Utilização do Seletor `:host` no Angular

O seletor `:host` no Angular é utilizado em arquivos de estilo (CSS ou SCSS) de componentes para aplicar estilos diretamente ao **elemento raiz** do componente, ou seja, o host DOM element que contém o componente.

---

1. O que é o `:host`?

O `:host` permite estilizar o elemento que encapsula o componente. Isso é útil para aplicar estilos no contêiner raiz do componente sem afetar elementos filhos ou encapsulados.

Por exemplo, se um componente tem o seguinte seletor:

```
@Component({
  selector: 'app-card',
  template: `<div class="content">Hello, world!</div>`,
  styleUrls: ['./card.component.scss']
})
```

O `:host` estiliza `<app-card>` no DOM.

---

2. Sintaxe

A sintaxe básica do `:host` é:

```
:host {
  /* estilos para o elemento raiz do componente */
}
```

---

3. Exemplos de Uso

3.1 Estilizando o Elemento Raiz do Componente

Se você quiser aplicar um fundo ao elemento `<app-card>`:

```
:host {
```

```
  background-color: #f9f9f9;
  display: block;
  padding: 16px;
  border-radius: 8px;
}
```

3.2 Aplicando Estilos com Classes no Elemento Host

Você pode estilizar o host com base em uma classe adicionada a ele no DOM:

```
:host(.highlight) {
  background-color: yellow;
}
```

HTML:

```
<app-card class="highlight"></app-card>
```

3.3 Usando Estados Dinâmicos (Hover, Focus)

Você pode aplicar estilos dinâmicos ao elemento host:

```
:host(:hover) {
  border: 2px solid #3498db;
  cursor: pointer;
}
```

3.4 Combinando `:host` com `::ng-deep`

Se você quiser aplicar estilos ao host e aos elementos dentro dele:

```
:host {
  display: block;
  padding: 20px;
}

:host ::ng-deep .child-element {
  color: red;
}
```

3.5 Modificando o Layout com Base em Propriedades Dinâmicas

Se o componente Angular estiver alterando classes dinamicamente com `[ngClass]` ou `[class]`, você pode estilizar essas classes no `:host`:

```
:host(.error) {
  border: 1px solid red;
  background-color: #ffe6e6;
}
```

TS:

```
@Component({
  selector: 'app-card',
  template: `<div>Erro no sistema!</div>`,
  styleUrls: ['./card.component.scss'],
  host: {
    '[class.error]': 'hasError'
  }
})
export class CardComponent {
  hasError = true;
}
```

---

4. Utilizando `:host-context`

O `:host-context` é uma extensão poderosa que permite aplicar estilos ao host **com base no contexto do DOM** onde ele está inserido.

Exemplo: Se você quiser aplicar um estilo ao host somente quando ele estiver dentro de um contêiner com a classe `.dark-theme`:

```
:host-context(.dark-theme) {
  background-color: #333;
  color: #fff;
}
```

HTML:

```
<div class="dark-theme">
  <app-card></app-card>
</div>
```

---

5. Erros Comuns e Soluções

1. **Estilo Não Aplicado:**
    - **Causa:** O estilo no `:host` não está associado ao elemento raiz.
    - **Solução:** Certifique-se de que o seletor do componente está correto e o estilo está encapsulado no arquivo de estilo do componente.
2. **Encapsulamento de Estilo:**
    - **Causa:** Alterações globais no elemento host podem não funcionar devido ao encapsulamento.
    - **Solução:** Use `ViewEncapsulation.None` se quiser desativar o encapsulamento.

---

6. Quando Usar o `:host`

- Para aplicar estilos diretamente no elemento que encapsula o componente.
- Para personalizar estilos de componentes baseados em suas classes ou estados.
- Para controlar o layout do componente com base em seu contexto no DOM.

O seletor `:host` é uma ferramenta essencial para criar componentes Angular reutilizáveis, mantendo o isolamento de estilo e o controle sobre o design e layout de cada componente no aplicativo.