

記事 LP 修正のご依頼

対象記事 URL :

- <https://sb-draft-preview.squadbeyond.com/articles/jYCNrvdkcuBqCSJELIOw/draft?token=fb9319129a62d44c30c6b9a687be19e6>
-

修正したい箇所 :

- 現状

クーポンの受け取り方法・使い方は**カンタン**です。

①以下をタップして**広告コードをコピー**して↓

MOUSE20C

コピー

②以下のボタンからウィキッド・マウスのAmazonページに移動↓

 **Amazonで購入する**

＼この広告を閉じたら二度と開けないかもしれないので注意／

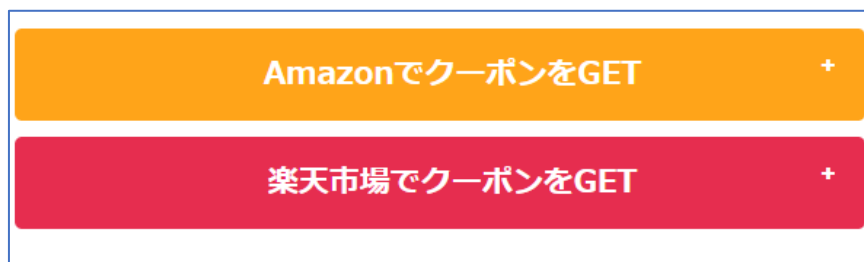
③**Amazonの購入画面で入力すると20%オフになります！**

ギフト券、種類別商品券、またはプロモーションコード

コードを入力

適用

- 修正後 :
- CTA を Amazon・楽天市場双方に対応したものに拡張したい



◇

◇

イメージは、上部対応するボタンをクリック（タップ）するとアコーディオンが広がり、現状イメージの説明（Amazon・楽天それぞれに対応したクーポンの受取～利用法）が表示される形（以下楽天の例）

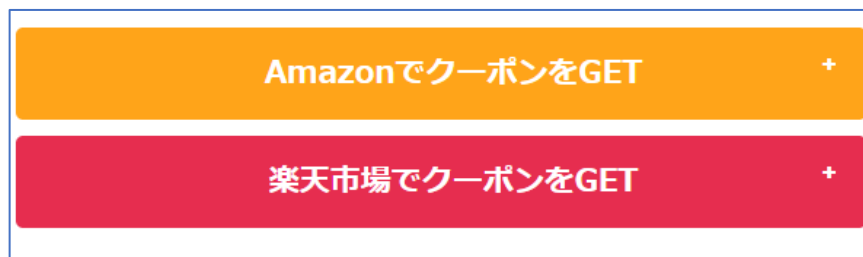


■ 現状の問題

- 以下①～③のモジュールを組み合わせて上記実現したく考えております（①ボタン展開後表示される部分に②・③を埋め込みたい形）

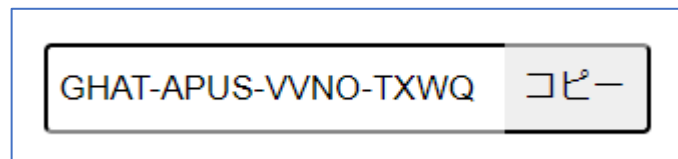
◇

①アコーディオンボタンのモジュール

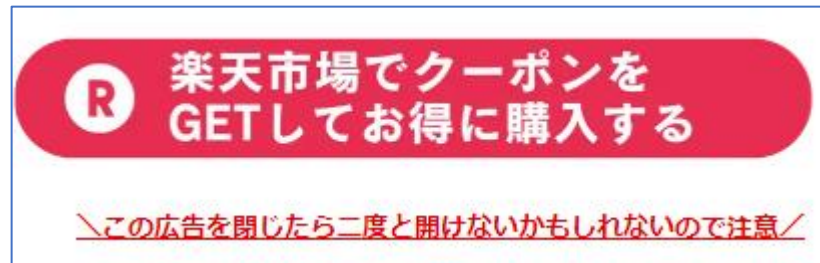


◇

②クーポンコードコピーのモジュール（“コピー”タップでクリップボードにコードが自動貼付）



◇ ③遷移ボタンのモジュール



- 一方、現状以下の様な不具合があり困っております
 - ◇ ②・③のレイアウトが崩れる（①のアコーディオン枠からはみ出してしまう）
 - ◇ ②のコピーボタンタップ時、①アコーディオンが閉じてしまう

ご依頼：

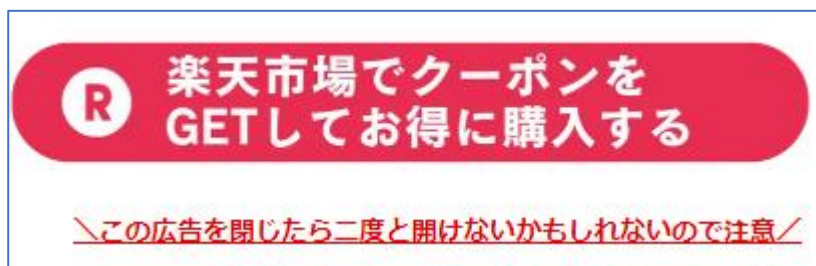
- 各モジュールのコードを以下にそれぞれ記載しますので、上記①～③を統合した HTML・CSS 案をご教示頂けますでしょうか？

モジュール①（アコーディオンモジュール）

- 以下コードは要素が 4 ボタン・カラーが 1 色ですが、上記イメージの通り 2 ボタン 2 カラーにしたいです（楽天・Amazon でボタンカラーを別にしたい為）



- - 本モジュール自体を 2 個挿入する、1 個のモジュール内部でボタンカラーを変える、いずれも検討可能です
 - 添付の画像の様に、“テキストが入ります”という文言が入っているボタン部分左側にアイコン画像を挿入できるとベターです



HTML

```
1 <script src="https://code.jquery.com/jquery-3.6.1.min.js" integrity="sha256-
2 o88AwQnZB+VDvE9tvIXrMQaPIFFSUTR+nldQm1LuPXQ=" crossorigin="anonymous"></script>
3 <script>
4     var accBtn = accBtn || {};
5     (function(_) {
6         const target = '.js-accbtn';
7
8         __.acc = function(selector) {
9             const eList = $(selector);
10            eList.each(function() {
11                const widget = $(this);
12                const btns = widget.find('.js-btn');
13                widget.find('.js-slide').hide();
14                btns.each(function() {
15                    const btn = $(this);
16                    btn.on('click', (e) => {
17                        const t = $(e.currentTarget);
18                        if (!t.hasClass('is-checked')) {
19                            t.addClass('is-checked');
20                            t.find('.js-slide').slideDown();
21                        } else {
22                            t.find('.js-slide').slideUp();
23                            t.removeClass('is-checked');
24                        }
25                        const otherBtns = widget.find('.js-btn').not(t);
26                        otherBtns.each(function() {
27                            if ($(this).hasClass('is-checked')) {
28                                $(this).find('.js-slide').slideUp();
29                                $(this).removeClass('is-checked');
30                            }
31                        });
32                    });
33                });
34            });
35        };
36
37        __.init = function() {
38            __.acc(target);
39        };
40    })(accBtn);
41 </script>
42 <script>
43     document.addEventListener('DOMContentLoaded', accBtn.init, false);
44 </script>
```

```
45 <!-- HTML ここから -->
46 <div class="c-accbtn js-accbtn">
47   <!-- アコーディオン要素↓ここから↓ -->
48   <div class="c-item js-btn">
49     <div class="p-lead">テキストが入ります</div>
50     <!-- スライド要素ここから -->
51     <div class="p-cont js-slide">
52       スライドで表示される要素<br>
53       
56     </div>
57     <!-- スライド要素ここまで -->
58   </div>
59   <!-- ↑ここまで↑ -->
60   <div class="c-item js-btn">
61     <div class="p-lead">テキストが入ります</div>
62     <div class="p-cont js-slide">
63       スライドで表示される要素<br>
64       
67     </div>
68   </div>
69   <div class="c-item js-btn">
70     <div class="p-lead">テキストが入ります</div>
71     <div class="p-cont js-slide">
72       スライドで表示される要素<br>
73       
76     </div>
77   </div>
78   <div class="c-item js-btn">
79     <div class="p-lead">テキストが入ります</div>
80     <div class="p-cont js-slide">
81       スライドで表示される要素<br>
82       
85     </div>
86   </div>
87 </div>
88
89
90
```

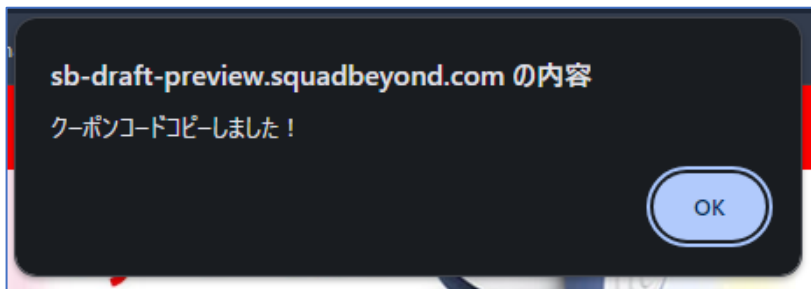
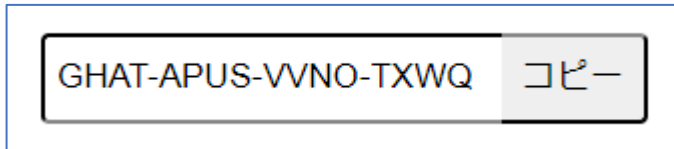
91

CSS

```
1  .c-accbtn {
2      max-width: 750px;
3      width: 100%;
4      margin: 0 auto 20px;
5      box-sizing: border-box;
6  }
7
8  .c-item {
9      border: 5px solid #ffa70f;
10     background: #ffa70f;
11     border-radius: 10px;
12     box-shadow: 2px 2px 2px rgb(0 0 0 / 30%);
13     cursor: pointer;
14     box-sizing: border-box;
15 }
16
17 .c-item.is-checked {
18     background: #ff5b0f;
19     border-color: #ff5b0f;
20     box-shadow: none;
21 }
22
23 .c-item+.c-item {
24     margin-top: 10px;
25 }
26
27 .c-item img {
28     max-width: 100%;
29     height: auto;
30     vertical-align: bottom;
31     margin: 0 auto;
32 }
33
34 .p-lead {
35     font-size: 18px;
36     text-align: center;
37     color: #fff;
38     padding: 10px;
39     box-sizing: border-box;
40 }
41
42 .p-cont {
43     background: #fff;
44 }
```

モジュール②（コードコピーモジュール）

- モジュール②・③は、モジュール①でボタンをタップした後に表示される要素（＝“スライドで表示される要素”）となります
- コードがタップでコピーできること、コピーした場合にアコーディオンが閉じないことを担保したいです



HTML

```

1 <!--HTML ここから-->
2 <div class="clipboard">
3   <!--コピーするテキストは value=" " 内を変更ください--><input class="coupon" type="text" value="GHAT-
4 APUS-VVNO-TXWQ"><!--ボタンの文言変更は下記--><button onclick="copyToClipboard('.coupon')">コピー
5 </button>
6 </div>
7 <!--HTML ここまで-->

```

CSS

```

1 .clipboard {
2   /*外側の余白*/
3   margin: 6px 0;
4   display: flex;
5   justify-content: center;
6   height: max-content;
7 }
8
9 input.coupon {
10  /*テキストエリアの線の色*/
11  border-color: #333;
12  /*テキストエリア内側の余白。上下 | 左右*/
13  padding: 5px 6px;
14  /*テキストエリアの角丸。上 | 右 | 下 | 左*/
15  border-radius: 4px 0 0 4px;
16  /*背景色*/
17  background-color: #fff;

```

```

18     border-right-width: 0;
19 }
20
21 button {
22     /*ボタンの線の色*/
23     border-color: #333;
24     /*背景色*/
25     background-color: #efefef;
26     /*ボタン内側の余白。上下 | 左右*/
27     padding: 5px 10px;
28     /*ボタンの角丸。上 | 右 | 下 | 左*/
29     border-radius: 0 4px 4px 0;
30     border-left-width: 0;
31 }

```

モジュール③（遷移ボタンモジュール）

- モジュール②・③は、モジュール①でボタンをタップした後に表示される要素（＝“スライドで表示される要素”）となります
- ボタン画像はこちらで用意しますので、コードの統合のみご対応いただけると幸いです

HTML

```

1 <div class="puyoBtn__area">
2     <!-- 下記 oooooo 部分にリンク先 URL を置いてください。もしくは左の表示されているボタンテキストを選択した後に上部のリンクボタ
3     ンから追加可能--> <a href="https://sb.nekorakudo.com/redirect_pages/b3d1f0cb-7b28-4dbc-8553-
4     c138f669a85f?sbrp=true&sbrpuid=b3d1f0cb-7b28-4dbc-8553-c138f669a85f&sb_tracking=true"
5     class="btn" draggable="false"><!-- 画像の入れ替えは下記 URL 変更、もしくは左の表示されている画像を選択した後に上部
6     の画像ボタンから入れ替え可能-->
7         <picture>
8             <source type="image/webp"
9             srcset="https://file.mysquadbeyond.com/uploads/article_photo/photo/3580633/abbe8984-162e-4c34-
10             8daf-8070669ae816.webp">
11             <source type="image/avif"
12             srcset="https://file.mysquadbeyond.com/uploads/article_photo/photo/3580633/abbe8984-162e-4c34-
13             8daf-8070669ae816.avif">
16         </picture>
17     </a>
18 </div>
19

```

CSS

```

1
2 /* 全体の設定 */
3 .puyoBtn__area {
4     /* 横幅（ボタンの大きさに影響します） */
5     width: 400px;

```



```
6      /* 最大幅 */
7      max-width: 95%;
8      /* 高さ */
9      height: auto;
10     /*外側の余白 | 上下余白 | 左右余白 (auto=センタリング) | */
11     margin: 10px auto;
12 }
13
14
15
16 /* ボタンリンクの設定 */
17 .puyoBtn__area .btn {
18     display: block;
19     /* 横幅 */
20     width: 100%;
21     /* 高さ */
22     height: auto;
23 }
24
25
26
27 /* ボタンの画像 */
28 .puyoBtn__area .btn img {
29     /* 横幅 */
30     width: 100%;
31     /* 高さ */
32     height: auto;
33     /*アニメーションの設定 | アニメーション名 | 1回のサイクルに要する時間 | 進め方 | 実行回数 | */
34     animation: puyoBtn 1.6s ease-in infinite;
35 }
36
37 /* アニメーションの詳細設定 */
38 @keyframes puyoBtn {
39
40     /* タイミング */
41     0% {
42         /* ボタンの拡大、縮小
43         ・値が 1 つ = X&Y 軸
44         ・値が 2 つ = X 軸, Y 軸 */
45         transform: scale(1.09);
46     }
47
48     10% {
49         transform: scale(1.09);
50     }
51 }
```

```
52 30% {
53   transform: scale(1.01);
54 }
55
56 40% {
57   transform: scale(1.01);
58 }
59
60 50% {
61   transform: scale(1, 1.04);
62 }
63
64 60% {
65   transform: scale(1.09);
66 }
67
68 70% {
69   transform: scale(1.02);
70 }
71
72 100% {
73   transform: scale(1.09);
74 }
75 }
```