

MINOR PROJRCT

DEVELOPING AN ONLINE CALCULATOR

HTML

```
<html>

<head>

  <link rel="stylesheet" type="text/css" href="style.css">

  <link href="https://fonts.googleapis.com/css?family=Open+Sans:600,700" rel="stylesheet">

  <title>A simple calculator</title>

</head>

<body>

  <div id="container">

    <div id="calculator">

      <div id="result">

        <div id="output-text">

          <div id="history">

            <p id="history-value"></p>

          </div>

          <div id="output">

            <p id="output-value"></p>

          </div>

        </div>

        <div id="output-microphone">

          <span id="microphone">

            <span class="tooltip">Operations are plus, minus, multiply, divide & reminder

              The user input format is number + operation + number<br><br>

              Eg: 2 multiply 5 will be 2 &times 5

            </span>

          </div>

        </div>

      <div id="keyboard">

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

        <button class="operator" id="backspace">CE</button>
```

```

<button class="operator" id="%"><button>
<button class="operator" id="/">÷</button>
<button class="number" id="7">7</button>
<button class="number" id="8">8</button>
<button class="number" id="9">9</button>
<button class="operator" id="*">×</button>
<button class="number" id="4">4</button>
<button class="number" id="5">5</button>
<button class="number" id="6">6</button>
<button class="operator" id="-">-</button>
<button class="number" id="1">1</button>
<button class="number" id="2">2</button>
<button class="number" id="3">3</button>
<button class="operator" id="+">+</button>
<button class="empty" id="empty"></button>
<button class="number" id="0">0</button>
<button class="empty" id="empty"></button>
<button class="operator" id="=">=</button>
</div>
</div>
</div>
<script src="script.js"></script>
</body>
</html>

```

CSS

```

body {
  font-family: 'Open Sans', sans-serif;
  background-color: black;
}
#container {
  width: 1000px;
  height: 550px;
}

```

```
background-image: linear-gradient(rgba(0, 0, 0, 0.3), rgba(0, 0, 0, 0.3)), url(bgImg.jpg);
margin: 20px auto;
}

#calculator {
width: 320px;
height: 520px;
background-color: #eaeef;
margin: 0 auto;
top: 20px;
position: relative;
border-radius: 5px;
box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
}

#result {
height: 120px;
}

#history {
text-align: right;
height: 20px;
margin: 0 20px;
padding-top: 20px;
font-size: 15px;
color: #919191;
}

#output {
text-align: right;
height: 60px;
margin: 10px 20px;
font-size: 30px;
}

#output-text {
width: 250px;
height: 120px;
```

```
float: left;
}
#microphone {
height: 30px;
width: 30px;
float: right;
background-image: url(mic.png);
background-size: 100% 100%;
border-radius: 50%;
margin: 20px;
margin-top: 45px;
}
.record {
animation: microphone-animation 1.5s infinite;
}
@Keyframes microphone-animation {
0% {
transform: scale(0.9);
}
70% {
transform: scale(1);
box-shadow: 0 0 0 10px rgba(0, 148, 255, 0.4);
}
100% {
transform: scale(0.9);
box-shadow: 0 0 0 0 rgba(0, 148, 255, 0);
}
}
.tooltip {
width: 200px;
font-size: 9px;
position: absolute;
background-color: #dddddd;
color: #636363;
```

```
padding: 10px;

top: 45px;

transform: translateX(50px);

border-radius: 5px;

visibility: hidden;
}

.tooltip::before {

content: "";

position: absolute;

border-width: 5px;

border-style: solid;

border-color: transparent #dddddd transparent transparent;

left: 0;

margin-left: -10px;
}

#output-microphone:hover .tooltip {

visibility: visible;
}

#keyboard {

height: 400px;
}

.operator,

.number,

.empty {

width: 50px;

height: 50px;

margin: 15px;

float: left;

border-radius: 50%;

border-width: 0;

font-weight: bold;

font-size: 15px;
}

.number,
```

```
.empty {  
    background-color: #eaeef;  
}  
.number,  
.operator {  
    cursor: pointer;  
}  
.operator:active,  
.number:active {  
    font-size: 13px;  
}  
.operator:focus,  
.number:focus,  
.empty:focus {  
    outline: 0;  
}  
button:nth-child(4) {  
    font-size: 20px;  
    background-color: #20b2aa;  
}  
button:nth-child(8) {  
    font-size: 20px;  
    background-color: #ffa500;  
}  
button:nth-child(12) {  
    font-size: 20px;  
    background-color: #f08080;  
}  
button:nth-child(16) {  
    font-size: 20px;  
    background-color: #7d93e0;  
}  
button:nth-child(20) {  
    font-size: 20px;
```

```
background-color: #9477af;
}
```

JAVASCRIPT

```
function getHistory() {
    return document.getElementById("history-value").innerText;
}
```

```
function printHistory(num) {
    document.getElementById("history-value").innerText = num;
}
```

```
function getOutput() {
    return document.getElementById("output-value").innerText;
}
```

```
function printOutput(num) {
    if (num === "") {
        document.getElementById("output-value").innerText = num;
    } else {
        document.getElementById("output-value").innerText = getFormattedNumber(num);
    }
}
```

```
function getFormattedNumber(num) {
    if (num === "-") {
        return "";
    }
    var n = Number(num);
    var value = n.toLocaleString("en");
    return value;
}
```

```
function reverseNumberFormat(num) {
    return Number(num.replace(/,/g, ""));
}
```

```

}

var operator = document.getElementsByClassName("operator");

for (var i = 0; i < operator.length; i++) {
    operator[i].addEventListener('click', function() {
        if (this.id == "clear") {
            printHistory("");
            printOutput("");
        } else if (this.id == "backspace") {
            var output = reverseNumberFormat(getOutput()).toString();
            if (output) { //if output has a value
                output = output.substr(0, output.length - 1);
                printOutput(output);
            }
        } else {
            var output = getOutput();
            var history = getHistory();
            if (output == "" && history != "") {
                if (isNaN(history[history.length - 1])) {
                    history = history.substr(0, history.length - 1);
                }
            }
            if (output != "" || history != "") {
                output = output == "" ? output : reverseNumberFormat(output);
                history = history + output;
                if (this.id == "=") {
                    var result = eval(history);
                    printOutput(result);
                    printHistory("");
                } else {
                    history = history + this.id;
                    printHistory(history);
                    printOutput("");
                }
            }
        }
    });
}

```



```

    }
  });
}

var number = document.getElementsByClassName("number");

for (var i = 0; i < number.length; i++) {
  number[i].addEventListener('click', function() {
    var output = reverseNumberFormat(getOutput());

    if (output !== NaN) { //if output is a number
      output = output + this.id;
      printOutput(output);
    }
  });
}

var microphone = document.getElementById('microphone');

microphone.onclick = function() {
  microphone.classList.add("record");

  var recognition = new(window.SpeechRecognition || window.webkitSpeechRecognition ||
window.mozSpeechRecognition || window.msSpeechRecognition)();

  recognition.lang = 'en-US';
  recognition.start();

  operations = {
    "plus": "+",
    "minus": "-",
    "multiply": "*",
    "multiplied": "*",
    "divide": "/",
    "divided": "/",
    "reminder": "%"
  }

  recognition.onresult = function(event) {
    var input = event.results[0][0].transcript;

    for (property in operations) {
      input = input.replace(property, operations[property]);
    }

    document.getElementById("output-value").innerText = input;
  }
}

```

```

    setTimeout(function() {
        evaluate(input);
    }, 2000);
    microphone.classList.remove("record");
}
}

function evaluate(input) {
    try {
        var result = eval(input);
        document.getElementById("output-value").innerText = result;
    } catch (e) {
        console.log(e);
        document.getElementById("output-value").innerText = "";
    }
}

```

RESULT



