

## **Diretiva com Seletor de Elemento: Modificando Textos no Angular**

As **diretivas com seletor de elemento** no Angular permitem modificar o comportamento e o estilo de elementos HTML de forma personalizada. Aqui, vamos explorar como criar uma diretiva que altera textos e exemplos de uso.

---

### **Criando a Diretiva para Modificar Textos**

Abaixo, vamos criar uma diretiva chamada `TextModifierDirective`, que permite modificar dinamicamente o conteúdo de um elemento.

#### **Código da Diretiva**

```
import { Directive, ElementRef, Input, OnInit } from '@angular/core';

@Directive({
  selector: 'app-Text-Modifier' // Seletor de elemento personalizado
})
export class TextModifierDirective implements OnInit {

  @Input() newText: string = ''; // Texto que será substituído
  @Input() textColor: string = 'black'; // Cor do texto
  @Input() textSize: string = '16px'; // Tamanho da fonte

  constructor(private el: ElementRef) {} // ElementRef acessa o elemento HTML

  ngOnInit() {
    // Se um novo texto for fornecido, substitui o conteúdo original
    if (this.newText) {
      this.el.nativeElement.innerText = this.newText;
    }

    // Aplica cor e tamanho da fonte diretamente no estilo do elemento
    this.el.nativeElement.style.color = this.textColor;
    this.el.nativeElement.style.fontSize = this.textSize;
  }
}
```

}

---

## Como Usar no HTML

Agora que criamos a diretiva, podemos usá-la em qualquer lugar da aplicação como um **elemento HTML personalizado**.

### ◆ Exemplo 1: Substituindo o Texto

```
<app-Text-Modifier newText="Novo Texto Exibido"></app-Text-Modifier>
```

#### ◆ O que acontece?

✓ O texto do elemento `<app-Text-Modifier>` será substituído por `"Novo Texto Exibido"`.

---

### ◆ Exemplo 2: Alterando Cor e Tamanho do Texto

```
<app-Text-Modifier newText="Texto Azul e Grande" textColor="blue"
textSize="24px"></app-Text-Modifier>
```

#### ◆ O que acontece?

✓ O texto será `"Texto Azul e Grande"`, com **cor azul** e **tamanho de 24px**.

---

### ◆ Exemplo 3: Mantendo o Texto Original e Apenas Estilizando

```
<app-Text-Modifier textColor="green" textSize="30px">Texto
Original</app-Text-Modifier>
```

#### ◆ O que acontece?

✓ O texto original será mantido, mas agora estará **verde** e com **tamanho de 30px**.

---



## O que mais podemos fazer com essa diretiva?

- ♦ **Adicionar efeitos de animação:** Podemos incluir classes CSS para criar animações.
  - ♦ **Aplicar estilos personalizados:** Além de cor e tamanho, podemos modificar `font-family`, `background-color`, `letter-spacing`, etc.
  - ♦ **Alterar automaticamente o conteúdo com base em regras:** Por exemplo, exibir `"Texto Padrão"` caso nenhum texto seja fornecido.
- 



## Expansão: Adicionando Estilos Dinâmicos

Podemos estender a diretiva para permitir estilos mais avançados, como deixar o texto em **negrito**, **itálico** ou até **sublinhado**.

```
@Input() fontWeight: string = 'normal'; // Peso da fonte (negrito, normal, etc.)
@Input() fontStyle: string = 'normal'; // Estilo da fonte (itálico, normal, etc.)
@Input() textDecoration: string = 'none'; // Decoração do texto (sublinhado, etc.)
```

```
ngOnInit() {
  this.el.nativeElement.style.fontWeight = this.fontWeight;
  this.el.nativeElement.style.fontStyle = this.fontStyle;
  this.el.nativeElement.style.textDecoration = this.textDecoration;
}
```

### ♦ Exemplo de Uso

```
<app-Text-Modifier
  newText="Texto Negrito e Sublinhado"
  fontWeight="bold"
  textDecoration="underline"
></app-Text-Modifier>
```



### O que acontece?

- ♦ O texto será **negrito** e **sublinhado**.
-

## Conclusão

Criar **diretivas com seletor de elemento** é uma forma poderosa de modificar textos dinamicamente no Angular. Podemos:

- ✓ Alterar conteúdo do elemento.
- ✓ Aplicar estilos personalizados.
- ✓ Criar experiências dinâmicas sem precisar modificar diretamente o HTML.

Essa abordagem ajuda a criar **componentes reutilizáveis e organizados**, melhorando a experiência do usuário e o desenvolvimento do código! 🚀