ng new ngCalculator

**Add Bootstrap 4**

We will use the bootstrap 4 to style our app. Copy the following code to the index.html

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

## Create Calculator Component

Create the calculator.comp`onent.ts & calculator.component.html under the folder calculator

import { Component} from '@angular/core';

@Component({

  selector: 'ng-calculator',

  templateUrl: './calculator.component.html',

})

export class CalculatorComponent {

}

Now open the calculator.component.html

<div class="container-fluid">

  <div class="jumbotron col-sm-4 p-2 m-0 bg-inverse">

    <h1 class="text-center">Angular Calculator</h1>

    //Displays the User Input

    <div class="input-group input-group-sm col-sm-12 m-0 p-0">

      <div class="col-sm-12 form-control text-lg-right" type="text">{{input}}</div>

    </div>

    //Displays the Results

    <div class="input-group input-group-sm col-sm-12 m-0 p-0">

      <div class="form-control text-sm-right" type="text">{{result}}</div>

    </div>

    <div class="col-sm-12 p-1 m-0">

      <button class="btn btn-info col-sm-6" type="button" (click)="allClear()">C</button>

      <button class="btn btn-warning col-sm-3" type="button" (click)="clear()">x</button>

      <button class="btn btn-secondary col-sm-3" type="button" (click)="pressOperator('/')">/</button>

    </div>

    <div class="col-sm-12 p-1 m-0">

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('7')">7</button>

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('8')">8</button>

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('9')">9</button>

      <button class="btn btn-lg btn-secondary col-sm-3 p-1" type="button" (click)="pressOperator('\*')">X</button>

    </div>

    <div class="col-sm-12 p-1 m-0">

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('4')">4</button>

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('5')">5</button>

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('6')">6</button>

      <button class="btn btn-lg btn-secondary col-sm-3 p-1" type="button" (click)="pressOperator('-')">-</button>

    </div>

    <div class="col-sm-12 p-1 m-0">

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('1')">1</button>

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('2')">2</button>

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('3')">3</button>

      <button class="btn btn-lg btn-secondary col-sm-3 p-1" type="button" (click)="pressOperator('+')">+</button>

    </div>

    <div class="col-sm-12 p-1 m-0">

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('.')">.</button>

      <button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('0')">0</button>

      <button class="btn btn-lg btn-success col-sm-6 p-1" type="button" (click)="getAnswer()">=</button>

    </div>

  </div>

calculator.component.ts

import { Component } from '@angular/core';

@Component({

  selector: 'ng-calculator',

  templateUrl: './calculator.component.html',

})

export class CalculatorComponent {

  input:string = '';

  result:string = '';

  pressNum(num: string) {

    //Do Not Allow . more than once

    if (num==".") {

      if (this.input !="" ) {

        const lastNum=this.getLastOperand()

        console.log(lastNum.lastIndexOf("."))

        if (lastNum.lastIndexOf(".") >= 0) return;

      }

    }

    //Do Not Allow 0 at beginning.

    //Javascript will throw Octal literals are not allowed in strict mode.

    if (num=="0") {

      if (this.input=="" ) {

        return;

      }

      const PrevKey = this.input[this.input.length - 1];

      if (PrevKey === '/' || PrevKey === '\*' || PrevKey === '-' || PrevKey === '+')  {

          return;

      }

    }

    this.input = this.input + num

    this.calcAnswer();

  }

  getLastOperand() {

    let pos:number;

    console.log(this.input)

    pos=this.input.toString().lastIndexOf("+")

    if (this.input.toString().lastIndexOf("-") > pos) pos=this.input.lastIndexOf("-")

    if (this.input.toString().lastIndexOf("\*") > pos) pos=this.input.lastIndexOf("\*")

    if (this.input.toString().lastIndexOf("/") > pos) pos=this.input.lastIndexOf("/")

    console.log('Last '+this.input.substr(pos+1))

    return this.input.substr(pos+1)

  }

  pressOperator(op: string) {

    //Do not allow operators more than once

    const lastKey = this.input[this.input.length - 1];

    if (lastKey === '/' || lastKey === '\*' || lastKey === '-' || lastKey === '+')  {

      return;

    }

    this.input = this.input + op

    this.calcAnswer();

  }

  clear() {

    if (this.input !="" ) {

      this.input=this.input.substr(0, this.input.length-1)

    }

  }

  allClear() {

    this.result = '';

    this.input = '';

  }

  calcAnswer() {

    let formula = this.input;

    let lastKey = formula[formula.length - 1];

    if (lastKey === '.')  {

      formula=formula.substr(0,formula.length - 1);

    }

    lastKey = formula[formula.length - 1];

    if (lastKey === '/' || lastKey === '\*' || lastKey === '-' || lastKey === '+' || lastKey === '.')  {

      formula=formula.substr(0,formula.length - 1);

    }

    console.log("Formula " +formula);

    this.result = eval(formula);

  }

  getAnswer() {

    this.calcAnswer();

    this.input = this.result;

    if (this.input=="0") this.input="";

  }

}

Second method

ng new ngCalculator

Copy and paste the following code to the “index.html“.

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

**app.component.ts**

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

})

export class AppComponent {

input: string = '';

result: string = '';

pressNum(num: string) {

//Do Not Allow . more than once

if (num == ".") {

if (this.input != "") {

const lastNum = this.getLastOperand()

console.log(lastNum.lastIndexOf("."))

if (lastNum.lastIndexOf(".") >= 0) return;

}

}

//Do Not Allow 0 at beginning.

//Javascript will throw Octal literals are not allowed in strict mode.

if (num == "0") {

if (this.input == "") {

return;

}

const PrevKey = this.input[this.input.length - 1];

if (PrevKey === '/' || PrevKey === '\*' || PrevKey === '-' || PrevKey === '+') {

return;

}

}

this.input = this.input + num

this.calcAnswer();

}

getLastOperand() {

let pos: number;

console.log(this.input)

pos = this.input.toString().lastIndexOf("+")

if (this.input.toString().lastIndexOf("-") > pos) pos = this.input.lastIndexOf("-")

if (this.input.toString().lastIndexOf("\*") > pos) pos = this.input.lastIndexOf("\*")

if (this.input.toString().lastIndexOf("/") > pos) pos = this.input.lastIndexOf("/")

console.log('Last ' + this.input.substr(pos + 1))

return this.input.substr(pos + 1)

}

pressOperator(op: string) {

//Do not allow operators more than once

const lastKey = this.input[this.input.length - 1];

if (lastKey === '/' || lastKey === '\*' || lastKey === '-' || lastKey === '+') {

return;

}

this.input = this.input + op

this.calcAnswer();

}

clear() {

if (this.input != "") {

this.input = this.input.substr(0, this.input.length - 1)

}

}

allClear() {

this.result = '';

this.input = '';

}

calcAnswer() {

let formula = this.input;

let lastKey = formula[formula.length - 1];

if (lastKey === '.') {

formula = formula.substr(0, formula.length - 1);

}

lastKey = formula[formula.length - 1];

if (lastKey === '/' || lastKey === '\*' || lastKey === '-' || lastKey === '+' || lastKey === '.') {

formula = formula.substr(0, formula.length - 1);

}

console.log("Formula " + formula);

this.result = eval(formula);

}

getAnswer() {

this.calcAnswer();

this.input = this.result;

if (this.input == "0") this.input = "";

}

}

**app.component.html**

<div class="container-fluid">

<div class="jumbotron col-sm-4 p-2 m-0 bg-inverse">

<h1 class="text-center">Angular Calculator</h1>

//Displays the User Input

<div class="input-group input-group-sm col-sm-12 m-0 p-0">

<div class="col-sm-12 form-control text-lg-right" type="text">{{input}}</div>

</div>

//Displays the Results

<div class="input-group input-group-sm col-sm-12 m-0 p-0">

<div class="form-control text-sm-right" type="text">{{result}}</div>

</div>

<div class="col-sm-12 p-1 m-0">

<button class="btn btn-info col-sm-6" type="button" (click)="allClear()">C</button>

<button class="btn btn-warning col-sm-3" type="button" (click)="clear()">x</button>

<button class="btn btn-secondary col-sm-3" type="button" (click)="pressOperator('/')">/</button>

</div>

<div class="col-sm-12 p-1 m-0">

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('7')">7</button>

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('8')">8</button>

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('9')">9</button>

<button class="btn btn-lg btn-secondary col-sm-3 p-1" type="button" (click)="pressOperator('\*')">X</button>

</div>

<div class="col-sm-12 p-1 m-0">

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('4')">4</button>

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('5')">5</button>

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('6')">6</button>

<button class="btn btn-lg btn-secondary col-sm-3 p-1" type="button" (click)="pressOperator('-')">-</button>

</div>

<div class="col-sm-12 p-1 m-0">

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('1')">1</button>

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('2')">2</button>

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('3')">3</button>

<button class="btn btn-lg btn-secondary col-sm-3 p-1" type="button" (click)="pressOperator('+')">+</button>

</div>

<div class="col-sm-12 p-1 m-0">

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('.')">.</button>

<button class="btn btn-lg btn-outline-secondary col-sm-3 p-1" type="button" (click)="pressNum('0')">0</button>

<button class="btn btn-lg btn-success col-sm-6 p-1" type="button" (click)="getAnswer()">=</button>

</div>

</div>