小瑪莉

Build School 助教
2022.11

Build School
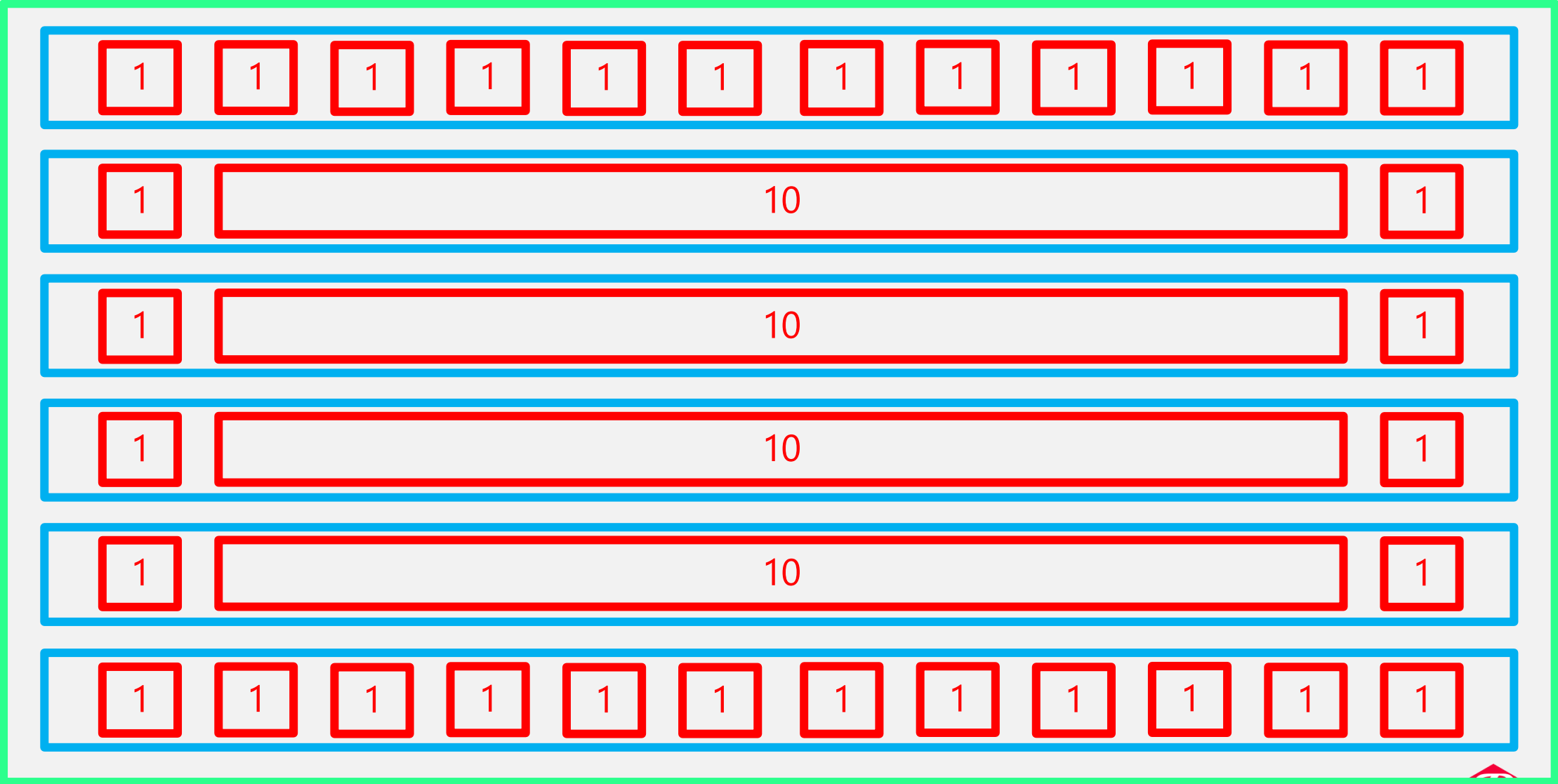
每個格子給一個客製的Attribute(box-id)


```html
<!-- 1 row -->
<div class="row">
    <div box-id="1" class="col-1"></div>
    <div box-id="2" class="col-1"></div>
    <div box-id="3" class="col-1"></div>
    <div box-id="4" class="col-1"></div>
    <div box-id="5" class="col-1"></div>
    <div box-id="6" class="col-1"></div>
    <div box-id="7" class="col-1"></div>
    <div box-id="8" class="col-1"></div>
    <div box-id="9" class="col-1"></div>
    <div box-id="10" class="col-1"></div>
    <div box-id="11" class="col-1"></div>
    <div box-id="12" class="col-1"></div>
</div>
```

Build School
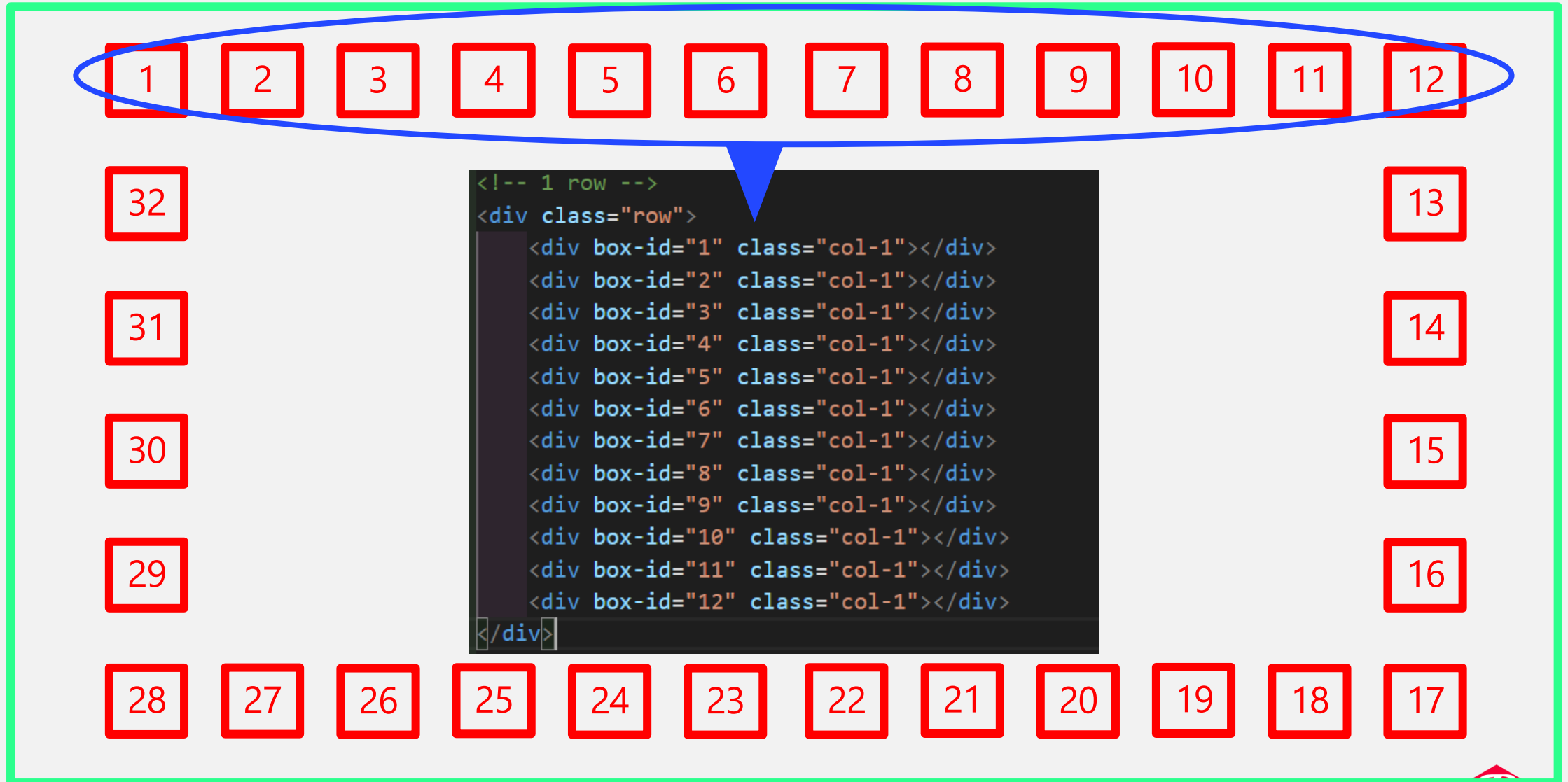
加入CSS樣式

```
<style>
    i {
        font-size: 48px;
    }


    [box-id] {
        width: 80px;        ← Bootstrap5  不需要
        height: 80px;
        display: flex;
        justify-content: center;
        align-items: center;
        border: none;
        transition: 0.3s;
    }


    [box-id].active {
        border: solid 5px ■red;
        box-shadow: 0 0 20px ■red;
        transition: 0s;
    }


    #msg {
        font-size: 36px;
        text-align: center;
    }
</style>
```

資料設計，有**32**格子就有**32**個格子資料

```
{
    id: '1',
    color: 'red',
    icon: 'fab fa-facebook-square',
    target: function () {
        return 'FaceBook';
    }
},
```

id:
格子的編號，與box-id一致

color、icon:
因為這裡用的是Font Awesome的icon所以這邊才這樣設計(記得引用CDN)

target:
當抽到這格時，要觸發的function

Build School

根據設定好的資料，開始渲染畫面

```javascript
let bricks = document.querySelectorAll('[box-id]');
bricks = Array.from(bricks)
.sort((a, b) =>
    {
        return a.getAttributeNode('box-id').value - b.getAttributeNode('box-id').value
    });
console.log(bricks)
bricks.forEach((x, index) => {
    let id = index + 1;
    let data = brickData.find(x => x.id == id);
    let icon = document.createElement('i');
    icon.setAttribute('class', data.icon);
    icon.style.color = data.color
    x.appendChild(icon);
})
```

這邊要注意我渲染畫面時要按照格子的box-id順序，所以這邊要寫自定義排序

Build School

宣告全域變數

```
let steps = 0; //剩餘的步數
let allSteps = 0; //全部的步數
let currentIndex = 0; //目前走到哪一格

let speed; //速率(越大代表越慢)
```

Build School

按下Start按鈕事件

```javascript
//Start Button
document.querySelector('button').addEventListener('click', function () {
    speed = 50;


    let random = Math.floor(Math.random() * brickData.length) + 1;
    //console.log(random);
    steps = random + (3 * bricks.length);
    allSteps = steps;
    turnAround();
});
```

加號後面的算式是為了呈現多跑幾圈才走到獎項

turnAround方法(一般版本)

```javascript
function turnAround() {
    bricks[currentIndex].classList.remove('active');
    currentIndex++;

    if (currentIndex >= bricks.length) currentIndex = 0;

    bricks[currentIndex].classList.add('active');
    steps--;

    //一般版本
    // if (steps > 0) {
    //     setTimeout(turnAround, speed);
    // } else {
    //     //跑完了
    //     let msgBox = document.getElementById('msg');
    //     let val = brickData[target].target();
    //     msgBox.innerText = `接下來30分鐘 -> ${val}`;
    // }
```

turnAround方法(變速版本)

```javascript
//變速版本
if (steps > 0) {

    setTimeout(turnAround, speed);

    //當剩下1/3時，減速度
    if (steps < Math.floor((allSteps / 3))) speed += 7;
} else {
    //跑完了
    let msgBox = document.getElementById('msg');
    let val = brickData[currentIndex].target();
    msgBox.innerText = `接下來30分鐘 -> ${val}`;
}
```

Build School