

sample

STUDY_html_003_E.html

```
<!doctype html>
<html lang="ja">

<style>
html{box-sizing:border-box;
-- カスタムカラー :#000000;
-- デレイタイム :.7s;}
html*,
html*::before,
html*::after{box-sizing:inherit;}
body{margin:0;
display:flex;
height:240px;
overflow:hidden;
align-items:center;
justify-content:center;
background-color:#afeeee;
-webkit-tap-highlight-color:transparent;
transition:background-color var(-- デレイタイム );}
. 種類分類 {margin:0;
display:flex;
width:32em;
font-size:24px;
padding:0 3em;
position:relative;
align-items:center;
justify-content:center;
background-color:var(-- カスタムカラー );}
. お品書き {all:unset;
flex-grow:1;
z-index:100;
display:flex;
cursor:pointer;
position:relative;
border-radius:50%;
align-items:center;
will-change:transform;
justify-content:center;
padding:0.55em 0 0.85em;
transition:transform var(-- カスタムタイムアウト ),var(-- デレイタイム );}
```

```

. お品書き ::before{content:"";
z-index:-1;
width:4.2em;
height:4.2em;
border-radius:50%;
position:absolute;
transform:scale(0);
transition:background-color var(-- デイレイタイム ),transform var(-- デイレイタイム );}
. お品書き .active{transform:translate3d(0,-.8em,0);}
. お品書き .active::before{transform:scale(1);
background-color:var(-- カスタムオブジェ );}
. 目印 {width:2.6em;
height:2.6em;
stroke:white;
fill:transparent;
stroke-width:1pt;
stroke-miterlimit:10;
stroke-linecap:round;
stroke-linejoin:round;
stroke-dasharray:400;}
. お品書き .active . 目印 {animation:strok 1.5s reverse;}
@keyframes strok{100%{stroke-dashoffset:400;}}
. 区分け {left:0;
bottom:99%;
width:10.9em;
height:2.4em;
position:absolute;
clip-path:url(# 上下左右 );
will-change:transform;
background-color:var(-- カスタムカラー );
transition:transform var(-- カスタムタイムアウト ,var(-- デイレイタイム ));}
. 白い線 {width:0;
height:0;}
@media screen and (max-width:50em)
{. 種類分類 {font-size:.8em;}}
</style>

```

```

<body translate="no">
<menu class=" 種類分類 ">
<button class=" お品書き active"style="-- カスタムオブジェ :#007f6f;">
<svg class=" 目印 "viewBox="0 0 24 24">
<path d="M3.8,6.6h16.4"/>
<path d="M20.2,12.1H3.8"/>
<path d="M3.8,17.5h16.4"/>
</svg>
</button>
<button class=" お品書き "style="-- カスタムオブジェ :#20b2aa;">
<svg class=" 目印 "viewBox="0 0 24 24">
<path d="M6.7,4.8h10.7c0.3,0,0.6,0.2,0.7,0.5l2.8,7.3c0,0.1,0,0.2,0,0.3v5.6c0,0.4-0.4,0.8-0.8,0.8H3.8
C3.4,19.3,3,19.3,18.5v-5.6c0-0.1,0-0.2,0.1-0.3L6.5,3C6.1,5,6.4,4.8,6.7,4.8z"/>
<path d="M3.4,12.9H8l1.6,2.8h4.9l1.5-2.8h4.6"/>
</svg>
</button>
<button class=" お品書き "style="-- カスタムオブジェ :#6495ed;">
<svg class=" 目印 "viewBox="0 0 24 24">
<path d="M3.4,11.9l8.8,4.4l8.4-4.4"/>
<path d="M3.4,16.2l8.8,4.5l8.4-4.5"/>
<path d="M3.7,7.8l8.6-4.5l8,4.5l-8,4.3L3.7,7.8z"/>
</svg>
</button>

```

```

<button class=" お品書き "style="-- カスタムオブジェ :#8a2be2;">
<svg class=" 目印 "viewBox="0 0 24 24">
<path d="M5.1,3.9h13.9c0.6,0,1.2,0.5,1.2,1.2v13.9c0,0.6-0.5,1.2-1.2,1.2H5.1c-0.6,0-1.2-0.5-1.2-1.2V5.1
C3.9,4.4,4.4,3.9,5.1,3.9z"/>
<path d="M4.2,9.3h15.6"/>
<path d="M9.1,9.5v10.3"/>
</button>
<button class=" お品書き "style="-- カスタムオブジェ :#ff1493;">
<svg class=" 目印 "viewBox="0 0 24 24">
<path d="M5.1,3.9h13.9c0.6,0,1.2,0.5,1.2,1.2v13.9c0,0.6-0.5,1.2-1.2,1.2H5.1c-0.6,0-1.2-0.5-1.2-1.2V5.1
C3.9,4.4,4.4,3.9,5.1,3.9z"/>
<path d="M5.5,20l9.9-9.9l4.7,4.7"/>
<path d="M10.4,8.8c0,0.9-0.7,1.6-1.6,1.6c-0.9,0-1.6-0.7-1.6-1.6C7.3,8,8,7.3,8.9,7.3C9.7,7.3,10.4,8,10.4,8.8z"/>
</svg>
</button>
<div class=" 区分け "></div>
</menu>
<div class=" 白い線 ">
<svg viewBox="0 0 202.9 45.5">
<clipPath id=" 上下左右 "clipPathUnits="objectBoundingBox"transform="scale(0.0049285362247413 0.021978021978022)">
<path d="M6.7,45.5c5.7,0.1,14.1-0.4,23.3-4c5.7-2.3,9.9-5,18.1-10.5c10.7-7.1,11.8-9.2,20.6-14.3c5-2.9,9.2-5.2,15.2-7
c7.1-2.1,13.3-2.3,17.6-2.1c4.2-0.2,10.5,0.1,17.6,2.1c6.1,1.8,10.2,4.1,15.2,7c8.8,5.9,9.7,1,20.6,14.3c8.3,5.5,12.4,8.2,18.1,10.5
c9.2,3.6,17.6,4.2,23.3,4H6.7z"/>
</clipPath>
</svg>
</div>

<script>
"use strict";
const 全体=document.body;
const 背景の色=["#98fb98","#008b8b","#87cefa","#ee82ee","#ffc0cb"];
const 種類分類 = 全体 .querySelector(". 種類分類 ");
const あいてむ = 種類分類 .querySelectorAll(". お品書き ");
const 仕切り線 = 種類分類 .querySelector(". 区分け ");
let 動き方 = 種類分類 .querySelector(".active");
function <りっく (item,index)
{ 種類分類 .style.removeProperty("-- カスタムタイムアウト ");
if( 動き方 ==item)return;
if( 動き方 ){ 動き方 .classList.remove("active");}
item.classList.add("active");
全体 .style.backgroundColor= 背景の色 [index];
動き方 =item;
境界線 ( 動き方 , 仕切り線 );}
function 境界線 (element, 仕切り線 )
{const 分かれ目 =element.getBoundingClientRect();
const 左側=Math.floor( 分かれ目 .left- 種類分類 .offsetLeft-( 仕切り線 .offsetWidth- 分かれ目 .width)/2)+"px";
仕切り線 .style.transform='translate3d($ { 左側 },0,0)`;}
境界線 ( 動き方 , 仕切り線 );
あいてむ .forEach((item,index)=>
{item.addEventListener("click",()=><りっく (item,index));});
window.addEventListener("resize",()=>{ 境界線 ( 動き方 , 仕切り線 );
種類分類 .style.setProperty("-- カスタムタイムアウト ","none");});
</script>

</html>

```