

行を折り返す

```
1 <!doctype html>
2 <html>
3 <body>
4 <script>
5
6   /* ゲーム画面の構築 */
7   const canvas = document.createElement('canvas');
8   const ctx = canvas.getContext('2d');
9
10  canvas.width = 400;
11  canvas.height = 400;
12
13  canvas.setAttribute('style', 'display:block;margin:auto;background-color: #c
14
15  document.body.appendChild(canvas);
16
17
18  /*
19  // ★ここからスネークゲームのプログラム★
20  */
21
22  const GRID = 20; //グリッドの1マス（お好みで調整可能）
23  const STAGE = canvas.width / GRID;
24
25  const snake = {
26    x: null,
27    y: null,
28    dx: 1,
29    dy: 0,
30    tail: null,
31
32    update: function() {
33      this.body.push({x: this.x, y: this.y});
34      this.x += this.dx;
35      this.y += this.dy;
36
37      ctx.fillStyle = 'green';
38      this.body.forEach(obj => {
39        ctx.fillRect(obj.x*GRID, obj.y*GRID, GRID-2, GRID-2);
40        if(this.x === obj.x && this.y === obj.y) init(); //自分自身に接触した
41      })
42
43      //ヘビが設定した長さ以上にならないように制限
44      if(this.body.length >= this.tail) this.body.shift();
45    }
46  }
47  const item = {
48    x: null,
49    y: null,
50
51    update: function() {
52      ctx.fillStyle = 'red';
53      ctx.fillRect(this.x*GRID, this.y*GRID, GRID, GRID);
54    }
55  }
56
57  // 初期化处理
58  const init = () => {
```

```

59     snake.x = STAGE / 2;
60     snake.y = STAGE / 2;
61     snake.tail = 4;
62     snake.body = [];
63
64     item.x = Math.floor(Math.random() * STAGE);
65     item.y = Math.floor(Math.random() * STAGE);
66 }
67
68 //メインの繰り返し処理
69 const loop = () => {
70     ctx.clearRect(0,0,canvas.width,canvas.height);
71
72     snake.update();
73     item.update();
74
75     //ヘビが画面外に消えないように制御
76     if(snake.x < 0)         snake.x = STAGE-1;
77     if(snake.y < 0)         snake.y = STAGE-1;
78     if(snake.x > STAGE-1) snake.x = 0;
79     if(snake.y > STAGE-1) snake.y = 0;
80
81     //アイテムを食べれるようにする
82     if(snake.x === item.x && snake.y === item.y) {
83         snake.tail++;
84         item.x = Math.floor(Math.random() * STAGE);
85         item.y = Math.floor(Math.random() * STAGE);
86     }
87 }
88
89 init();
90 setInterval(loop, 1000/15); //15フレームでゲームを描画（お好みで調整可能）
91
92 document.addEventListener('keydown', e => {
93     switch(e.key) {
94         case 'ArrowLeft':
95             snake.dx = -1;snake.dy = 0;
96             break;
97         case 'ArrowRight':
98             snake.dx = 1;snake.dy = 0;
99             break;
100        case 'ArrowUp':
101            snake.dx = 0;snake.dy = -1;
102            break;
103        case 'ArrowDown':
104            snake.dx = 0;snake.dy = 1;
105            break;
106    }
107 });
108 </script>
109 </body>
110 </html>
111

```