



Angular Pipes — Complete Overview

TRANSFORM DATA | BETTER TEMPLATES | CLEANER APPS

BUILD
LEARN
GROW
WITH
ANGULAR

1 What are Angular Pipes?

Angular pipes transform data before displaying it in a template. They change only the displayed value; the original data remains unchanged.



Template example:

```
{{ name | uppercase }}
```

2 Types of Pipes

SIMPLE
PIPES
POWERFUL
POSSIBILITIES



Built-in Pipes

Ready-made pipes provided by Angular.



Custom Pipes

User-defined pipes created with the `@Pipe` decorator.



Pipe Classification

- **Pure pipe:** runs when input value/reference changes (default)
- **Impure pipe:** can run during every change-detection cycle

3 Built-in Pipes with Examples

USE POWERFUL BUILT-IN PIPES TO FORMAT DATA EFFORTLESSLY

#	Pipe	Purpose	Example	Output
1	UpperCasePipe	Converts text to uppercase	{{ 'angular' uppercase }}	ANGULAR
2	LowerCasePipe	Converts text to lowercase	{{ 'ANGULAR' lowercase }}	angular
3	TitleCasePipe	Capitalizes each word	{{ 'learn angular' titlecase }}	Learn Angular
4	DecimalPipe	Formats decimal numbers	{{ 1234.5 number:'1.2-2' }}	1,234.50
5	PercentPipe	Formats as percentage	{{ 0.75 percent }}	75%
6	CurrencyPipe	Formats currency	{{ 2500 currency:'INR':'symbol' }}	₹2,500.00
7	DatePipe	Formats date and time	{{ today date:'dd-MM-yyyy' }}	14-09-2026
8	SlicePipe	Extracts part of string/array	{{ 'Angular' slice:0:3 }}	Ang
9	JsonPipe	Displays an object as JSON	{{ user json }}	{ "name": "Syed" }
10	AsyncPipe	Reads Observable/Promise values	{{ users\$ async }}	Latest emitted value

4 Import Details (Standalone Angular)

Import only the pipes you use

```
import { Component } from '@angular/core';
import { UpperCasePipe, DatePipe, CurrencyPipe }
  from '@angular/common';

@Component({
  selector: 'app-demo',
  standalone: true,
  imports: [UpperCasePipe, DatePipe, CurrencyPipe],
  templateUrl: './demo.html'
})
export class Demo {}
```

NgModule-based applications: import `CommonModule` in the feature module.

CommonModule provides most common built-in pipes and directives.

5 Parameterized Pipe Syntax

```
{{ value | pipeName:parameter1:parameter2 }}
```

Use a colon (:) to pass parameters.

Example:

```
{{ 2500 | currency:'INR':'symbol' }}
```

6 Key Points

- ✓ Pipes use the `|` symbol in templates.
- ✓ Built-in and custom pipes are the two main types.
- ✓ Pipes can be chained:
`{{ name | lowercase | titlecase }}`
- ✓ Pipes can accept parameters separated by colons.
- ✓ Pipes format displayed data without changing the original value.
- ✓ Pure pipes are the default and offer better performance.
- ✓ In standalone components, add required pipes to the imports array.
- ✓ Avoid expensive calculations inside pipes.

What are Angular Pipes?

Angular Pipes *transform data into a required display format* inside an HTML template *without changing the original value*.

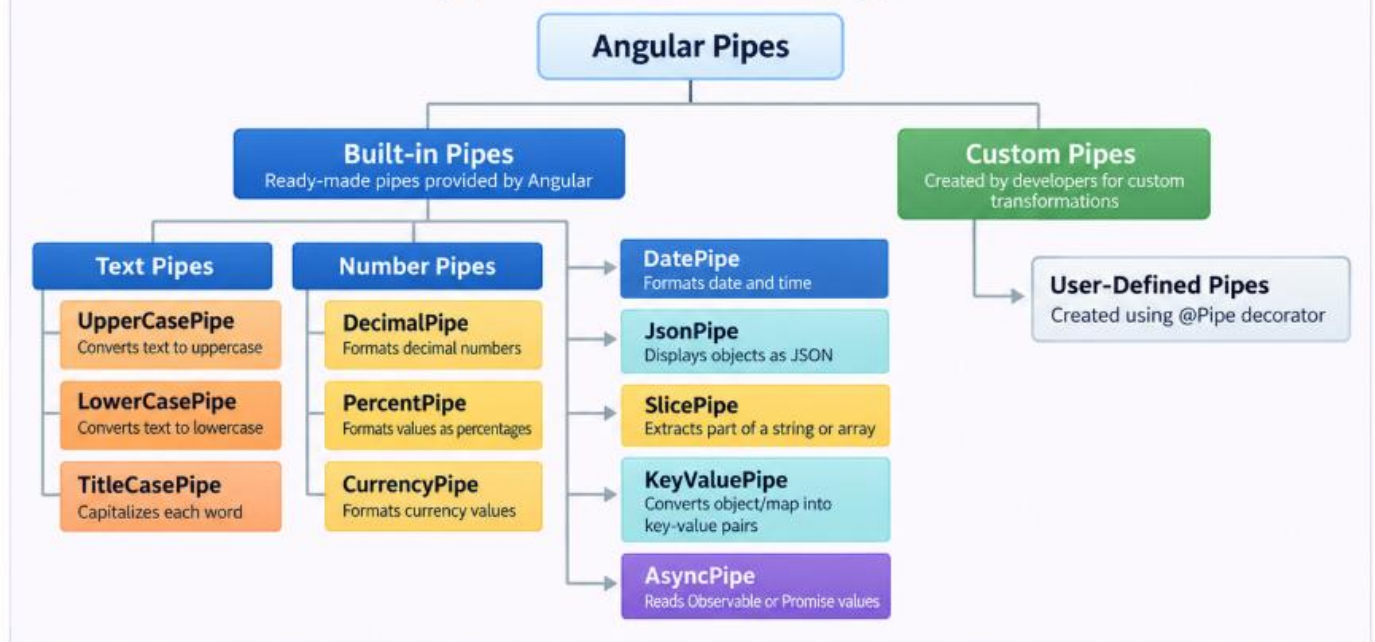
Pipes in Angular



Pipes transform data before displaying it in the template.

Types of Pipes in Angular?

Types of Pipes in Angular



What are Text Pipes in Angular?

Text Pipes are Angular's built-in pipes used to change the **display format of string values** in an HTML template.

*They do **not** change the original value stored in the component.*



TYPES OF TEXT PIPES

Format String Values for Display

BUILD
LEARN
SHARE
ANGULAR

Better
Developers
Brighter
Web Apps

<Angular Pipes/>

Original Value → **Text Pipe** → Formatted Output

The original component value remains unchanged.



UPPERCASE PIPE

Converts all letters into uppercase.

Syntax `{{ value | uppercase }}`

Input angular tutorial

Output ANGULAR TUTORIAL



LOWERCASE PIPE

Converts all letters into lowercase.

Syntax `{{ value | lowercase }}`

Input ANGULAR TUTORIAL

Output angular tutorial



TITLECASE PIPE

Capitalizes the first letter of each word.

Syntax `{{ value | titlecase }}`

Input angular pipes tutorial

Output Angular Pipes Tutorial

KEY POINTS

-  Built-in Angular pipes
-  Used inside interpolation with the `|` operator
-  Change only displayed text, not original data
-  Standalone: import `UpperCasePipe`, `LowerCasePipe` and `TitleCasePipe` — or `CommonModule`

— Syed Mohamed | Build Today. A Better Tomorrow.

SHARE KNOWLEDGE SPREAD GROWTH

 **REPOST**

Types of Text Pipes

Pipe	Purpose	Input	Output
uppercase	Converts all letters to uppercase	angular pipes	ANGULAR PIPES
lowercase	Converts all letters to lowercase	ANGULAR PIPES	angular pipes
titlecase	Capitalizes the first letter of each word	angular pipes	Angular Pipes

What are Number Pipes in Angular?

Number Pipes are Angular built-in pipes used to format numeric values, currencies, and percentages inside an HTML template.

They change only the display format. The original value in the component remains unchanged.



Number Pipes in Angular

Number pipes format numeric values before displaying them in the template.

Transform Numbers Beautifully

NUMBER VALUE
1234.5



FORMATTED OUTPUT
1,234.50

Same Value Better Display

1.2 DecimalPipe
Formats a number with decimal places.

Syntax `{{ value | number:'1.2-2' }}`

Example value = 1234.5

Output 1,234.50

% PercentPipe
Converts a decimal value into a percentage.

Syntax `{{ value | percent:'1.0-0' }}`

Example value = 0.75

Output 75%

₹ CurrencyPipe
Formats a number as a currency value.

Syntax `{{ value | currency:'INR':'symbol' }}`

Example value = 2500

Output ₹2,500.00

Standalone imports:
DecimalPipe, PercentPipe, CurrencyPipe from [@angular/common](#)

Key Points:
Use `:` to pass parameters • Formatting does not change the original value

BUILD BETTER WEB APPS WITH ANGULAR

LEARN | PRACTICE | GROW



Share Knowledge Spread Angular!

Types of Number Pipes

Pipe	Purpose
number (DecimalPipe)	Formats a number and controls decimal digits
currency (CurrencyPipe)	Formats a number as currency
percent (PercentPipe)	Formats a decimal value as a percentage

Slice and JSON Pipes in Angular

Both are Angular built-in pipes. The **Slice Pipe** extracts part of a string or array, while the **JSON Pipe** displays an object in JSON format.

They affect only the displayed output and do not modify the original data.

SLICE AND JSON PIPES IN ANGULAR
Transform Data for Display — Original Data Remains Unchanged

SLICE PIPE RETURNS A PORTION OF A STRING OR ARRAY.

Syntax: `{{ value | slice:start:end }}`

Rules:

- start index is included
- end index is excluded
- negative index counts from the end

STRING EXAMPLE

Value: `courseName = 'Angular Tutorial'`

Template: `{{ courseName | slice:0:7 }}`

Output: Angular

ARRAY EXAMPLE

Value: `skills = ['C#', 'SQL', 'Angular', 'Azure']`

Template: `{{ skills | slice:1:3 }}`

Output: SQL, Angular

JSON PIPE CONVERTS AN OBJECT OR ARRAY INTO READABLE JSON TEXT.

Syntax: `{{ value | json }}`

Purpose:

- Mainly used for development and debugging
- Use `<pre>` to preserve formatting

OBJECT EXAMPLE

Value: `employee = { id: 101, name: 'Syed' }`

Template: `<pre>{{ employee | json }}</pre>`

Output:

```
{
  "id": 101,
  "name": "Syed"
}
```

 JSON OUTPUT

KEY DIFFERENCE

SLICE PIPE: Extracts selected content from strings or arrays.

JSON PIPE: Displays objects or arrays as JSON-formatted text.

KEY POINTS

- Both are built-in Angular pipes
- They change only displayed output
- Standalone: import SlicePipe and JsonPipe — or CommonModule

Created by Syed Mohamed

1. Slice Pipe

The Slice Pipe returns a portion of a string or array based on starting and ending indexes.

2. JSON Pipe

The JSON Pipe converts an object or array into a JSON-formatted string for display.

It is primarily useful for **debugging and development**.

What are Parameterized Pipes in Angular?

Parameterized Pipes are pipes that **accept one or more arguments** to control **how a value is formatted or transformed**.

Parameters are passed after the pipe name using the colon (:) symbol.



Parameterized Pipes in Angular

Parameterized pipes accept one or more arguments to control how a value is formatted or transformed.

INPUT VALUE

PIPE + PARAMETERS

CUSTOMIZED OUTPUT

```
{{ value | pipeName:parameter1:parameter2 }}
```

Use a colon (:) to pass each parameter.

DatePipe

Controls the date format.

Input:

today = new Date(2026, 8, 14)

Template:

{{ today | date:'dd-MM-yyyy' }}

Output:

14-09-2026

DecimalPipe

Controls minimum and maximum digits.

Input:

price = 1234.567

Template:

{{ price | number:'1.2-2' }}

Output:

1,234.57

CurrencyPipe

Controls currency code and symbol display.

Input:

amount = 2500

Template:

{{ amount | currency:'INR':'symbol' }}

Output:

₹2,500.00

SlicePipe

Controls the start and end positions.

Input:

name = 'Angular'

Template:

{{ name | slice:0:3 }}

Output:

Ang

Key Points

- Parameters follow the pipe name
- Multiple parameters are separated by :
- Parameters can be literal values or component properties
- The original value is not changed

Standalone components: import the required pipe from @angular/common

REPOST