



SMIT SIKKIM
MANIPAL
UNIVERSITY
SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

Assignment Of Scripting Lab

Submitted By

Name: Sudarshan Singh

Registration: 201900078

Create a calculator app using Angular which is capable of performing following operations:

1. Addition of two numbers
2. Subtraction of two numbers
3. Multiplication of two numbers
4. Division of two numbers
5. Factorial of a number
6. Checking if a given number is Prime or not

App.component.ts

```
calculator > src > app > app.component.ts > AppComponent > getAnswer > isPrime
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'calculator';
10  subDisplayText = '';
11  mainDisplayText = '';
12  operand1: number;
13  operand2: number;
14  operator = '';
15  calculationString = '';
16  answered = false;
17
18  operatorSet = false;
19
20  pressKey(key: string) {
21    if (key === '/' || key === 'x' || key === '-' || key === '+' || key === 'Factorial' || key === 'Prime') {
22      const lastKey = this.mainDisplayText[this.mainDisplayText.length - 1];
23      if (lastKey === '/' || lastKey === 'x' || lastKey === '-' || lastKey === '+' || key === 'Factorial' || key === 'Prime') {
24        this.operatorSet = true;
25      }
26      if ((this.operatorSet) || (this.mainDisplayText === '')) {
27        return;
28      }
29      this.operand1 = parseFloat(this.mainDisplayText);
30      this.operator = key;
31      this.operatorSet = true;
32    }
33    if (this.mainDisplayText.length === 10) {
34      return;
35    }
36    this.mainDisplayText += key;
37  }
38  allClear() {
39    this.mainDisplayText = '';
40    this.subDisplayText = '';
41    this.operatorSet = false;
42  }
43
44  getAnswer() {
45    this.calculationString = this.mainDisplayText;
46    this.operand2 = parseFloat(this.mainDisplayText.split(this.operator)[1]);
47    if (this.operator === '/') {
48      this.subDisplayText = this.mainDisplayText;
49      this.mainDisplayText = (this.operand1 / this.operand2).toString();
50      this.subDisplayText = this.calculationString;
51      if (this.mainDisplayText.length > 9) {
52        this.mainDisplayText = this.mainDisplayText.substr(0, 9);
53      }
54    } else if (this.operator === 'x') {
55      this.subDisplayText = this.mainDisplayText;
56      this.mainDisplayText = (this.operand1 * this.operand2).toString();
57      this.subDisplayText = this.calculationString;
58      if (this.mainDisplayText.length > 9) {
59        this.mainDisplayText = 'ERROR';
60        this.subDisplayText = 'Range Exceeded';
61      }
62    } else if (this.operator === '-') {
63      this.subDisplayText = this.mainDisplayText;
64      this.mainDisplayText = (this.operand1 - this.operand2).toString();
65      this.subDisplayText = this.calculationString;
66    } else if (this.operator === '+') {
67      this.subDisplayText = this.mainDisplayText;
68      this.mainDisplayText = (this.operand1 + this.operand2).toString();
69      this.subDisplayText = this.calculationString;
70      if (this.mainDisplayText.length > 9) {
71        this.mainDisplayText = 'ERROR';
72        this.subDisplayText = 'Range Exceeded';
73      }
74    }
75    else {
76      this.subDisplayText = 'ERROR: Invalid Operation';
77    }
78  }
79 }
```

```

        this.subDisplayText = this.mainDisplayText;
        this.mainDisplayText = (factorial).toString();
        this.subDisplayText = this.calculationString;
        if (this.mainDisplayText.length > 9) {
            this.mainDisplayText = 'ERROR';
            this.subDisplayText = 'Range Exceeded';
        }
    }
    else if (this.operator === 'Prime') {

        this.subDisplayText = this.mainDisplayText;
        this.mainDisplayText = (isPrime(this.operand1)).toString();
        this.subDisplayText = this.calculationString;
        if (this.mainDisplayText.length > 9) {
            this.mainDisplayText = 'ERROR';
            this.subDisplayText = 'Range Exceeded';
        }
    }
    else {
        this.subDisplayText = 'ERROR: Invalid Operation';
    }
    this.answered = true;
}
}

function isPrime(num: number) {
    for(var i = 2; i < num; i++)
        if(num % i === 0) return 'Not Prime';
    return 'Prime';
}

function calcFact( num: number )
{
    var i;
    var fact = 1;
    for( i = 1; i <= num; i++ )
    {
        fact = fact * i;
    }
    return fact;
}

```

App.component.html

```
calculator > src > app > app.component.html > body > div.container > div.row > div.col-md-4 > div.base > div.keypad > table > tr > td.keys.opkey
Go to component
1 <body>
2   <div class="container">
3     <div class="row">
4       <div class="col-md-4"></div>
5       <div class="col-md-4">
6         <div class="base">
7           <div class="maindisplay">
8             <div class="subdisplay">{{ subDisplayText }}</div>
9             {{ mainDisplayText }}
10          </div>
11          <div class="keypad">
12            <table style="width: 100%;">
13              <tr>
14                <td class="keys ackey" colspan="3" (click)="allClear()">AC</td>
15                <td class="keys opkey" colspan="1" (click)="pressKey('/')"></td>
16              </tr>
17              <tr>
18                <td class="keys numkey" (click)="pressKey('7')">7</td>
19                <td class="keys numkey" (click)="pressKey('8')">8</td>
20                <td class="keys numkey" (click)="pressKey('9')">9</td>
21                <td class="keys opkey" (click)="pressKey('x')">x</td>
22              </tr>
23              <tr>
24                <td class="keys numkey" (click)="pressKey('4')">4</td>
25                <td class="keys numkey" (click)="pressKey('5')">5</td>
26                <td class="keys numkey" (click)="pressKey('6')">6</td>
27                <td class="keys opkey" (click)="pressKey('.')">.</td>
28              </tr>
29              <tr>
30                <td class="keys numkey" (click)="pressKey('3')">3</td>
31                <td class="keys numkey" (click)="pressKey('2')">2</td>
32                <td class="keys numkey" (click)="pressKey('1')">1</td>
33                <td class="keys opkey" (click)="pressKey('+')">+</td>
34              </tr>
35              <tr>
36                <td colspan="2" class="keys numkey" (click)="pressKey('0')">0</td>
37                <td class="keys numkey" (click)="pressKey('.')">.</td>
38                <td class="keys equalkey" (click)="getAnswer()">=</td>
39              </tr>
40              <tr>
41                <td colspan="2" class="keys opkey" (click)="pressKey('!')">Factorial</td>
42                <td colspan="2" class="keys opkey" (click)="pressKey('P!')">Prime</td>
43              </tr>
44            </table>
45          </div>
46        </div>
47      </div>
48      <div class="col-md-4"></div>
49    </div>
50  </div>
51 </body>
52
53
```

App.component.css

```
calculator > src > app > app.component.css > .base
1  body {
2    background-color: #000000;
3    box-shadow: 0px 0px 0px 10px #666;
4    border: 5px solid black;
5    border-radius: 10px;
6  }
7  .base {
8    background: black;
9    margin-top: 5vh;
10   margin-left: 65vh;
11   border: 3px solid black;
12   width: 35%;
13 }
14
15
16 .maindisplay {
17   background: #3A4655;
18   height: 20vh;
19   padding: 5% !important;
20   font-size: 4rem;
21   text-align: right;
22   font-family: Courier, monospace;
23   overflow: auto;
24 }
25
26 .subdisplay {
27   border-bottom: 1px solid #727B86;
28   height: 15%;
29   font-size: 2rem;
30   overflow: auto;
31 }
32
33 .keypad {
34   height: calc(50%);
35 }
36
37 .keys {
38   margin: 0;
39   height: 5%;
40   background: whitesmoke;
41   color: #425062;
42   padding: 5%;
```

```
42 padding: 5%;
43 font-size: 2rem;
44 text-align: center;
45 cursor: pointer;
46 opacity: 0.9;
47 }
48
49 .keys:hover {
50   opacity: 1;
51 }
52
53 .ackey {
54   color: red;
55   background: rgb(48, 45, 45);
56 }
57
58 .equalkey {
59   color: white;
60   background-color: orangered;
61 }
62
63 .numkey {
64   color: skyblue;
65   background-color: grey;
66 }
67
68 .opkey {
69   color: white;
70   background-color: rgb(48, 45, 45);
71 }
72
```

Output:

| | | | |
|-----------|---|-------|---|
| | | | |
| AC | | | / |
| 7 | 8 | 9 | x |
| 4 | 5 | 6 | - |
| 3 | 2 | 1 | + |
| 0 | | . | = |
| Factorial | | Prime | |