

O **CSS Class Binding** no Angular é uma técnica que permite adicionar ou remover classes CSS dinamicamente a elementos HTML com base nos valores das propriedades de um componente. Ele fornece controle direto sobre as classes aplicadas, possibilitando mudanças no estilo da interface de forma dinâmica e responsiva.

---

## Como usar CSS Class Binding

Existem três formas principais de utilizar o CSS Class Binding:

1. **Associar uma classe única** com base em uma condição booleana.
  2. **Vincular várias classes dinamicamente** usando objetos.
  3. **Misturar CSS Class Binding com outras classes estáticas.**
- 

## Exemplos

### 1. Classe Única com Condição

```
<button [class.active]="isActive">Botão</button>
```

```
isActive = true; // Define se a classe 'active' será aplicada
```

- **O que acontece?** A classe **active** será aplicada ao botão somente se **isActive** for **true**.

### 2. Várias Classes Dinâmicas com Objeto

```
<button [class]="{'btn-primary': isPrimary, 'btn-disabled': isDisabled}'">  
  Botão Dinâmico  
</button>
```

```
isPrimary = true; // Controla a classe 'btn-primary'  
isDisabled = false; // Controla a classe 'btn-disabled'
```

- **O que acontece?**
  - A classe **btn-primary** será aplicada se **isPrimary** for **true**.

- A classe `btn-disabled` será aplicada se `isDisabled` for `true`.

### 3. Classe Estática Combinada com CSS Class Binding

```
<button class="btn-base" [class.active]="isActive">Botão Combinado</button>
```

```
isActive = false;
```

- **O que acontece?**

- A classe `btn-base` será aplicada sempre.
- A classe `active` será adicionada somente se `isActive` for `true`.

### 4. Alterar Classes Dinamicamente com Clique

```
<button  
  [class.active]="isActive"  
  [class.inactive]="!isActive"  
  (click)="toggleActive()">  
  Clique Aqui  
</button>
```

```
isActive = false;
```

```
toggleActive() {  
  this.isActive = !this.isActive; // Altera o valor de isActive  
}
```

- **O que acontece?**

- Ao clicar, a classe do botão alterna entre `active` e `inactive`.

---

## Aplicação com Estilos

### HTML:

```
<button [class.red-button]="isRed" [class.green-button]="!isRed"  
(click)="toggleColor()">  
  Botão Dinâmico  
</button>
```

## CSS:

```
.red-button {  
  background-color: red;  
  color: white;  
}  
  
.green-button {  
  background-color: green;  
  color: white;  
}
```

## TypeScript:

```
isRed = true;  
  
toggleColor() {  
  this.isRed = !this.isRed; // Alterna entre as classes  
}
```

---

## CSS Classes com Diretivas ngClass

O Angular oferece a diretiva `ngClass` para adicionar múltiplas classes com maior flexibilidade.

### Exemplo com ngClass

```
<div [ngClass]="{'alert': isAlert, 'alert-success': isSuccess, 'alert-danger':  
!isSuccess }">  
  Mensagem Dinâmica  
</div>
```

```
isAlert = true;  
isSuccess = true;
```

- **O que acontece?**
  - A classe `alert` será sempre aplicada.

- A classe `alert-success` será aplicada se `isSuccess` for `true`.
  - A classe `alert-danger` será aplicada se `isSuccess` for `false`.
- 

## Misturando Estilos Dinâmicos e Fixos

```
<div class="card" [class.selected]="isSelected"
[class.highlight]="isHighlighted">
  Conteúdo do Card
</div>
```

```
isSelected = true;
isHighlighted = false;
```

- **Resultado:** O card terá a classe `selected` aplicada, mas não a classe `highlight`.
- 

## Quando Usar CSS Class Binding

- Quando os estilos dependem de variáveis ou condições dinâmicas.
- Para alternar entre estados (ativo/inativo, selecionado/desmarcado, etc.).
- Para simplificar a lógica de estilo sem manipular diretamente o DOM.

Com **CSS Class Binding**, você obtém controle dinâmico sobre a aparência de seus elementos sem precisar acessar o DOM diretamente, tornando o código mais limpo e eficiente.