

Exemplos de uso com explicação clara e precisa do vídeo "Curso Angular #15: Criando Diretivas Personalizadas"

O vídeo "Curso Angular #15: Criando Diretivas Personalizadas" da Loiane Groner apresenta diversas técnicas para criar diretivas personalizadas em Angular. Abaixo, detalhamos alguns dos principais exemplos de uso demonstrados no vídeo, com explicações claras e precisas:

1. Criar uma diretiva personalizada para destacar texto em negrito:

Cenário: Imagine que você precisa de uma maneira fácil de destacar texto em negrito em qualquer parte do seu aplicativo Angular.

Solução: Para criar essa diretiva, siga estes passos:

1. **Crie um arquivo para a diretiva:** Crie um novo arquivo TypeScript, por exemplo, `destaque-texto.directive.ts`.
2. **Importe as classes necessárias:** Importe as classes `Directive`, `ElementRef` e `HostListener` do Angular.

TypeScript

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

3. **Defina o seletor da diretiva:** Utilize o decorador `@Directive` para definir o seletor da diretiva, que indica o elemento HTML ao qual ela será aplicada. Neste caso, o seletor será `[appDestaqueTexto]`.

TypeScript

```
@Directive({  
  selector: '[appDestaqueTexto]'  
})
```

4. **Implemente a lógica da diretiva:** No construtor da diretiva, injete o `ElementRef` para ter acesso ao elemento HTML ao qual a diretiva está sendo aplicada.

TypeScript

```
constructor(private elRef: ElementRef) { }
```

5. **Crie um método para aplicar o estilo negrito:** Crie um método que aplique o estilo CSS `font-weight: bold` ao elemento HTML.

TypeScript

```
private aplicarNegrito() {  
  this.elRef.nativeElement.style.fontWeight = 'bold';  
}
```

6. **Escute o evento de entrada da diretiva:** Utilize o decorador `@HostListener` para escutar o evento `mouseenter` do elemento HTML. Quando o mouse passar sobre o elemento, chame o método `aplicarNegrito` para aplicar o estilo negrito.

TypeScript

```
@HostListener('mouseenter')  
onMouseEnter() {  
  this.aplicarNegrito();  
}
```

7. **Remova o estilo negrito quando o mouse sair do elemento:** Utilize o decorador `@HostListener` para escutar o evento `mouseleave` do elemento HTML. Quando o mouse sair do elemento, remova o estilo negrito definindo `font-weight` como `normal`.

TypeScript

```
@HostListener('mouseleave')  
onMouseLeave() {  
  this.elRef.nativeElement.style.fontWeight = 'normal';  
}
```

8. **Utilize a diretiva em seu HTML:** No HTML, aplique a diretiva `appDestaqueTexto` ao elemento que você deseja destacar em negrito.

HTML

```
<p appDestaqueTexto>Este texto será destacado em negrito quando o mouse passar sobre ele.</p>
```

2. **Criar uma diretiva personalizada para adicionar uma classe CSS a um elemento:**

Cenário: Suponha que você precise de uma maneira fácil de adicionar uma classe CSS específica a um elemento HTML em seu aplicativo Angular.

Solução: Para criar essa diretiva, siga estes passos:

8. **Crie um arquivo para a diretiva:** Crie um novo arquivo TypeScript, por exemplo, `adicionar-classe.directive.ts`.

9. **Importe as classes necessárias:** Importe as classes `Directive`, `ElementRef` e `Input` do Angular.

TypeScript

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

10. **Defina o seletor da diretiva:** Utilize o decorador `@Directive` para definir o seletor da diretiva, que indica o elemento HTML ao qual ela será aplicada. Neste caso, o seletor será `[appAdicionarClasse]`.

TypeScript

```
@Directive({  
  selector: '[appAdicionarClasse]'  
})
```

11. **Defina uma propriedade de entrada para a classe CSS:** Utilize o decorador `@Input` para criar uma propriedade de entrada chamada `classe`, que receberá o nome da classe CSS que você deseja adicionar ao elemento.

TypeScript

```
@Input() classe: string;
```

12. **Implemente a lógica da diretiva:** No construtor da diretiva, injete o `ElementRef` para ter acesso ao elemento HTML ao qual a diretiva está sendo aplicada.

TypeScript

```
constructor(private elRef: ElementRef) { }
```

13. ****Adicione a**