Pipes

A pipe takes in data as input and transforms it to a desired output. For Example,

If data=”Rajeev”

{{ data | uppercase}} will produce RAJEEV

Here uppercase is the pipe applied on the input called data.

Generally if we apply a pipe on array it will be filtered out.

There are inbuilt pipes in angular to transform number into currency text, date formatting , number formatting and percentage calculation

We can create our own custom pipes which can be applied on strings, numbers, dates and arrays.

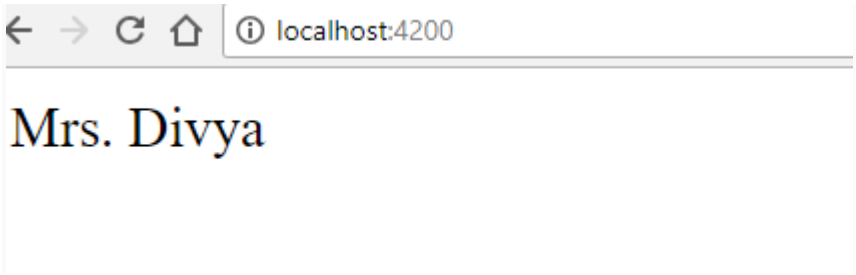
Custom Pipes

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

@Pipe({
  name: 'greet'
})
export class GreetPipe implements PipeTransform {

  transform(value: any, gender: any,status): any {
    if(gender == 'M' )
      return "Mr. "+value;
    if(gender == 'F' && status == 'married')
      return  "Mrs. "+value;
    else
      return "Miss. "+value;
  }

}
```

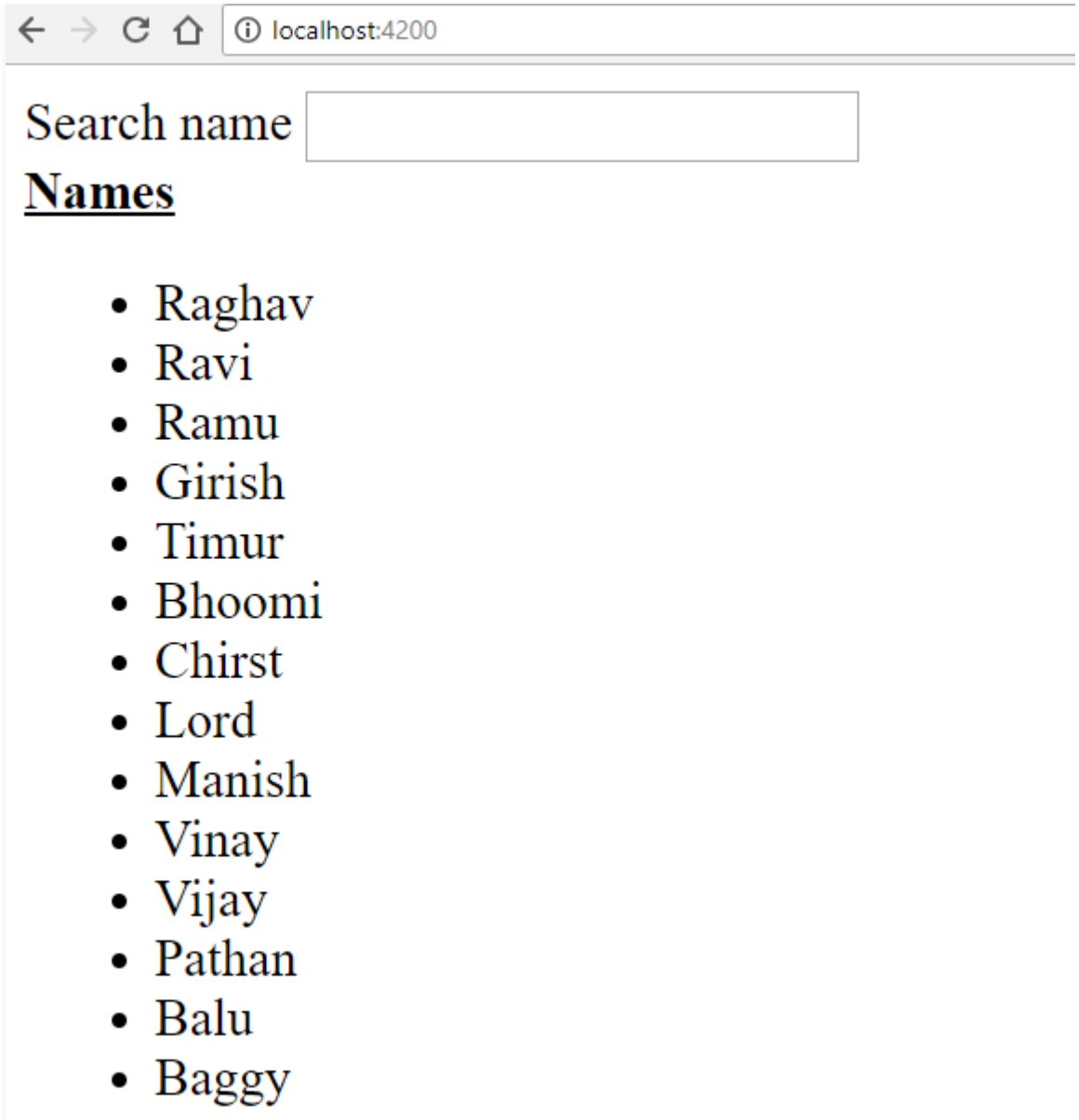
In the above pipe , if we apply it on a string it returns value based on the gender and marital status, here is the syntax how it is applied

```
1  {{"Divya" | greet:'F':'married'}}
```

The pipe will produce the following output



A custom pipe can also be applied on a array consider the following case



Before applying the pipes, the data appears as follows. And we create a pipe to pass a pattern and filter the string

And It is applied as follows

```
dule.ts   # child.component.css   <> app.component.html ×   TS filternames.pipe.ts   TS app.component.ts
1  Search name <input type="text" [(ngModel)]="pattern">
2  <br>
3  <b><u>Names</u></b>
4  <br>
5  <ul>
6      <li *ngFor="let n of developers |fiternames:pattern">{{n}}</li>
7  </ul>
8
```

It produces following result