Custom Pipes in Angular :

You can define a custom pipe by implementing a TypeScript class with the @Pipe decorator. A pipe must have two things:

A name, specified in the pipe decorator
A method named transform that performs the value transformation.
The TypeScript class should additionally implement the PipeTransform interface to ensure that it satisfies the type signature for a pipe.

Here is an example of a custom pipe that transforms strings to kebab case:

```
import { Pipe, PipeTransform } from '@angular/core';
@Pipe({
 name: 'kebabCase',
})
export class KebabCasePipe implements PipeTransform {
 transform(value: string): string {
   return value.toLowerCase().replace(/ /g, '-');
 }
}
```

To generate pipes in Angular we have to follow the commands :

```
ng generate  pipe “pipeName”
```

Open the ts file and make changes as we require.
After changes we have to import the custom pipe in the component ts file then we are able to use the custom pipe.

Example : Creating custom pipe for phone number extension.

- Open terminal and use command to create : ng g p phoneNumberExtension
- Open the phoneNumberExtension.ts file and make changes.
- Save the changes.
- Import the phoneNumberExtension pipe in the component typescript file in the imports array.
- Open component html file to use the custom pipe (phoneNumber Extension)
- Save the changes.

It will successfully work.

I created a Reverse String :

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

@Pipe({
  name: 'reverse'
})
```

```
export class ReversePipe implements PipeTransform {

  transform(value : string): string {
    let reversedString : string ='';
    for (let index = value.length-1; index >= 0; index--){
      reversedString += value[index];
    }
    return reversedString;
  }

}
```

Phone Number Extension Pipe :

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

@Pipe({
  name: 'phoneNumberExtension'
})
export class PhoneNumberExtensionPipe implements PipeTransform {

  transform(value: number): string {
    if (value.toString().length == 10){
      return '+91'+value
    } return 'NA';
  }

}
```