# Pipes in Angular

We can use pipes to transform our output in template .

Pipes are referred as filters. It helps to transform data and manage data within interpolation, denoted by {{ | }}. It accepts data, arrays, integers and strings as inputs which are separated by ‘|’ symbol.

**Date Pipe :**

Decalare presentDate variable in typescript file

presentDate = new Date();

Now, add the below code in your aboutus.component.html file.

<div>

Today's date :- {{presentDate}}

</div>

Now, run the application, it will show the following output –

Today's date :- Mon Jun 15 2020 10:25:05 GMT+0530 (IST)

Here,

Date object is converted into easily readable format.

Add Date pipe

Let’s add date pipe in the above html file.

<div>

Today's date :- {{presentDate | date }}

</div>

You could see the below output –

Today's date :- Jun 15, 2020

**Parameters in Date**

We can add parameter in pipe using : character. We can show short, full or formatted dates using this parameter.

<div>

short date :- {{presentDate | date:'shortDate' }} <br/>

Full date :- {{presentDate | date:'fullDate' }} <br/>

Formatted date:- {{presentDate | date:'M/dd/yyyy'}} <br/>

Hours and minutes:- {{presentDate | date:'h:mm'}}

</div>

You could see the below response on your screen –

short date :- 6/15/20

Full date :- Monday, June 15, 2020

Formatted date:- 6/15/2020

Hours and minutes:- 12:00

**Chained pipes**

We can combine multiple pipes together. This will be useful when a scenario associates with more than one pipe that has to be applied for data transformation.

<div>

Date with uppercase :- {{presentDate | date:'fullDate' | uppercase}} <br/>

Date with lowercase :- {{presentDate | date:'medium' | lowercase}} <br/>

</div>

You could see the below response on your screen –

Date with uppercase :- MONDAY, JUNE 15, 2020 Date with lowercase :- jun 15, 2020, 12:00:00 am

**AsyncPipe**

If data comes in the form of observables, then Async pipe subscribes to an observable and returns the transmitted values.

Now create custom observable in component typescript file.

import {  of } from 'rxjs';

import { delay } from 'rxjs/operators';

 myObservable: any;

  ngOnInit() {

    this.myObservable = of('Some text').pipe(

      delay(3000)

    )

  }

Add the below code inside your aboutus.component.html.

<div>

    My Observable :- {{myObservable | async}}

</div>

**CurrencyPipe**

It is used to convert the given number into various countries currency format.

Declare one variable in aboutus.component.ts

price : number = 20000;

Add code in aboutus.component.html

<div>

    <h3>

        Currency Pipe

    </h3>

    <p> {{price | currency:'USD'}}</p>

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

</div>

Ouput will be

$20,000.00

₹20,000.00

**JsonPipe**

It is used to transform a JavaScript object into a JSON string.

Declare one object in aboutus.component.ts

 jsonData = {

   id: 'one', name: 'codemind'

 }

Add code in aboutus.component.html

<div>

    {{jsonData | json}}

</div>

Output :-

{ "id": "one", "name": "codemind" }

**Creating custom pipe**

As we have seen already, there is a number of pre-defined Pipes available in Angular 8 but sometimes, we may want to transform values in custom formats.

Syntax : n g p digitcount

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

@Pipe({

  name: 'digitcount'

})

export class DigitcountPipe implements PipeTransform {

  transform(val: number): number {

    return val.toString().length;

  }

}

Declare one variable in aboutus.component.ts

digits : number = 1000;

Add code in aboutus.component.html

<div>

    <p>DigitCount Pipe</p>

    <h1>{{digits | digitcount}}</h1>

</div>

Output :-

4