Introdução aos Pipes no Angular

Os **pipes** no Angular são ferramentas que permitem transformar os dados diretamente nos templates. Eles são úteis para formatar strings, números, datas ou qualquer outro dado antes de exibi-los na interface do usuário, sem modificar a fonte original dos dados.

Uso Básico de Pipes

No Angular, você aplica um pipe utilizando o operador `|` no template:

{{ dado | nomeDoPipe }}

- `dado`: O valor a ser transformado.
- `nomeDoPipe`: O pipe que será aplicado.

Pipes Comuns no Angular

1. UpperCase e LowerCase

- Transforma o texto em letras maiúsculas ou minúsculas.

Exemplo:

{{ 'angular pipes' | uppercase }} <!-- Resultado: ANGULAR PIPES -->
{{ 'ANGULAR PIPES' | lowercase }} <!-- Resultado: angular pipes -->

---

2. Date

- Formata uma data.

Exemplo:

{{ today | date }} <!-- Resultado: 1/11/2025 -->

{{ today | date: 'fullDate' }} <!-- Resultado: Saturday, January 11, 2025 -->
{{ today | date: 'dd/MM/yyyy' }} <!-- Resultado: 11/01/2025 -->

Componente:

today = new Date();

---

3. Currency

- Formata números como valores monetários.

Exemplo:

{{ 12345.678 | currency }} <!-- Resultado: $12,345.68 -->
{{ 12345.678 | currency: 'EUR' }} <!-- Resultado: €12,345.68 -->
{{ 12345.678 | currency: 'BRL':'symbol':'1.2-2' }} <!-- Resultado: R$12.345,68 -->

---

4. Decimal

- Formata números para um número específico de casas decimais.

Exemplo:

{{ 1234.5678 | number: '1.2-2' }} <!-- Resultado: 1,234.57 -->
{{ 1234 | number: '3.0-0' }} <!-- Resultado: 001,234 -->

---

5. Percent

- Exibe números como porcentagens.

Exemplo:

{{ 0.25 | percent }} <!-- Resultado: 25% -->

{{ 0.25 | percent: '1.0-0' }} <!-- Resultado: 25% -->

---

6. Slice

- Retorna uma parte de um array ou string.

Exemplo:

{{ 'Angular Pipes' | slice: 0:7 }} <!-- Resultado: Angular -->
{{ [1, 2, 3, 4, 5] | slice: 1:4 }} <!-- Resultado: [2, 3, 4] -->

---

7. Json

- Formata objetos em JSON legível.

Exemplo:

<pre>{{ {name: 'Angular', version: 14} | json }}</pre>
<!-- Resultado:
{
  "name": "Angular",
  "version": 14
}
-->

---

8. Async

- Trabalha com observables e promises.

Exemplo:

{{ observableData | async }}

Componente:

observableData = of('Dados carregados');

Pipes com Argumentos

Alguns pipes, como `date` e `currency`, aceitam argumentos para personalizar sua funcionalidade. Por exemplo:

{{ today | date: 'shortTime' }} <!-- Mostra apenas a hora -->
{{ 12345.678 | currency: 'USD': 'symbol-narrow' }} <!-- Exibe o cifrão estreito -->

Criando Pipes Personalizados

Você pode criar pipes personalizados para atender às suas necessidades específicas.

Passo 1: Criar o Pipe

ng generate pipe nomeDoPipe

Código Exemplo:

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

@Pipe({
  name: 'reverso'
})
export class ReversoPipe implements PipeTransform {
  transform(value: string): string {
    return value.split('').reverse().join('');
  }
}
```

Passo 2: Usar no Template

{{ 'Angular' | reverso }} <!-- Resultado: ralugnA -->

Chaining Pipes

Você pode aplicar vários pipes em sequência.

Exemplo:

{{ 'angular pipes' | uppercase | slice: 0:7 }} <!-- Resultado: ANGULAR -->

Resumo do que é possível fazer com Pipes

- **Transformar dados diretamente no template.**
- **Formatar texto, números, datas, objetos e arrays.**
- **Criar pipes personalizados para casos específicos.**
- **Combinar vários pipes para transformações mais complexas.**

Os pipes são extremamente poderosos no Angular e ajudam a manter o código do template mais limpo, movendo a lógica de transformação para um lugar centralizado.