

## **\*\* Pipes \*\***

Pipes transform data in the template before displaying it. We can think of them as filters or formatters we can apply directly in our HTML.

Syntax: `{{ value | pipeName: pipeArgument }}`  
data ← angular pipe ← optional parameter(s)

### **\* Common Built-in pipes :-**

- Date Pipe → Format dates.

Eg. .ts

```
today = new Date();
```

.html

```
<p>Today: {{ today | date: 'fullDate' }} </p>
```

output:

Today: Thursday, September 12, 2025

- UpperCase/LowerCase pipes → Changes text casing
- Currency Pipe → Formats numbers as currency  
args: 'CAD', 'INR', etc.
- Decimal/Number Pipe → Formats numbers with decimal points.

Eg: `<p>{{ score | number: '1.2-2' }} </p>`  
↳ 2 decimal digits

### **\* Custom Pipes :-**

to create, CLI cmd:-

```
> ng g pipe pipes/<pipeName>
```

Eg:- to filter todos:-

```
@Pipe({
  name: 'filterTodos', // call it using this in pipes and pass the searchTerm as argument.
  standalone: true,
})
export class FilterTodosPipe implements PipeTransform {
  transform(todos: Todo[], searchTerm: string): Todo[] {
    if (!searchTerm) {
      return todos;
    }
    const text = searchTerm.toLowerCase();
    return todos.filter((todo) => {
      return todo.title.toLowerCase().includes(text);
    });
  }
}
```