

Dynamic JavaScript WebPages DOM CSSGrid Create Page Elements

JavaScript DOM how to create page elements update style properties add events to web pages
Interactive Dynamic Web

Explore how you can use JavaScript code to create a fully responsive Dynamically created grid of page elements. Learn more about JavaScript DOM and how you can create interactive page content with JavaScript Code

Please NOTE that JavaScript HTML and CSS are a prerequisite to this course

Included - bonus downloadable PDF resource guide with source code and links to resources.

Dynamic Web Page Grid with JavaScript - Create Elements add Event Listeners to make page elements interactive.

- **Document.querySelector()** - to select page elements from the DOM
- **Document.createElement()** - Create elements with JavaScript
- **EventTarget.addEventListener()** - adding interaction with click events on elements created using JavaScript Code
- Select all the page elements that were created and update the style background color
- Add buttons that allow the user to click and reset new background colors to each page element
- Interactive page elements when clicked will update the background color of the entire body with the random color of the element set when it was created.

Have fun with code while learning more about the Document Object Model and how you can select, create and manipulate the page elements using JavaScript code.

JavaScript code learning made fun with step by step lessons demonstrating at the start of each lesson what will be presented in the lesson. Take this opportunity to learn more about how to code with JavaScript.

```

const output = document.querySelector('.output');
const grid = {rows:5,cols:7};
const total = grid.rows * grid.cols;
const output1 = document.createElement('div');

const output2 = document.createElement('div');
output2.style.margin = '10px auto';
output2.style.border = '1px solid #eee';
output2.style.maxWidth = '80%';
output2.style.textAlign = 'center';
output2.style.padding = '10px';

const btn1 = document.createElement('button');
btn1.textContent = 'Random Colors 1:>';
const btn2 = document.createElement('button');
btn2.textContent = 'Random Colors 2:>';
const holder = [];
btn1.addEventListener('click',(e)=>{
  holder.forEach((el)=>{
    el.style.backgroundColor = ranBack();
  })
})
btn2.addEventListener('click',(e)=>{
  const eles = document.querySelectorAll('.box');
  eles.forEach((el)=>{
    el.style.backgroundColor = ranBack();
  })
})

document.body.prepend(output1);
document.body.prepend(output2);
output2.append(btn1);
output2.append(btn2);
output1.classList.add('output');
output1.style.marginBottom = '10px';
createGrid(total,output,grid.cols);
createGrid(total,output1,grid.cols);

```

```

function ranBack(){
    return '#'+(Math.random().toString(16).substr(-6));
}

function createGrid(tot,parent,cols){
    for(let i=0;i<tot;i++){
        const ele = maker(i,parent);
        holder.push(ele);
        ele.addEventListener('click',updateBack);
    }

parent.style.setProperty(`grid-template-columns`,`repeat(${cols}
,1fr)` )
}

function updateBack(e){
    const ele = e.target;
    console.log(ele.style.backgroundColor);
    document.body.style.backgroundColor =
ele.style.backgroundColor;
}

function maker(i,parent){
    const ele = document.createElement('div');
    ele.textContent = `${i+1}`;
    ele.classList.add('box');
    ele.style.backgroundColor = ranBack();
    return parent.appendChild(ele);
}
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Code</title>
    <style>
        .box{

```

```
    border:1px solid #ddd;
    text-align:center;
    cursor:grab;
}
.output{
    display:grid;
    width:80vw;
    margin:auto;
}
</style>
</head>
<body>
    <div class="output"></div>
    <script src="code1.js"></script>
</body>
</html>
```