**Understanding Pure and Impure Pipes in Angular**

**The Basics**

Angular pipes are a powerful mechanism for transforming data within templates. They offer a declarative way to format, filter, and manipulate data without cluttering component logic. Essentially, pipes take an input value and return a transformed output.

[1. Angular Pipes Explained: How to Transform Template Data Directly - freeCodeCamp](https://www.freecodecamp.org/news/angular-pipes-explained-how-to-transform-template-data-directly/" \l ":~:text=Pipes%20transform%20template%20data%20directly,class%20lean%20of%20basic%20transformations." \t "_blank)

[[Source icon](https://www.freecodecamp.org/news/angular-pipes-explained-how-to-transform-template-data-directly/#:~:text=Pipes%20transform%20template%20data%20directly,class%20lean%20of%20basic%20transformations.)](https://www.freecodecamp.org/news/angular-pipes-explained-how-to-transform-template-data-directly/" \l ":~:text=Pipes%20transform%20template%20data%20directly,class%20lean%20of%20basic%20transformations." \t "_blank)

[www.freecodecamp.org](https://www.freecodecamp.org/news/angular-pipes-explained-how-to-transform-template-data-directly/" \l ":~:text=Pipes%20transform%20template%20data%20directly,class%20lean%20of%20basic%20transformations." \t "_blank)

[2. Pure and Impure pipes in Angular - Mahimarnab Hajra - Medium](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438" \l ":~:text=Pipes%20are%20a%20way%20to,complex%20logic%20in%20our%20component." \t "_blank)

[[Source icon](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438#:~:text=Pipes%20are%20a%20way%20to,complex%20logic%20in%20our%20component.)](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438" \l ":~:text=Pipes%20are%20a%20way%20to,complex%20logic%20in%20our%20component." \t "_blank)

[mahimarnab.medium.com](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438" \l ":~:text=Pipes%20are%20a%20way%20to,complex%20logic%20in%20our%20component." \t "_blank)

[3. Understanding Pipes - Angular](https://angular.io/guide/pipes-overview" \l ":~:text=What%20is%20a%20pipelink,and%20return%20a%20transformed%20value." \t "_blank)

[[Source icon](https://angular.io/guide/pipes-overview#:~:text=What%20is%20a%20pipelink,and%20return%20a%20transformed%20value.)](https://angular.io/guide/pipes-overview" \l ":~:text=What%20is%20a%20pipelink,and%20return%20a%20transformed%20value." \t "_blank)

[angular.io](https://angular.io/guide/pipes-overview" \l ":~:text=What%20is%20a%20pipelink,and%20return%20a%20transformed%20value." \t "_blank)

* **Pure Pipes:**
  + Executed only when the input value changes.

[1. Change detection with pipes - Angular](https://angular.dev/guide/pipes/change-detection" \l ":~:text=By%20default%2C%20pipes%20are%20defined,the%20input%20value%20or%20parameters." \t "_blank)

[[Source icon](https://angular.dev/guide/pipes/change-detection#:~:text=By%20default%2C%20pipes%20are%20defined,the%20input%20value%20or%20parameters.)](https://angular.dev/guide/pipes/change-detection" \l ":~:text=By%20default%2C%20pipes%20are%20defined,the%20input%20value%20or%20parameters." \t "_blank)

[angular.dev](https://angular.dev/guide/pipes/change-detection" \l ":~:text=By%20default%2C%20pipes%20are%20defined,the%20input%20value%20or%20parameters." \t "_blank)

* + Ideal for calculations or transformations based solely on input data.

[1. Pure and Impure pipes in Angular - Mahimarnab Hajra - Medium](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438" \t "_blank)

[[Source icon](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438)](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438" \t "_blank)

[mahimarnab.medium.com](https://mahimarnab.medium.com/pure-and-impure-pipes-in-angular-d918bafb2438" \t "_blank)

* + Improve performance by avoiding unnecessary re-calculations.
* **Impure Pipes:**
  + Executed on every change detection cycle, regardless of input changes.

[1. Explain pure and impure pipe in Angular - GeeksforGeeks](https://www.geeksforgeeks.org/explain-pure-and-impure-pipe-in-angular/" \l ":~:text=Impure%20pipes%20execute%20every%20time,It%20uses%20the%20impure%20function." \t "_blank)

[[Source icon](https://www.geeksforgeeks.org/explain-pure-and-impure-pipe-in-angular/#:~:text=Impure%20pipes%20execute%20every%20time,It%20uses%20the%20impure%20function.)](https://www.geeksforgeeks.org/explain-pure-and-impure-pipe-in-angular/" \l ":~:text=Impure%20pipes%20execute%20every%20time,It%20uses%20the%20impure%20function." \t "_blank)

[www.geeksforgeeks.org](https://www.geeksforgeeks.org/explain-pure-and-impure-pipe-in-angular/" \l ":~:text=Impure%20pipes%20execute%20every%20time,It%20uses%20the%20impure%20function." \t "_blank)

* + Useful for accessing external data or performing actions with side effects.

[1. Angular Pipes 2: Understanding Pure Pipes and Impure Pipes | by Yuvaraj S - Medium](https://medium.com/@yuvidev/angular-pipes-2-understanding-pure-pipes-and-impure-pipes-dbe99d43a492" \l ":~:text=Use%20pure%20pipes%20when%20dealing,depend%20on%20external%20state%20changes." \t "_blank)

[[Source icon](https://medium.com/@yuvidev/angular-pipes-2-understanding-pure-pipes-and-impure-pipes-dbe99d43a492#:~:text=Use%20pure%20pipes%20when%20dealing,depend%20on%20external%20state%20changes.)](https://medium.com/@yuvidev/angular-pipes-2-understanding-pure-pipes-and-impure-pipes-dbe99d43a492" \l ":~:text=Use%20pure%20pipes%20when%20dealing,depend%20on%20external%20state%20changes." \t "_blank)

[medium.com](https://medium.com/@yuvidev/angular-pipes-2-understanding-pure-pipes-and-impure-pipes-dbe99d43a492" \l ":~:text=Use%20pure%20pipes%20when%20dealing,depend%20on%20external%20state%20changes." \t "_blank)

* + Can impact performance if used excessively or inefficiently.

[1. The essential difference between pure and impure pipes in Angular and why that matters](https://angularindepth.com/posts/1061/the-essential-difference-between-pure-and-impure-pipes-in-angular-and-why-that-matters" \l ":~:text=Conclusion,not%20used%20wisely%20and%20carefully." \t "_blank)

[[Source icon](https://angularindepth.com/posts/1061/the-essential-difference-between-pure-and-impure-pipes-in-angular-and-why-that-matters#:~:text=Conclusion,not%20used%20wisely%20and%20carefully.)](https://angularindepth.com/posts/1061/the-essential-difference-between-pure-and-impure-pipes-in-angular-and-why-that-matters" \l ":~:text=Conclusion,not%20used%20wisely%20and%20carefully." \t "_blank)

[angularindepth.com](https://angularindepth.com/posts/1061/the-essential-difference-between-pure-and-impure-pipes-in-angular-and-why-that-matters" \l ":~:text=Conclusion,not%20used%20wisely%20and%20carefully." \t "_blank)

**Built-in Pipes: A Deep Dive**

Angular provides a set of built-in pipes for common transformations:

* **DatePipe:** Formats dates according to locale and specified format.

TypeScript

{{ birthDate | date:'medium' }}

Use code [with caution.](/faq#coding)

* **UpperCasePipe:** Converts text to uppercase.

TypeScript

{{ name | uppercase }}

Use code [with caution.](/faq#coding)

[1. UpperCasePipe - Angular](https://v17.angular.io/api/common/UpperCasePipe" \l ":~:text=UpperCasePipelink&Transforms%20text%20to%20all%20upper%20case." \t "_blank)

[[Source icon](https://v17.angular.io/api/common/UpperCasePipe#:~:text=UpperCasePipelink&Transforms%20text%20to%20all%20upper%20case.)](https://v17.angular.io/api/common/UpperCasePipe" \l ":~:text=UpperCasePipelink&Transforms%20text%20to%20all%20upper%20case." \t "_blank)

[v17.angular.io](https://v17.angular.io/api/common/UpperCasePipe" \l ":~:text=UpperCasePipelink&Transforms%20text%20to%20all%20upper%20case." \t "_blank)

* **LowerCasePipe:** Converts text to lowercase.

TypeScript

{{ title | lowercase }}

Use code [with caution.](/faq#coding)

[1. LowerCasePipe - Angular](https://v17.angular.io/api/common/LowerCasePipe" \l ":~:text=Usage%20noteslink,text%20to%20all%20lower%20case." \t "_blank)

[[Source icon](https://v17.angular.io/api/common/LowerCasePipe#:~:text=Usage%20noteslink,text%20to%20all%20lower%20case.)](https://v17.angular.io/api/common/LowerCasePipe" \l ":~:text=Usage%20noteslink,text%20to%20all%20lower%20case." \t "_blank)

[v17.angular.io](https://v17.angular.io/api/common/LowerCasePipe" \l ":~:text=Usage%20noteslink,text%20to%20all%20lower%20case." \t "_blank)

* **NumberPipe:** Formats numbers with different styles (currency, percent, etc.).

TypeScript

{{ price | currency:'USD' }}

Use code [with caution.](/faq#coding)

* **JsonPipe:** Stringifies JSON objects.

TypeScript

{{ data | json }}

Use code [with caution.](/faq#coding)

**The PipeTransform Interface**

All custom pipes must implement the PipeTransform interface. This interface defines the transform method, which is the core of the pipe's logic.

TypeScript

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

@Pipe({ name: 'myCustomPipe' })

export class MyCustomPipe implements PipeTransform {

transform(value: any, ...args: any[]): any

[1. medium.com](https://medium.com/@softwaretechsolution/angular-pipes-and-custom-pipe-8515b3bde7d0" \t "_blank)

[medium.com](https://medium.com/@softwaretechsolution/angular-pipes-and-custom-pipe-8515b3bde7d0" \t "_blank)

{

// Transformation logic here

return transformedValue;

}

}

Use code [with caution.](/faq#coding)

**Creating Custom Pipes**

You can create custom pipes to handle specific transformations not covered by built-in pipes.

[1. Angular Pipe Customization: Building Your Own Data Transformers - Medium](https://medium.com/@ayushgrwl365/angular-pipe-customization-building-your-own-data-transformers-15974ad0da83" \l ":~:text=Built%2Din%20pipes%20provide%20common,to%20cater%20to%20unique%20data" \t "_blank)

[[Source icon](https://medium.com/@ayushgrwl365/angular-pipe-customization-building-your-own-data-transformers-15974ad0da83#:~:text=Built%2Din%20pipes%20provide%20common,to%20cater%20to%20unique%20data)](https://medium.com/@ayushgrwl365/angular-pipe-customization-building-your-own-data-transformers-15974ad0da83" \l ":~:text=Built%2Din%20pipes%20provide%20common,to%20cater%20to%20unique%20data" \t "_blank)

[medium.com](https://medium.com/@ayushgrwl365/angular-pipe-customization-building-your-own-data-transformers-15974ad0da83" \l ":~:text=Built%2Din%20pipes%20provide%20common,to%20cater%20to%20unique%20data" \t "_blank)

TypeScript

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

@Pipe({ name: 'highlight' })

export class HighlightPipe implements PipeTransform {

transform(value:

[1. www.xspdf.com](https://www.xspdf.com/resolution/58019799.html" \t "_blank)

[www.xspdf.com](https://www.xspdf.com/resolution/58019799.html" \t "_blank)

string, term: string): string {

if

[1. github.com](https://github.com/javico2609/angular-mvd-meetup" \t "_blank)

[github.com](https://github.com/javico2609/angular-mvd-meetup" \t "_blank)

(!value || !term) {

return value;

}

return value.replace(new RegExp(term, 'gi'), '<span class="highlight">$&</span>');

}

}

Use code [with caution.](/faq#coding)

**Performance Implications of Impure Pipes**

Impure pipes can significantly impact performance, especially in large applications or with frequent change detection cycles. Consider the following factors:

[1. Increase Your Apps Performance With Angular Pipes | by Joscelyn James | Medium](https://medium.com/@josce.james7/increase-your-apps-performance-with-angular-pipes-4690ff001cf9" \t "_blank)

[[Source icon](https://medium.com/@josce.james7/increase-your-apps-performance-with-angular-pipes-4690ff001cf9)](https://medium.com/@josce.james7/increase-your-apps-performance-with-angular-pipes-4690ff001cf9" \t "_blank)

[medium.com](https://medium.com/@josce.james7/increase-your-apps-performance-with-angular-pipes-4690ff001cf9" \t "_blank)

* **Avoid unnecessary calculations:** Optimize pipe logic to minimize computations.
* **Caching:** Store intermediate results to avoid redundant calculations.
* **Change detection strategy:** Use OnPush change detection strategy for components with pure pipes to improve performance.
* **Profiling:** Use performance profiling tools to identify performance bottlenecks.

**Example: Impure Pipe with Caching**

TypeScript

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

@Pipe({ name: 'expensive', pure: false })

export class ExpensivePipe implements PipeTransform {

private cache: Map<any, any> = new Map();

transform(value: any, ...args: any[]): any {

if (this.cache.has(value)) {

return this.cache.get(value);

}

// Expensive calculation

const result = /\* ... \*/;

this.cache.set(value, result);

return result;

}

}

Use code [with caution.](/faq#coding)

**Key Considerations:**

* Use pure pipes whenever possible for optimal performance.

[1. What are the best practices for using Angular Pipes? | Reintech media](https://reintech.io/blog/best-practices-for-using-angular-pipes-70357" \t "_blank)

[[Source icon](https://reintech.io/blog/best-practices-for-using-angular-pipes-70357)](https://reintech.io/blog/best-practices-for-using-angular-pipes-70357" \t "_blank)

[reintech.io](https://reintech.io/blog/best-practices-for-using-angular-pipes-70357" \t "_blank)

* Create impure pipes only when necessary and optimize them carefully.
* Consider using OnPush change detection strategy for components with pure pipes.
* Profile your application to identify performance issues and optimize accordingly.

By understanding the nuances of pure and impure pipes, you can create efficient and performant Angular applications.

**Would you like to explore a specific use case or delve deeper into any of these concepts?**

Sources and related content

[medium.com](https://medium.com/@softwaretechsolution/angular-pipes-and-custom-pipe-8515b3bde7d0" \t "_blank)

[medium.com](https://medium.com/@softwaretechsolution/angular-pipes-and-custom-pipe-8515b3bde7d0" \t "_blank)

­