Pipes in Angular

Every application starts out with what seems like a simple task: get data, transform them, and show them to users. Getting data could be as simple as creating a local variable or as complex as streaming data over a WebSocket.

Once data arrives, for a good user experience, we can transform the view of data using pipes. SO, Angular pipes, a way to write display-value transformations that you can declare in your HTML.

Built-in pipes

The following are built-in pipes of Angular framework.

uppercase

lowercase

titlecase

currency

percent

json

Syntax to apply pipe:

{{ variable-name | pipe-name : attributes }}

Ex:

Consider the following data in a class.

```
name:string="good morning to all of you";

salary:number=54000;

rateOfInterest=0.25;

today=new Date();
```

Now apply pipes to that data to transform their view like below

```
<div>
   Date: {{today}}
</div>

<div>
   Date with date pipe: {{today  | date}}
</div>

<div>
   Date with date pipe with short attribute: {{today  | date :'short'}}
</div>

<div>
Upper case :{{ name | uppercase}}
</div>

<div>
   Lower case :{{ name | lowercase}}
</div>

<div>
   Initcap case :{{ name | titlecase}}
</div>
<div>
   original case :{{ name }}
</div>

<div>
   salary:{{salary}}
</div>

<div>
   salary in dollars:{{salary | currency}}
</div>

<div>
   salary in INR:{{salary | currency : "INR"}}
</div>

<div>
   Rate of interest : {{rateOfInterest | percent}}
</div>


<div>
   value is : {{v}}
</div>

<div>
   square of the value is : {{ v | square}}
</div>

<div>
   square root of the value is : {{ v | squareroot}}
</div>
```

Please visit this link to know more about date pipe https://angular.io/api/common/DatePipe

Custom pipe

The facilty to create cutom pipe is provided by Angular using the “PipeTransform” interface.

```
interface PipeTransform {
    transform(value: any, ...args: any[]): any
}
```

An interface that is implemented by pipes in order to perform a transformation. Angular invokes the transform method with the value of a binding as the first argument, and any parameters as the second argument in list form.

To create a pipe ,use the following CLI command

```
ng generate pipe pipe-name
```

Let us create a simple pipe to find square of a given number

```
ng generate pipe square
```

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

@Pipe({
  name: 'square'
})
export class SquarePipe implements PipeTransform {

  transform(n: number, ...args: unknown[]): number {
    return n*n;
  }

}
```

Let us apply it.

n:number=20;

{{n | square}} will print 400.