



Angular 6/7 | Create Custom Pipes AKA Filters in Angular 2.X

Our applications may have a lot of data for presentation, But for creating a good user experience it is preferable to see in a more understandable format like March 12, 2010 is better to read then Mon Mar 12 2010 15:23:40 GMT-0700 (Pacific Daylight Time).

<p>Globalization Partners</p> <p>Globalization Partners</p>	<p>Not All PEOs Are Cr Questions to Ask Be Global PEO</p>
--	---

For such small and repeated tasks of transforming values in other formats we used to use Filters in AngularJS version 1, But after Angular 2 we have pipes which serve similar tasks and once defined can be used anywhere in the application.

1. NETWORK MONITORING SOFTWARE	>
2. 10 FASTEST INTERNET PROVIDERS	>
3. BEST VPN SOFTWARE	>
4. CAREERS IN COMPUTER NETWORKING	>
5. COMPLIANCE MONITORING	>

We will discuss more Pipes using Angular's latest stable version 7. Our todo list will have the following tasks for Pipes.

How to Create Simple Custom Pipes?

How to pass parameters in custom Pipes?

How to Use OrderBy Filter Pipe in Angular 5+ Versions?

Other Usefull Custom Angular 2+ Filters/ Pipes

SPONSORED SEARCHES

angular 7 custom filter pipe



import export data



typescript tutorial



angular example



How to Create Simple Custom Pipes?

First, create a new Angular application by running following in CMD Prompt using Angular CLI.

Install NodeJS then Angular CLI as follows

```
1 $ npm install -g @angular/cli
```

JavaScript

Create new Angular project

```
1 $ ng new NG7Pipes
2 $ cd NG7Pipes
3 $ ng serve --open
```

JavaScript

As now we have a new project ready, create a new file **custom.pipe.ts** in the app folder.

```
1 import { Pipe, PipeTransform } from '@angular/core';
2
3 @Pipe({
4   name: 'appendDollar'
5 })
6 export class appendDollarPipe implements PipeTransform {
7   transform(value:string, pretext:string): string{
8     return value+pretext;
9   }
10 }
```

JavaScript

This is the simplest Pipe which we just created above. This will simply append a \$ sign after value we pass then return it.

Add this pipe in **app.module.ts** file in declarations array.

Add even more warmth to
home with hardwood floors

SHOP OUR HARDWOOD SELECTION

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { FormsModule } from '@angular/forms';
5 import { AppComponent } from './app.component';
```

JavaScript

Privacy - Terms

```

6  import { appendDollarPipe } from './pipes/appendText.pipe';
7
8  @NgModule({
9    declarations: [
10     AppComponent,
11     appendDollarPipe
12   ],
13   imports: [
14     BrowserModule,
15     FormsModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }

```

In **app.component.ts** file, define some value to variable to test our Pipe.

SPONSORED SEARCHES

angular 7 custom filter pipe	Q	import export data	C
typescript tutorial	Q	angular cli	C

```

1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.scss']
7  })
8  export class AppComponent {
9    title = 'Angular 7 Pipes';
10   costOfMyCoffeeCup:number;
11
12   constructor(){
13     this.costOfMyCoffeeCup = 6;
14   }
15 }

```

In the **app.component.html** file, we can use the filter as follows

```

1  <p>
2    {{costOfMyCoffeeCup | appendDollar}}
3  </p>

```

This will output as **6\$**

How to pass parameters in custom Pipes?

In some situations, we may need to pass some parameters with value. To pass parameters we use a colon (:) separated values up to any number.

for example, let's modify above Pipe to take currency name then show output.

```

1  @Pipe({
2    name: 'appendCurrency'
3  })
4  export class appendCurrencyPipe implements PipeTransform {
5    transform(value:string, currency:string): string{
6      return value+currency;
7    }
8  }

```

In HTML we will change usage as follows



JavaScript

```
1 {{costOfMyCoffeeCup | appendCurrency:'EUR'}}
```

How to Use OrderBy Filter Pipe in Angular 5+ Versions?

According to Angular them, new versions of doesn't provide filters/ pipes for list ordering. They removed OrderBy filter as it was poor on performance and aggressive minification processes. You can read more about the details [here](#).

But here we will only discuss if Angular not providing it then What to do? Don't worry we have a solution for it 😊 We can use [ngx-order-pipe](#) package in our application to achieve the same.

How to use Ngx Order Pipe?

This package is very simple to install and use in the application.

Step 1 – Install the package by running the following

JavaScript

```
1 $ npm install ngx-order-pipe --save
```

Step 2 – Import in the app's main module **app.module.ts**

JavaScript

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { FormsModule } from '@angular/forms';
5 import { AppComponent } from './app.component';
6 import { OrderModule } from 'ngx-order-pipe';
7
8 @NgModule({
9   declarations: [
10     AppComponent
11   ],
12   imports: [
13     BrowserModule,
14     FormsModule,
15     OrderModule
16   ],
17   providers: [],
18   bootstrap: [AppComponent]
19 })
20 export class AppModule { }
```

Step 3 – Now its available to be used anywhere in the application.

let's try it on some dummy Data. In **app.component.ts** replace the following code

JavaScript

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 export class AppComponent {
9   title = 'Angular 7 Pipes';
10   dummyData: any[] = [
11     { name: 'G' },
12     { name: 'B' },
13     { name: 'Z' },
14     { name: 'A' },
```

[Privacy](#) - [Terms](#)

```

15     { name: 'H' }
16   ];
17
18   constructor() { }
19 }

```

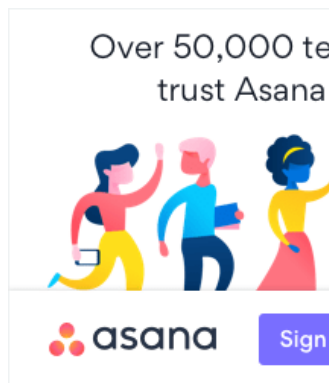
In **app.component.html** use, OrderBy Pipe/ Filter as follows

```

1      <ul>
2        <li *ngFor="let item of (dummyData | orderBy:'name') ">
3          {{item.name}}
4        </li>
5      </ul>

```

Check more advanced usage options [here](#)



Indian Rupee (INR) Currency Filter Pipe in Angular 2 Plus Versions

```

1  @Pipe({
2    name:'INRCurrency'
3  })
4  export class INRCurrencyPipe implements PipeTransform {
5    transform(value:number): string{
6
7      value = Math.round(value);
8      var result = value.toString().split('.');
9      var lastThree = result[0].substring(result[0].length - 3);
10     var otherNumbers = result[0].substring(0, result[0].length - 3);
11     if (otherNumbers != '' && otherNumbers != '-')
12       lastThree = ',' + lastThree;
13     var output = otherNumbers.replace(/\B(?=(\d{2})+(?!\d))/g, ",") + lastThree;
14
15     if (result.length > 1) {
16       output += "." + result[1];
17     }
18
19     return "₹"+output;
20   }
21 }
22

```

This will convert **10000000000** into **₹1,00,00,00,00,000**

Angular Pipes can be used in many ways to reuse logic and transform data on views. Check more details on Angular Pipes [here](#)

Checkout Similar Posts:

- Angular 7/6 + Typescript: Create Filter List with Check-boxes to Select from List
- Angular 7/6 | Add AutoComplete in Angular Application using Angular Material
- Angular 7/6 | Handle HTTP Request and Response using Interceptors in Angular 4.3...
- Ionic 4 | How to Call a Number Directly from Ionic 4 Native Application

- Bootstap in Angular 7 | How to use Bootstrap Components in Angular 4+ Versions in...

Subscribe to Latest Tutorials

SUBSCRIBE

📅 December 27, 2018 👤 Jolly.exe 📁 **Angular, Angular 5, Angular 6, Angular 7**

[CONTACT US](#) | [DONATE!](#) | [OTHER TOOLS](#)

Copyright © www.FreakyJolly.com 2019. All rights reserved.