

A diretiva de atributo **NgStyle** no Angular permite aplicar estilos dinamicamente a um elemento HTML. É útil quando os estilos precisam ser calculados ou mudados com base em variáveis, estados ou condições da aplicação. Abaixo, segue uma explicação detalhada com exemplos para explorar todas as possibilidades do **NgStyle**.

Exemplo Básico

HTML:

```
<div [ngStyle]="{'color': textColor, 'font-size': fontSize + 'px'}">  
  Este texto muda de cor e tamanho dinamicamente.  
</div>
```

TypeScript:

```
@Component({  
  selector: 'app-ngstyle-example',  
  templateUrl: './app-ngstyle-example.component.html',  
  styleUrls: ['./app-ngstyle-example.component.scss']  
})  
export class NgStyleExampleComponent {  
  textColor: string = 'blue';  
  fontSize: number = 16;  
}
```

O que acontece:

- O texto dentro do **div** será azul e terá tamanho de 16px.
 - Alterar **textColor** ou **fontSize** no componente atualizará automaticamente o estilo do **div**.
-

Exemplo com Condições

HTML:

```
<div [ngStyle]="{ 'background-color': isActive ? 'green' : 'red', 'color': 'white' }">
  {{ isActive ? 'Ativo' : 'Inativo' }}
</div>
<button (click)="toggleActive()">Alterar Status</button>
```

TypeScript:

```
export class NgStyleExampleComponent {
  isActive: boolean = true;

  toggleActive() {
    this.isActive = !this.isActive;
  }
}
```

O que acontece:

- O `div` terá fundo verde e o texto "Ativo" se `isActive` for `true`.
 - Caso contrário, o fundo será vermelho e o texto será "Inativo".
 - O botão altera o estado de `isActive`, atualizando o estilo.
-

Exemplo com Objeto de Estilos

HTML:

```
<div [ngStyle]="styleObject">
  Texto com estilos dinâmicos usando objeto.
</div>
```

TypeScript:

```
export class NgStyleExampleComponent {
  styleObject = {
    'font-weight': 'bold',
    'border': '1px solid black',
    'padding': '10px',
  }
}
```

```
'text-align': 'center',  
'color': 'blue'  
};  
}
```

O que acontece:

- Os estilos no objeto `styleObject` são aplicados diretamente ao `div`.
 - Alterar qualquer propriedade dentro do `styleObject` refletirá no estilo do elemento.
-

Exemplo com Estilos Dependentes de Função

HTML:

```
<div [ngStyle]="getDynamicStyles()">  
  Estilos aplicados via função.  
</div>
```

TypeScript:

```
export class NgStyleExampleComponent {  
  condition: boolean = true;  
  
  getDynamicStyles() {  
    return {  
      'background-color': this.condition ? 'yellow' : 'gray',  
      'font-size': this.condition ? '20px' : '14px',  
      'margin': this.condition ? '20px' : '5px',  
    };  
  }  
}
```

O que acontece:

- A função `getDynamicStyles()` retorna um objeto de estilos baseado no estado da variável `condition`.

- Alterar `condition` resultará em uma atualização dinâmica dos estilos.
-

Exemplo com Interpolação no Objeto

HTML:

```
<div [ngStyle]="{  
  'border': '2px solid ' + borderColor,  
  'padding.px': padding  
}">  
  Texto com borda dinâmica.  
</div>
```

TypeScript:

```
export class NgStyleExampleComponent {  
  borderColor: string = 'purple';  
  padding: number = 10;  
}
```

O que acontece:

- A cor da borda é controlada pela variável `borderColor`.
 - O preenchimento (padding) é controlado pela variável `padding`.
-

Exemplo de Combinação com Eventos

HTML:

```
<div [ngStyle]="{  
  'color': hover ? 'orange' : 'black',  
  'font-style': hover ? 'italic' : 'normal'  
}"  
(mouseenter)="onMouseEnter()"  
(mouseleave)="onMouseLeave()">  
  Passe o mouse sobre este texto.
```

</div>

TypeScript:

```
export class NgStyleExampleComponent {  
  hover: boolean = false;  
  
  onMouseEnter() {  
    this.hover = true;  
  }  
  
  onMouseLeave() {  
    this.hover = false;  
  }  
}
```

O que acontece:

- Quando o mouse entra no elemento, o texto muda para laranja e itálico.
 - Quando o mouse sai, o texto volta para preto e estilo normal.
-

Exemplo com Animação de Cores

HTML:

```
<div [ngStyle]="{'background-color': backgroundColor}">  
  Cor de fundo muda automaticamente.  
</div>
```

TypeScript:

```
export class NgStyleExampleComponent {  
  backgroundColor: string = 'blue';  
  
  ngOnInit() {  
    setInterval(() => {  
      this.backgroundColor = this.backgroundColor === 'blue' ? 'green' : 'blue';  
    }, 1000);  
  }  
}
```

```
}  
}
```

O que acontece:

- A cor de fundo alterna entre azul e verde a cada segundo usando `setInterval`.

Usos Avançados

1. **Misturar com `ngClass`:** Combine `ngStyle` com `ngClass` para definir estilos e classes dinamicamente.
2. **Aplicar em Componentes Customizados:** `NgStyle` pode ser usado em componentes customizados para alterar estilos baseados em `@Input`.
3. **Interatividade:** Use `NgStyle` em elementos interativos como botões, inputs e sliders para feedback visual.

Esses exemplos cobrem desde o básico até o avançado do uso do `NgStyle`. Ele é ideal para aplicações que requerem estilos dinâmicos baseados em lógica de negócios ou interação do usuário.