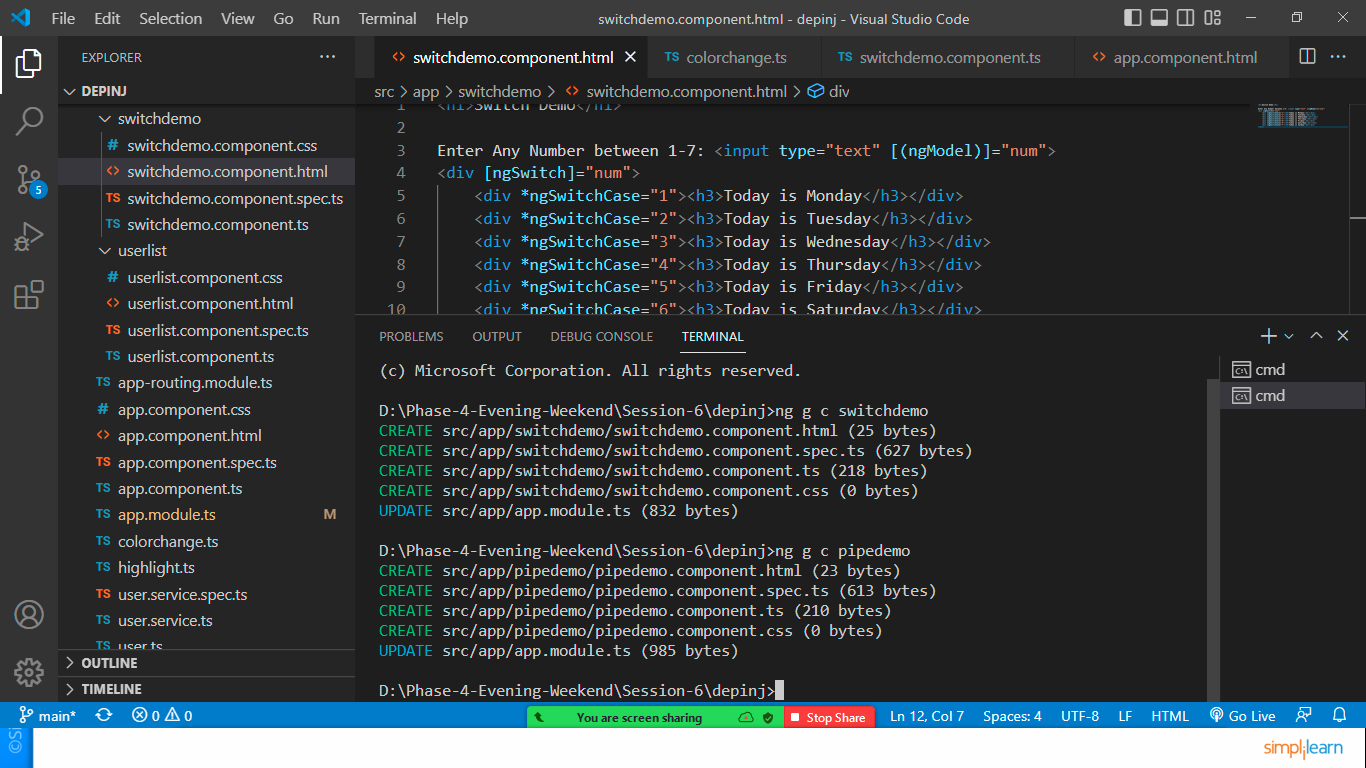
Pipe Demo

Create a component



Include component in app.comp.html

<div class="container">

  <app-pipedemo></app-pipedemo>

</div>

Open pipedemo.component.ts

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

@Component({

  selector: 'app-pipedemo',

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

  styleUrls: ['./pipedemo.component.css']

})

export class PipedemoComponent {

  name:string="sonam soni"

  message:string="GOOD MORNING"

  pi:number=3.1415927

  a=0.12345

  b=12345.6789

  today=new Date();

  object={name:"sonam",

  email:"sonam@gmail.com"}

}

Edit pipedemo.component.html

<H1>Pipe Demo</H1>

<h3>Uppercase and Lowercase Pipe</h3>

<hr>

<p>{{name}}, after applying pipe {{name | uppercase}}</p>

<p>{{message}},after pipe: {{message | lowercase}}</p>

<h3>Number Pipe</h3>

<hr>

<p>pi: {{pi}}</p>

<p>pi: {{pi | number:'5.5-5'}}</p>

<p>pi: {{pi | number:'.10-10'}}</p>

<p>pi: {{pi | number:'.3-6'}}</p>

<h3>Currency Pipe</h3>

<hr>

<!--Dfeault value of display currency is true and

    decimal value is .2-2 -->

<p>{{a | currency: 'INR'}}</p>

<p>{{a | currency: 'USD':true:'3.3-3'}}</p>

<p>{{a | currency: 'EUR':true:'5.3-4'}}</p>

<p>{{b | currency: 'INR':true:'5.2-4'}}</p>

<h3>Date Pipe</h3>

<hr>

<p>Today is {{today}}</p>

<p>Default:  {{today | date}}</p>

<p>Short Date:  {{today | date : 'shortDate'}}</p>

<p>Long Date:  {{today | date: 'longDate'}}</p>

<p>Format:  {{today | date: 'dd/MM/yyyy'}}</p>

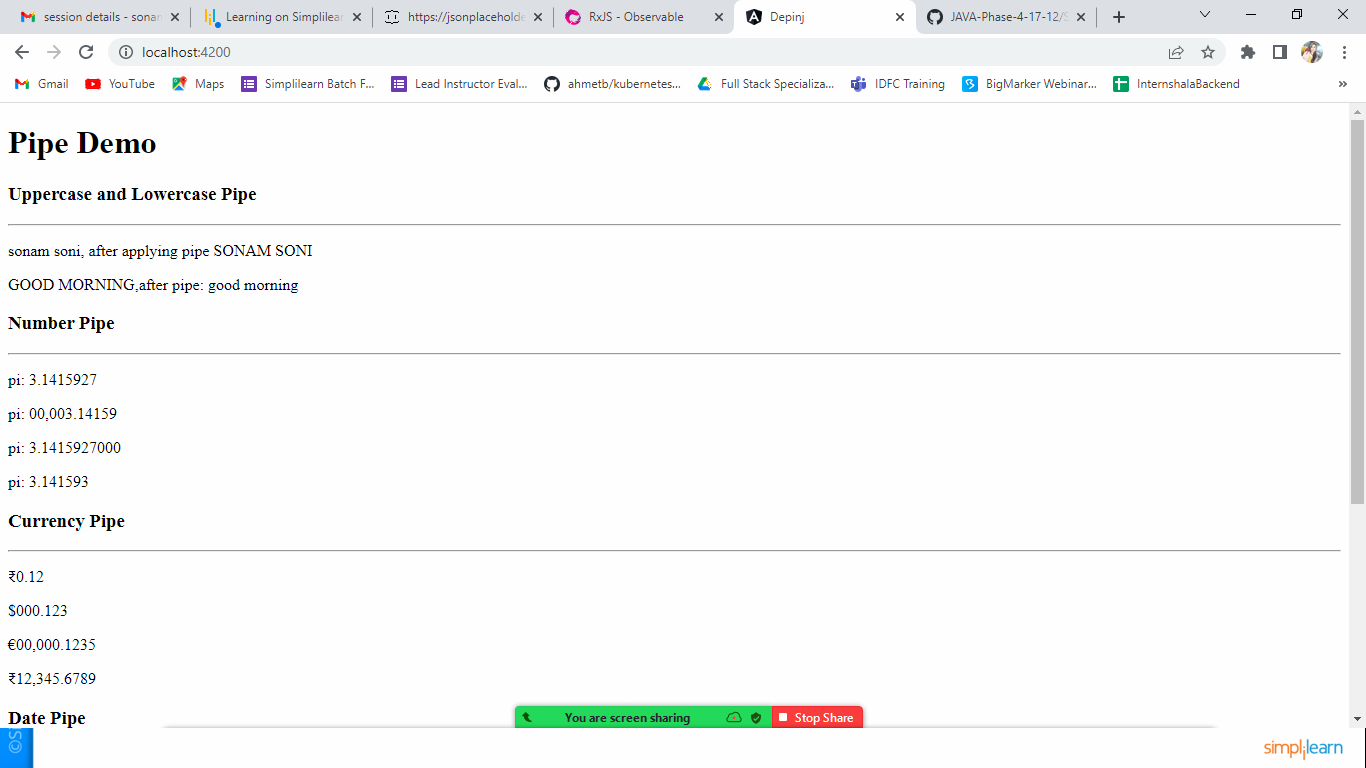
<h3>Json Pipe</h3>

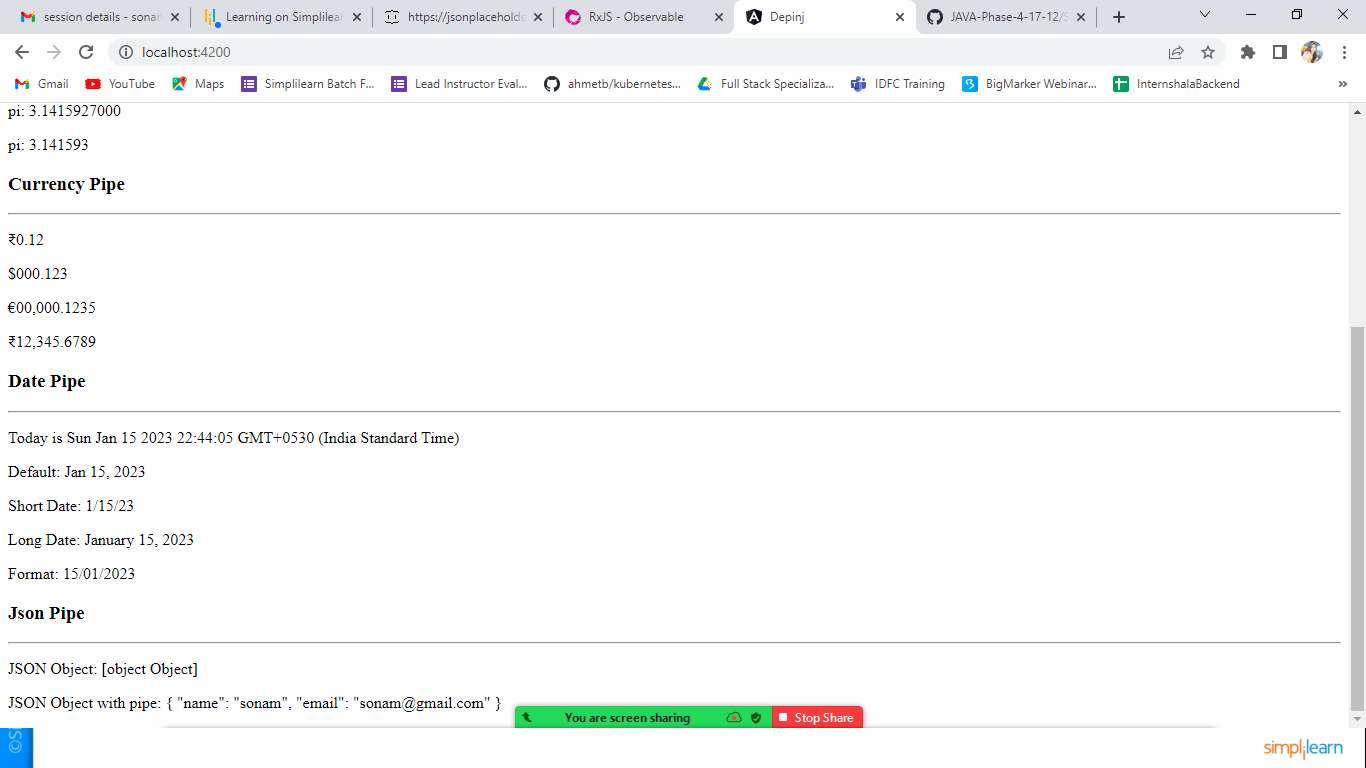
<hr>

<p>JSON Object: {{object}}</p>

<p>JSON Object with pipe: {{object | json}}</p>

Save both files and check the output





Custom Pipe

Create one folder named pipes inside the app folder

Inside pipe folder create one file named formattext.ts

import { Pipe, PipeTransform } from "@angular/core";

@Pipe({

    name:'changespace'

})

export class FormatText implements PipeTransform{

    //overridden method of interface

    transform(value: string, data:string) {

        return value.replace(' ',data)

    }

}

Let’s Declare it inside app.module.ts

import { FormatText } from './pipes/formattext';

@NgModule({

  declarations: [

    AppComponent,

    UserlistComponent,

    Highlight,

    ColorChange,

    SwitchdemoComponent,

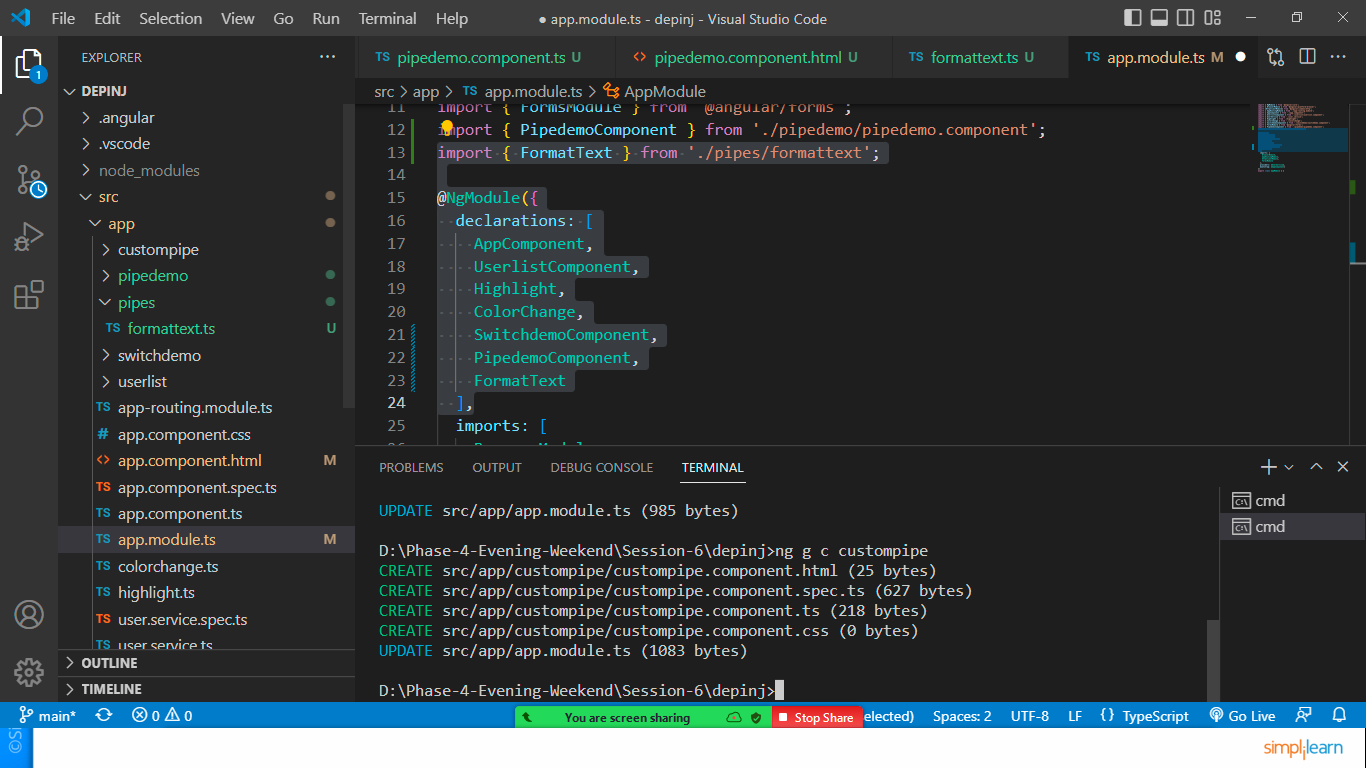
    PipedemoComponent,

    FormatText

  ],

Let’s use this custom pipe.

Create component called custompipe: ng g c custompipe



Let’s edit custompipe.component.ts

export class CustompipeComponent {

  name="sonam soni"

}

Let’s edit customepipe.component.html

<h1>Custome Pipe</h1>

<p>Original Value: {{name}}</p>

<p>Custom Value: {{name | changespace : '#'}}</p>

<p>Custom Value: {{name | changespace : ','}}</p>

<p>Custom Value: {{name | changespace : '@'}}</p>

Let’s include this component in app component

<div class="container">

  <app-custompipe></app-custompipe>

</div>

