# **ステップ4：メモリーカード（神経衰弱）**

****ねらい：**** ****配列・シャッフル・状態管理****、条件分岐と遅延処理。

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

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>ステップ4：メモリーカード</title>

<style>

body{font-family: system-ui, sans-serif; margin:0; display:grid; place-items:center; min-height:100vh; background:#eef2ff}

.wrap{background:#fff; padding:24px; border-radius:16px; box-shadow:0 6px 18px rgba(0,0,0,.08); text-align:center}

.grid{display:grid; grid-template-columns:repeat(4,90px); gap:10px; margin:16px auto}

.card{width:90px; height:120px; border-radius:10px; background:#c7d2fe; display:grid; place-items:center; font-size:42px; cursor:pointer; user-select:none; transition:transform .15s}

.open{background:#fef9c3; transform:rotateY(180deg)}

.matched{background:#86efac}

</style></head><body>

<div class="wrap">

<h1>メモリーカード</h1>

<div>手数：<b id="moves">0</b> ／ そろえた枚数：<b id="found">0</b>/8</div>

<div class="grid" id="grid"></div>

</div>

<script>

const emojis = [',',',',',',','];

const deck = [...emojis, ...emojis];

// シャッフル（フィッシャー–イェーツ）

for (let i=deck.length-1; i>0; i--) {

const j = Math.floor(Math.random()\*(i+1));

[deck[i], deck[j]] = [deck[j], deck[i]];

}

const grid = document.getElementById('grid');

const movesEl = document.getElementById('moves');

const foundEl = document.getElementById('found');

let first=null, second=null, lock=false, moves=0, found=0;

deck.forEach((ch, idx)=>{

const d = document.createElement('div');

d.className = 'card';

d.dataset.symbol = ch;

d.dataset.idx = idx;

d.textContent = '❓';

grid.appendChild(d);

});

function openCard(card){

if (lock) return;

if (card.classList.contains('open') || card.classList.contains('matched')) return;

card.classList.add('open');

card.textContent = card.dataset.symbol;

if (!first) { first = card; return; }

second = card; lock = true; moves++; movesEl.textContent = moves;

if (first.dataset.symbol === second.dataset.symbol){

first.classList.add('matched');

second.classList.add('matched');

found+=1; foundEl.textContent = found;

[first, second] = [null, null]; lock=false;

if(found===8) setTimeout(()=>alert('クリア！手数：'+moves), 10);

} else {

setTimeout(()=>{

first.classList.remove('open'); first.textContent='❓';

second.classList.remove('open'); second.textContent='❓';

[first, second] = [null, null]; lock=false;

}, 800);

}

}

grid.addEventListener('click', (e)=>{

const card = e.target.closest('.card');

if (card) openCard(card);

});

</script></body></html>

### **チャレンジ問題**

1. ****タイマー****をつけて、早解きに挑戦。
2. 5×4（10ペア）に拡張。絵文字の種類を増やす。

****難易度ダイヤル****：カードサイズを小さく／大きく、遅延時間を短く。