**Lesson03 Custom Numeric Directive**

**Steps:-**

**1-on the custom Directive we set the following code**

import { Directive, HostListener, Self, ElementRef, Input } from '@angular/core';

import { NumberTypes } from '../models/models';

import { NgControl } from '@angular/forms';

import { DecimalPipe, PercentPipe } from '@angular/common';

@Directive({selector: '[appNumeric]'})

export class NumericDirective {

//we declare the HTMLInputElement

private inputElement: HTMLInputElement;

//we declare the following input variable

@Input() numberType;

@Input() minDecimalPoint;

@Input() maxDecimalPoint;

@Input() allowNegative;

@Input() min;

@Input() max;

@Input() currencyFormat;

@Input() value;

//we declare the following dependency injection

//1-Decimal Pipe , Percent Pipe

//2-NgControl : we use it to patch value of the form control in reactive form

//3-elementRef : we use it to pass object that assign selector appNumeric on the HTML element to the ElementRef

constructor(@Self() private ngControl: NgControl, private elementRef: ElementRef, private decimalPipe: DecimalPipe, private percentagePipe: PercentPipe) {

this.inputElement = this.elementRef.nativeElement;}

//we intialize the getNumericValue of the inputElement

ngOnInit() {this.getNumericValue(this.inputElement.value);}

//we assign event blur on the element html that call the method getNumericValue

@HostListener("blur", ["$event.target.value"])

onBlur(value) {

if (value){this.getNumericValue(value);}

//we assign event focus on the element html that call the method getNumericValue

@HostListener("focus", ["$event.target.value"])

onfocus(value) {

if (value)

this.getNumericValue(value);}

getNumericValue(value) {

if (!this.allowNegative && value < 0) {

value = value \* -1;}

let digitInfo = "1.";

let numericValue = "0";

if (this.numberType == NumberTypes.Decimal) {

digitInfo += this.minDecimalPoint ? this.minDecimalPoint : '0';

digitInfo += this.maxDecimalPoint ? '-' + this.maxDecimalPoint : '-3';

if (value.indexOf(',') > -1) {

let arrValue = value.split(',');

let v = '';

arrValue.forEach(element => {

v += element;});

value = parseFloat(v);}

numericValue = this.decimalPipe.transform(value, digitInfo);

if (!this.currencyFormat)

numericValue = numericValue.replace(",", "");}

else if (this.numberType == NumberTypes.Percentage) {

numericValue = this.percentagePipe.transform((parseFloat(value) / 100), "1.0-5");}

else if (this.numberType == NumberTypes.Integer) {

digitInfo += '0-0';

numericValue = this.decimalPipe.transform(value, digitInfo).replace(",", "");}

else

numericValue = value;

if (this.min && numericValue < this.min)

numericValue = this.min;

if (this.max && numericValue > this.max)

numericValue = this.max;

this.inputElement.value = numericValue;

this.ngControl.control.patchValue(this.inputElement.value);}

//we assign event keypress on the element html that call the method getNumericValue

@HostListener('keypress', ['$event'])

keyEvent(event: KeyboardEvent) {

if (this.numberType != undefined && this.numberType != '') {

let regexStr = '^[0-9\_\\.\\-]\*$';

return new RegExp(regexStr).test(event.key);}}}