間の余白を調整する

最後に余白を調整します。

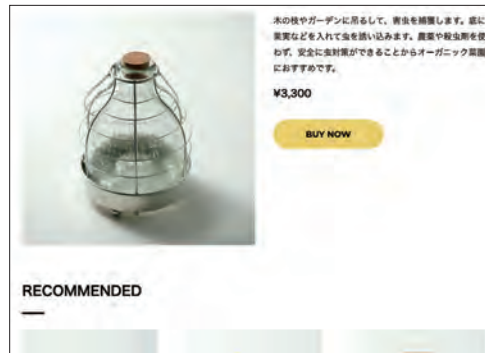
STEP 1 RECOMMENDEDエリアの余白を調整する

まずはRECOMMENDEDエリアの上部の余白を調整します。クラス名「recommended」をセレクトにして上部に60pxの余白を作ります。

```
.recommended {
  margin-top: 60px;
}
```



RECOMMENDEDエリアの上部に余白ができました。

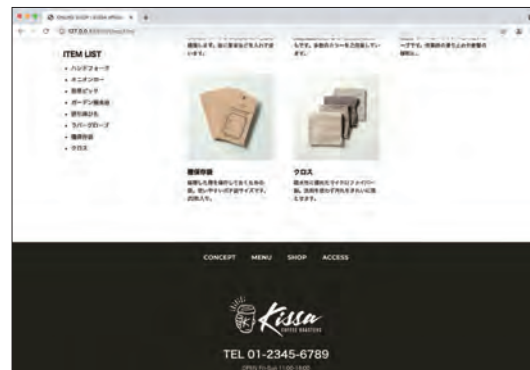
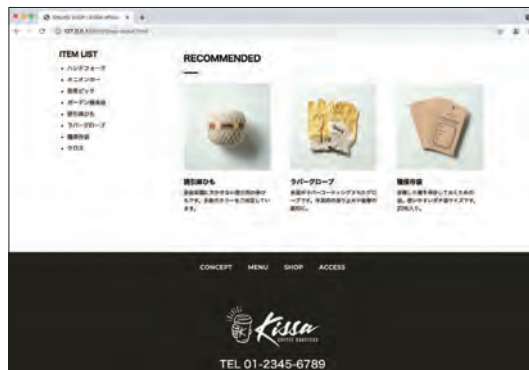


STEP 2 フッターの余白を調整する

最後にフッターの上部に100pxの余白を指定します。

```
.footer {
  margin-top: 100px;
}
```

```
155 .footer {
156   margin-top: 100px;
157 }
```



フッター上部の余白が調整されました。shop.htmlも確認しましょう。これでPC版のスタイリングは完成です。

10-5 2カラムレイアウトのモバイル用CSSを書いてみよう

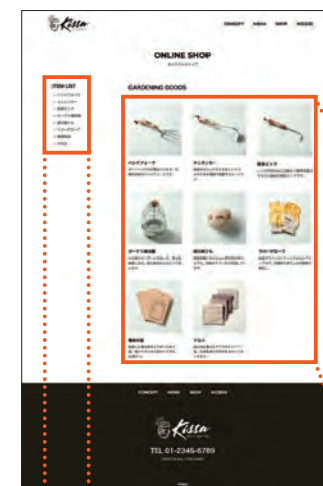
続いてモバイル用のCSSを記述します。デベロッパーツールで表示を確認し、モバイル表示用に最適化していきます。

1 モバイル用レイアウトのデザインカンプを確認する

モバイルでは表示幅が狭く2カラムレイアウトは使いにくいので、サイドバーを下部に移動し縦並びのレイアウトに変更します。サイドバーにはグレーの背景を敷いて、コンテンツエリアとの境界を明確にします。そして、3列で並んでいた商品もモバイルで表示するには窮屈なため、2列表示にします。こ

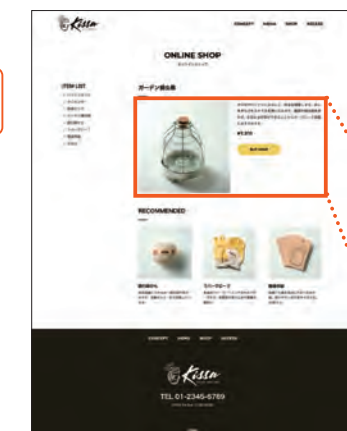
れらは、商品一覧、商品詳細ページ共通の要素のため、一度の指定で両方のページに適用されます。また、商品詳細ページでは、画像と商品説明の並び順を変えて、ボタンもモバイル用に押しやすいサイズに変更します。

商品一覧ページ



格子状のレイアウトを2列表示に変更

商品詳細ページ



商品画像と商品説明を縦並びに変更

ボタンのサイズをエリア幅いっぱいに変更

サイドバーを下部に移動し、薄いグレーの背景色を敷く

2 デベロッパーツールで表示を確認する

デベロッパーツールを起動し、Chromeをレスポンシブモードに切り替え、まずは商品一覧ページの表示を確認します。

サイドバーが左にはみ出してレイアウトが崩れています。



3 エリア全体のレイアウトを調整する

まずは全体のレイアウトをモバイル用に作り変えていきます。

STEP 1 メディアクエリを記述する

これまでのモバイル対応と同様にメディアクエリを記述しましょう。`.footer`への指定の下に入力します。

```
@media (max-width: 800px) {}
```

```
155 .footer {
156   margin-top: 100px;
157 }
158
159 @media (max-width: 800px) {}
```

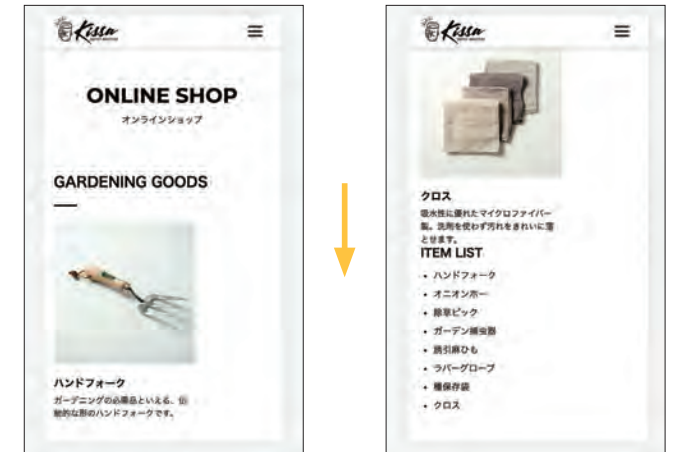
STEP 2 子要素の横並びを解除する

まずは、「`.shop-contents`」に`display`プロパティで「`block`」を指定し、子要素の横並びを解除します。

```
@media (max-width: 800px) {
  .shop-contents {
    display: block;
  }
}
```

```
159 @media (max-width: 800px) {}
160 .shop-contents {
161   display: block;
162 }
163
```

サイドバーが見えなくなりました。スクロールして、サイドバーがメインコンテンツエリアの下部に移動していることを確認します。



STEP 3 全体の幅と余白を調整する

エリア全体に`max-width: 90%`の指定がされていますが、これを100%にします。そして、上部の余白を60pxに変更します。

```
@media (max-width: 800px) {
  .shop-contents {
    display: block;
    max-width: 100%;
    margin-top: 60px;
  }
}
```

左側の余白がなくなりました。

note

左右の余白をなくして幅いっぱいにしたのは、サイドバーの装飾で背景色を幅いっぱいになくす際に必要となるためです。一方、メインコンテンツエリアでは左右の余白が必要になるので、この次の工程で作成します。

```
159 @media (max-width: 800px) {}
160 .shop-contents {
161   display: block;
162   max-width: 100%;
163   margin-top: 60px;
164 }
165
```



4 メインコンテンツエリアの表示を調整する

続いて、商品リストが並ぶメインコンテンツエリアの表示を調整します。

note

同じスタイルを指定している商品詳細ページのRECOMMENDEDエリアも変更されます。

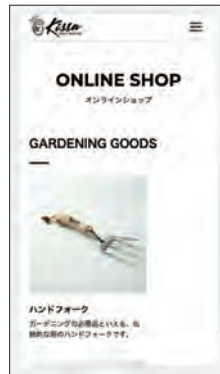
STEP 1 メインコンテンツエリアのサイズと配置を指定する

「.shop-item」をセレクトにして、エリアの幅と配置を調整します。最大幅を540pxに指定し中央に配置、左右の内側に20pxずつの余白を指定します。

```
.shop-item {
  max-width: 540px;
  margin-left: auto;
  margin-right: auto;
  padding-left: 20px;
  padding-right: 20px;
}
```

左側に余白ができました。

```
166 .shop-item {
167   max-width: 540px;
168   margin-left: auto;
169   margin-right: auto;
170   padding-left: 20px;
171   padding-right: 20px;
172 }
173
```



STEP 2 グリッドの値を変更する

次に、セレクト「.item-list」に指定されているグリッドの値を変更します。grid-template-columnsプロパティで「repeat(2, 1fr);」を指定し、2分の1の幅で列が作られるようにします。そして、column-gapプロパティで列の間隔を「35px」に指定します。

```
.item-list {
  grid-template-columns: repeat(2, 1fr);
  column-gap: 35px;
}
```

商品が2列で表示されました。これでメインコンテンツエリアの調整は完了です。

```
174 .item-list {
175   grid-template-columns: repeat(2, 1fr);
176   column-gap: 35px;
177 }
178
```



5 サイドバーの表示を調整する

次にサイドバーの表示を調整します。

note

同じスタイルを指定している商品詳細ページのサイドバーも変更されます。

STEP 1 サイドバー全体のスタイルを調整する

まずはサイドバー全体の表示を調整します。メインコンテンツエリアとの差別化するために、背景を薄いグレーにします。aside要素の「.shop-menu」をセレクトにして、background-colorプロパティで「f8f8f8」を指定し、さらにmarginとpaddingで余白を調整します。

```
.shop-menu {
  background-color: #f8f8f8;
  padding-top: 50px;
  padding-bottom: 50px;
  margin-top: 60px;
  margin-right: 0;
}
```

サイドバーの背景に薄いグレーが敷かれました。

STEP 2 サイドバー内側のボックスのサイズを調整する

サイドバー内側のボックス「.shop-menu-inner」のサイズを調整します。「.shop-item」に指定したスタイルと同じなので、167～171行目をコピーしてペーストします。

```
.shop-menu-inner {
  max-width: 540px;
  margin-left: auto;
  margin-right: auto;
  padding-left: 20px;
  padding-right: 20px;
}
```

これでサイドバーの調整も完了です。フッターとのあいだに余白がありますが、これは詳細ページのスタイル調整と併せて行うので、一覧ページは一旦これで完成となります。

```
179 .shop-menu {
180   background-color: #f8f8f8;
181   padding-top: 50px;
182   padding-bottom: 50px;
183   margin-top: 60px;
184   margin-right: 0;
185 }
186
```



```
187 .shop-menu-inner {
188   max-width: 540px;
189   margin-left: auto;
190   margin-right: auto;
191   padding-left: 20px;
192   padding-right: 20px;
193 }
194
```



左ページのSTEP.1のスタイルのみコピーしてペースト

6 商品詳細ページのレイアウトを調整する

次に、商品詳細ページのレイアウトを調整していきます。shop-detail.htmlを開き、Chromeのデベロッパーツールで確認してみましょう。下までスクロールしていくと、商品一覧ページで指定したスタイルにより、RECOMMENDEDエリアとサイドバーのレイアウトがすでに変更されていることがわかります。

しかし商品詳細エリアでは商品画像が半分の幅に押し込められ、縦に伸びて表示されています。これらを調整していきます。



STEP 1 子要素の横並びを解除する

「.item-area」のdisplayプロパティに「block」を指定し、子要素の横並びを解除します。

```
.item-area {
  display: block;
}
```

要素が縦並びになりました。

```
195 .item-area {
196   display: block;
197 }
198
```



STEP 2 画像のサイズを調整する

商品画像が幅いっぱい広がるようにしましょう。「.item-area img」にwidthとmax-widthプロパティでどちらも「100%」を指定します。

```
199 .item-area img {
200   width: 100%;
201   max-width: 100%;
202 }
203
```

```
.item-area img {
  width: 100%;
  max-width: 100%;
}
```

商品画像が幅いっぱいに表示されました。



STEP 3 商品説明エリアの配置を調整する

商品説明エリア「.about-item」の余白を調整し、配置を整えます。

```
.about-item {
  margin-top: 20px;
  margin-left: 0;
}
```

商品説明エリアの配置が調整されました。

```
204 .about-item {
205   margin-top: 20px;
206   margin-left: 0;
207 }
208
```



STEP 4 購入ボタンの幅を調整する

購入ボタンの幅を変更し、エリア全体に広がるようにします。widthプロパティに「100%」を指定しましょう。

```
.about-item a {
  width: 100%;
}
```

```
209 .about-item a {
210   width: 100%;
211 }
212
```

note

スマートフォンなど指先で操作する端末では、ボタン類を扱いやすいサイズにすることが大切です。ここではエリア全体の幅に広げることで押しやすいボタンを作成します。

ボタンの幅が広がりました。



7 フッターの余白を調整する

最後にサイドバーとフッターのあいだの余白を調整します。

「.footer」をセレクトに、marginプロパティで上部の余白を「0」にします。

```
.footer {  
  margin-top: 0;  
}
```

サイドバーとフッターのあいだの余白がなくなりました。shop.htmlページも確認しましょう。これでモバイル版のレイアウトも完了し、オンラインショップページは完成となります。

