Day 8: Buttons Container

by AvmnuSng

Button Containers in JavaScript

Table Of Contents
Creating a Button Container
Scaling Button Layout

We want to create four buttons and lay them out in the form of a 2×2 grid. To do this, we'll define a button container to contain the four buttons and then set the width of the container and the buttons in such a way that only two buttons can be contained in each row of the container.

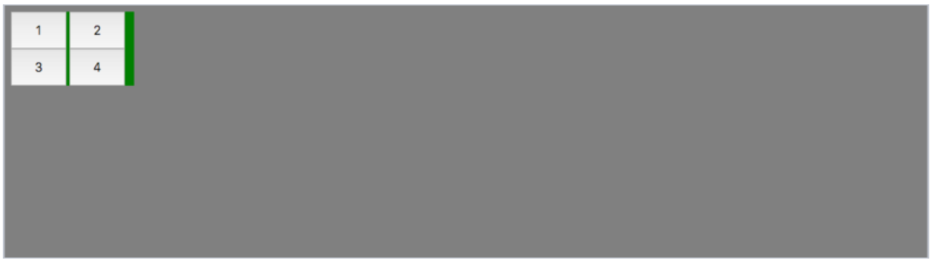
Creating a Button Container

We use the HTML `div` tag to section off (or create a division in) a page. Any items between the `<div>` and `</div>` tags are contained within the *div*.

Consider the following HTML code:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>

        <style>
            /* Style for the 'buttonContainer' class */
            .buttonContainer {
                width: 148px;
            }
            /* Style for 'buttonClass' elements inside a 'buttonContainer'

               We can think of the '>' combinator as a sort of binary operator:

               the syntax 'parent > child' specifies that we're selecting ONLY

               elements of type 'child' that are within elements of type 'parent'.
            */
            .buttonContainer > .buttonClass {
                    width: 72px;
                    height: 48px;
                    font-size: 16px;
            }
        </style>
    </head>

    <body>
        <div id="btns" class="buttonContainer">
            <button id="btn1" class="buttonClass">1</button>
            <button id="btn2" class="buttonClass">2</button>
            <button id="btn3" class="buttonClass">3</button>
            <button id="btn4" class="buttonClass">4</button>
        </div>
    </body>
</html>
```

The code above generates the following page:



In this image, the *gray* background denotes the window and the *green* background denotes the button container.

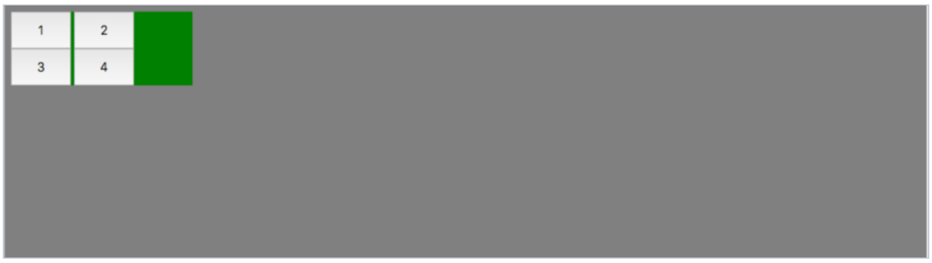
Scaling Button Layout

Setting fixed widths for our container and buttons to ensure they're laid out in a 2×2 grid only works if we know the width of the screen it's being viewed on. To ensure our buttons are laid out exactly how we want them to be regardless of who's viewing them, we can set the width of our elements relative to the width of the document body (i.e., screen).

Consider the following HTML code:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>

        <style>
            .buttonContainer {
                width: 20%;
            }

            .buttonContainer > .buttonClass {
                width: 33%;
                height: 48px;
                font-size: 16px;
            }
        </style>
    </head>

    <body>
        <div id="btns" class="buttonContainer">
            <button id="btn1" class="buttonClass">1</button>
            <button id="btn2" class="buttonClass">2</button>
            <button id="btn3" class="buttonClass">3</button>
            <button id="btn4" class="buttonClass">4</button>
        </div>
    </body>
</html>
```

The code above generates the following page:



Observe that:

- We set the `width` property of our *button container* to 20% of the screen width.
- We set the `width` property of our individual *buttons* relative to the width of our *button container*. This means that if we increase the width of our *buttons* from 33% to 100%, then the width of each *button* will be equal to the width of the *button container*.

These specifications ensure our grid scales properly regardless of the size of the screen displaying it.

EXAMPLE

In this code, we set the width of each *button* within a *button container* to be 100% (i.e., the width of the container).

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
        
            <style>
                .buttonContainer {
                    width: 20%;
                }
                
                .buttonContainer > .buttonClass {
                    width: 100%;
                    height: 48px;
                    font-size: 16px;
                }
        </style>
    </head>
    
    <body>
        <div id="btns" class="buttonContainer">
            <button id="btn1" class="buttonClass">1</button>
            <button id="btn2" class="buttonClass">2</button>
            <button id="btn3" class="buttonClass">3</button>
            <button id="btn4" class="buttonClass">4</button>
        </div>
    </body>
</html>
```

The code above generates the following page: