Day 9: Binary Calculator

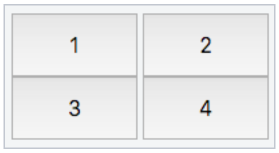
by AvmnuSng

Problem Submissions Leaderboard Discussions Editorial Tutorial

Simple Calculator in JavaScript

Responding to Multiple Click Events

The image below depicts four buttons laid out in a 2×2 grid:



Now, let's write some code so that, when clicked, the clicked button's `innerHTML` increments by 1.

- EXAMPLE

This code uses separate *onclick* functions for each button that increment the button's `innerHTML` when it's clicked.

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .buttonContainer {
                width: 148px;
            }

            .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>
```

Go to Top

```
            <button id='btn4' class='buttonClass'>4</button>
        </div>

        <script>
            document.getElementById('btn1').onclick = function() {
                document.getElementById('btn1').innerHTML++;
            };

            document.getElementById('btn2').onclick = function() {
                document.getElementById('btn2').innerHTML++;
            };

            document.getElementById('btn3').onclick = function() {
                document.getElementById('btn3').innerHTML++;
            };

            document.getElementById('btn4').onclick = function() {
                document.getElementById('btn4').innerHTML++;
            };
        </script>
    </body>
</html>
```

Using a Single Function for All Buttons

We can approach this in a more elegant way by using the *same* function to increment the `innerHTML` for whichever button is clicked.

Approach: *onclick*

The function uses the click event's `target` or `srcElement` properties to get the `id` of the clicked button and modify its `innerHTML`.

- EXAMPLE

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .buttonContainer {
                width: 148px;
            }

            .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>

        <script>
            function action(e) {
                /* Older IE browsers have a srcElement property,
                but other browsers have a 'target' property;
                Set btn to whichever exists. */
                var btn = e.target || e.srcElement;

                /* Get the clicked element's innerHTML */
                document.getElementById(btn.id).innerHTML++;
```

Go to Top

```
            }

            /* Set each button to call action(e) when clicked */
            document.getElementById('btn1').onclick = action;
            document.getElementById('btn2').onclick = action;
            document.getElementById('btn3').onclick = action;
            document.getElementById('btn4').onclick = action;
        </script>
    </body>
</html>
```

Approach: *Event Listener*

The function uses the click event's `target` or `srcElement` properties to get the `id` of the clicked button and modify its `innerHTML`.

- EXAMPLE

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            .buttonContainer {
                width: 148px;
            }

            .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>

        <script>
            /* Parameter 'e' is the click Event */
            function action(e) {
                /* Older IE browsers have a srcElement property,
                but other browsers have a 'target' property;
                Set btn to whichever exists. */
                var btn = e.target || e.srcElement;

                /* Get the clicked element's innerHTML */
                document.getElementById(btn.id).innerHTML++;
            }

            /* Add a click event listener that calls action(e) when cl
icked */
            document.getElementById('btn1').addEventListener('click',
action);
            document.getElementById('btn2').addEventListener('click',
action);
            document.getElementById('btn3').addEventListener('click',
action);
            document.getElementById('btn4').addEventListener('click',
action);
        </script>
    </body>
</html>
```

Go to Top

Resources and Tips

This section reviews some functions that are helpful in completing the Binary Calculator challenge.

The `eval` Function

We can use this function to evaluate a string representing an expression. If the string consists of base-10 integers and mathematical operators, this function calculates the result of the mathematical expression.

EXAMPLE

```
const expression = '5+2-3';
console.log( eval(expression) );
```

Output

Run

Binary Numbers to Integer Strings

To convert a non-base-10 number, num, of radix r to a base-10 integer string, we use the syntax `num.toString(r)`.

EXAMPLE

Sample conversions from non-base-10 numeric strings to base-10 integer strings.

```
const two = '10';
console.log( parseInt(two, 2) );

const three = '11';
console.log( parseInt(three, 2) );

const five = '101';
console.log( parseInt(five, 2) );

const nine = three;
console.log( parseInt(nine, 8) );
```

Output

Run

Integer Division

Because we're implementing a simple calculator with no decimal values, our calculator must perform *integer division*. We can use the `Math.floor` function to ensure that our calculator discards any remainders.

EXAMPLE

```
const result = 3 / 2;
console.log( result );
console.log( Math.floor(result) );
```

Go to Top

Output

Run

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

Go to Top