



# Aditya College of Engineering & Technology

(Permanently Affiliated to JNTU, Kakinada, Approved by AICTE, New Delhi)  
(Accredited by NAAC, Recognized by UGC under Section (2f) and 12(B) of UGC Act 1965)

---

## Coding Practice

**Ms.J.Divya Lalitha Sri,**

Assistant Professor,

Department of C.S.E,

Aditya College of Engineering & Technology,

Surampalem

To improve coding ability, students are asked to make use of web resources to enhance their coding skills. This type of practice helps students in developing more complex coding. Students are advised to calculator program by applying CSS with the help of web resources.



Fig.1. Giving suggestions to coding practice



Fig.2. Clarifying doubts

## To do: Develop Calculator program using Java Script.

```
<h1>JavaScript Calculator</h1>
```

```
<p class="warning">Don't divide by zero</p>
```

```
<div id="calculator" class="calculator">
```

```
  <button id="clear" class="clear">C</button>
```

```
  <div id="viewer" class="viewer">0</div>
```

```
  <button class="num" data-num="7">7</button>
```

```
  <button class="num" data-num="8">8</button>
```

```
  <button class="num" data-num="9">9</button>
```

```
  <button data-ops="plus" class="ops">+</button>
```

```
  <button class="num" data-num="4">4</button>
```

```
  <button class="num" data-num="5">5</button>
```

```
  <button class="num" data-num="6">6</button>
```

```
  <button data-ops="minus" class="ops">-</button>
```

```
  <button class="num" data-num="1">1</button>
```

```
  <button class="num" data-num="2">2</button>
```

```
  <button class="num" data-num="3">3</button>
```

```
  <button data-ops="times" class="ops">*</button>
```

```
  <button class="num" data-num="0">0</button>
```

```
  <button class="num" data-num=".">.</button>
```

```
  <button id="equals" class="equals" data-result="">=</button>
```

```
  <button data-ops="divided by" class="ops">/</button>
```

```
</div>
```

```
<button id="reset" class="reset">Reset Universe?</button>
```

```
html {
```

```
  background: #100a1c;
```

```
  background-image:
```

```
    radial-gradient(50% 30% ellipse at center top, #201e40 0%, rgba(0,0,0,0) 100%),
```

```
    radial-gradient(60% 50% ellipse at center bottom, #261226 0%, #100a1c 100%);
```

```
  background-attachment: fixed;
```

```
  color: #6cacc5;
```

```
}
```

```
body {
```

```
color: #6cacc5;
font: 300 18px/1.6 "Source Sans Pro",sans-serif;
margin: 0;
padding: 5em 0 2em;
text-align: center;
}
```

```
h1 {
  font-weight: 300;
  margin: 0;
}
```

```
/* Gradient text only on Webkit */
```

```
.warning {
  background: -webkit-linear-gradient(45deg, #c97874 10%, #463042 90%);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  color: #8c5059;
  font-weight: 400;
  margin: 0 auto 6em;
  max-width: 9em;
}
```

```
.calculator {
  font-size: 28px;
  margin: 0 auto;
  width: 10em;
```

```
&::before,
&::after {
  content: " ";
  display: table;
}
```

```
&::after {
  clear: both;
}
}
```

```
/* Calculator after dividing by zero */
```

```
.broken {
  animation: broken 2s;
  transform: translate3d(0,-2000px,0);
  opacity: 0;
```

```
}
```

```
.viewer {  
  color: #c97874;  
  float: left;  
  line-height: 3em;  
  text-align: right;  
  text-overflow: ellipsis;  
  overflow: hidden;  
  width: 7.5em;  
  height: 3em;  
}
```

```
button {  
  border: 0;  
  background: rgba(42,50,113, .28);  
  color: #6cacc5;  
  cursor: pointer;  
  float: left;  
  font: inherit;  
  margin: 0.25em;  
  width: 2em;  
  height: 2em;  
  transition: all 0.5s;
```

```
&:hover {  
  background: #201e40;  
}
```

```
&:focus {  
  outline: 0; // Better check accessibility
```

```
/* The value fade-ins that appear */
```

```
&::after {  
  animation: zoom 1s;  
  animation-iteration-count: 1;  
  animation-fill-mode: both; // Fix Firefox from firing animations only once  
  content: attr(data-num);  
  cursor: default;  
  font-size: 100px;  
  position: absolute;  
    top: 1.5em;  
    left: 50%;  
  text-align: center;
```

```

    margin-left: -24px;
    opacity: 0;
    width: 48px;
  }
}

/* Same as above, modified for operators */
.ops:focus::after {
  content: attr(data-ops);
  margin-left: -210px;
  width: 420px;
}

/* Same as above, modified for result */
.equals:focus::after {
  content: attr(data-result);
  margin-left: -300px;
  width: 600px;
}

/* Reset button */

.reset {
  background: rgba(201,120,116,.28);
  color:#c97874;
  font-weight: 400;
  margin-left: -77px;
  padding: 0.5em 1em;
  position: absolute;
  top: -20em;
  left: 50%;
  width: auto;
  height: auto;

  &:hover {
    background: #c97874;
    color: #100a1c;
  }

  /* When button is revealed */
  &.show {
    top: 20em;
    animation: fadein 4s;
  }
}

```

```
}  
}  
  
/* Animations */  
  
/* Values that appear onclick */  
@keyframes zoom {  
  0% {  
    transform: scale(.2);  
    opacity: 1;  
  }  
  
  70% {  
    transform: scale(1);  
  }  
  
  100% {  
    opacity: 0;  
  }  
}  
  
/* Division by zero animation */  
@keyframes broken {  
  0% {  
    transform: translate3d(0,0,0);  
    opacity: 1;  
  }  
  
  5% {  
    transform: rotate(5deg);  
  }  
  
  15% {  
    transform: rotate(-5deg);  
  }  
  
  20% {  
    transform: rotate(5deg);  
  }  
  
  25% {  
    transform: rotate(-5deg);  
  }  
}
```

```
50% {
  transform: rotate(45deg);
}

70% {
  transform: translate3d(0,2000px,0);
  opacity: 1;
}

75% {
  opacity: 0;
}

100% {
  transform: translate3d(0,-2000px,0);
}
}

/* Reset button fadein */
@keyframes fadein {
  0% {
    top: 20em;
    opacity: 0;
  }

  50% {
    opacity: 0;
  }

  100% {
    opacity: 1;
  }
}

@media (min-width: 420px) {
  .calculator {
    width: 12em;
  }
  .viewer {
    width: 8.5em;
  }
  button {
    margin: 0.5em;
  }
}
```

```

}
/*
TODO:
    Limit number input
    Disallow . from being entered multiple times
    Clean up structure
*/

(function() {
    "use strict";

    // Shortcut to get elements
    var el = function(element) {
        if (element.charAt(0) === "#") { // If passed an ID...
            return document.querySelector(element); // ... returns single element
        }

        return document.querySelectorAll(element); // Otherwise, returns a nodelist
    };

    // Variables
    var viewer = el("#viewer"), // Calculator screen where result is displayed
        equals = el("#equals"), // Equal button
        nums = el(".num"), // List of numbers
        ops = el(".ops"), // List of operators
        theNum = "", // Current number
        oldNum = "", // First number
        resultNum, // Result
        operator; // Batman

    // When: Number is clicked. Get the current number selected
    var setNum = function() {
        if (resultNum) { // If a result was displayed, reset number
            theNum = this.getAttribute("data-num");
            resultNum = "";
        } else { // Otherwise, add digit to previous number (this is a string!)
            theNum += this.getAttribute("data-num");
        }

        viewer.innerHTML = theNum; // Display current number
    };

    // When: Operator is clicked. Pass number to oldNum and save operator

```



```

var moveNum = function() {
    oldNum = theNum;
    theNum = "";
    operator = this.getAttribute("data-ops");

    equals.setAttribute("data-result", ""); // Reset result in attr
};

// When: Equals is clicked. Calculate result
var displayNum = function() {

    // Convert string input to numbers
    oldNum = parseFloat(oldNum);
    theNum = parseFloat(theNum);

    // Perform operation
    switch (operator) {
        case "plus":
            resultNum = oldNum + theNum;
            break;

        case "minus":
            resultNum = oldNum - theNum;
            break;

        case "times":
            resultNum = oldNum * theNum;
            break;

        case "divided by":
            resultNum = oldNum / theNum;
            break;

        // If equal is pressed without an operator, keep number and continue
        default:
            resultNum = theNum;
    }

    // If NaN or Infinity returned
    if (!isFinite(resultNum)) {
        if (isNaN(resultNum)) { // If result is not a number; set off by, eg, double-clicking
            operators
            resultNum = "You broke it!";
        } else { // If result is infinity, set off by dividing by zero

```

```

    resultNum = "Look at what you've done";
    el('#calculator').classList.add("broken"); // Break calculator
    el('#reset').classList.add("show"); // And show reset button
  }
}

// Display result, finally!
viewer.innerHTML = resultNum;
equals.setAttribute("data-result", resultNum);

// Now reset oldNum & keep result
oldNum = 0;
theNum = resultNum;

};

// When: Clear button is pressed. Clear everything
var clearAll = function() {
  oldNum = "";
  theNum = "";
  viewer.innerHTML = "0";
  equals.setAttribute("data-result", resultNum);
};

/* The click events */

// Add click event to numbers
for (var i = 0, l = nums.length; i < l; i++) {
  nums[i].onclick = setNum;
}

// Add click event to operators
for (var i = 0, l = ops.length; i < l; i++) {
  ops[i].onclick = moveNum;
}

// Add click event to equal sign
equals.onclick = displayNum;

// Add click event to clear button
el("#clear").onclick = clearAll;

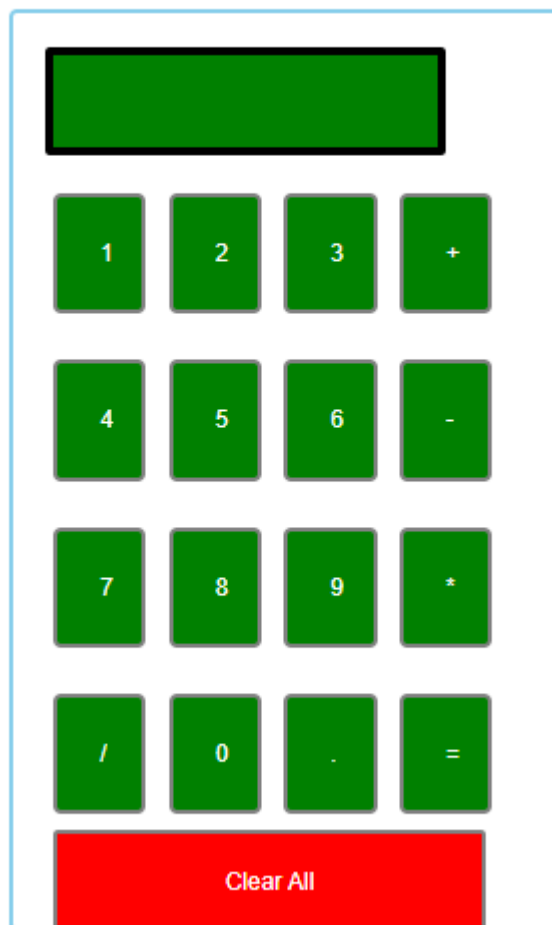
// Add click event to reset button
el("#reset").onclick = function() {

```

```
    window.location = window.location;  
};  
  
}());
```

Output:

## Calculator Program in JavaScript



Ref: <https://www.javatpoint.com/javascript-calculator>