Destaque Personalizado com Cores Personalizadas no Angular 2: Exemplos Práticos

O Angular oferece a flexibilidade de criar diretivas personalizadas para estender as funcionalidades dos seus componentes. Neste contexto, vamos explorar como criar uma diretiva de destaque personalizável com cores customizadas.

1. Definindo Cores Padrão:

A diretiva terá duas propriedades de entrada: `defaultColor` e `highlightColor`. Por padrão, `defaultColor` será branco e `highlightColor` será amarelo. Você pode definir essas cores no código da diretiva ou passá-las como atributos para a diretiva.

Exemplo:

HTML

```
<p highlight defaultColor="gray" highlightColor="red">Texto com destaque personalizado</p>
```

2. Usando `ngOnInit` para Inicializar a Cor de Fundo:

O método `ngOnInit` é chamado quando a diretiva é inicializada. Você pode usar este método para definir a cor de fundo padrão do elemento host.

Exemplo:

TypeScript

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

@Component({
  selector: 'highlight',
  template: '<ng-content></ng-content>',
})
export class HighlightDirective {
  @Input() defaultColor: string = 'white';
  @Input() highlightColor: string = 'yellow';
```

```
  ngOnInit() {
    this.element.style.backgroundColor = this.defaultColor;
  }

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

3. Usando `@HostBinding` para Atualizar a Cor de Fundo Dinamicamente:

O decorador `@HostBinding` permite vincular propriedades da diretiva a propriedades do elemento host. Você pode usar isso para atualizar a cor de fundo do elemento host quando o mouse passar sobre ele.

Exemplo:

TypeScript

```
import { Component, Input, HostBinding } from '@angular/core';

@Component({
  selector: 'highlight',
  template: '<ng-content></ng-content>',
})
export class HighlightDirective {
  @Input() defaultColor: string = 'white';
  @Input() highlightColor: string = 'yellow';

  @HostBinding('style.backgroundColor')
  backgroundColor: string;

  ngOnInit() {
    this.backgroundColor = this.defaultColor;
  }

  onMouseEnter() {
    this.backgroundColor = this.highlightColor;
  }

  onMouseLeave() {
    this.backgroundColor = this.defaultColor;
  }

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

```
}
```

4. Passando Cores Personalizadas como Atributos:

Você pode passar cores personalizadas para a diretiva usando os atributos `defaultColor` e `highlightColor`.

Exemplo:

```
HTML
<p highlight defaultColor="gray" highlightColor="red">Texto com destaque personalizado</p>
```

5. Usando `ngStyle` para Estilos Dinâmicos:

A diretiva `ngStyle` permite aplicar estilos dinâmicos a um elemento. Você pode usar isso para aplicar a cor de fundo e outras propriedades de estilo ao elemento host quando o mouse passar sobre ele.

Exemplo:

```
HTML
<p [ngStyle]="{ 'background-color': isHovered ? highlightColor : defaultColor }">Texto com destaque personalizado</p>
```

6. Criando uma Diretiva Reutilizável:

Você pode encapsular a lógica da diretiva personalizada em uma classe reutilizável. Isso facilita a reutilização da diretiva em diferentes partes do seu aplicativo.

Exemplo:

```
TypeScript
import { Component, Input, HostBinding, Directive } from '@angular/core';

@Directive({
```

```
  selector: '[highlight]',
})
export class HighlightDirective {
  @Input() defaultColor: string = 'white';
  @Input() highlightColor: string = 'yellow';

  @HostBinding('style.backgroundColor')
  backgroundColor: string;

  ngOnInit() {
    this.backgroundColor = this.defaultColor;
  }

  onMouseEnter() {
    this.backgroundColor = this.highlightColor;
  }

  onMouseLeave() {
    this.backgroundColor = this.defaultColor;
  }
}
```

7. Usando a Diretiva em Componentes Diferentes:

Você pode usar a diretiva personalizada em vários componentes dentro do seu aplicativo Angular. Basta importar a diretiva e adicioná-la ao seletor do componente.

Exemplo:

HTML

```
<p highlight defaultColor="gray" highlightColor="red">Texto com destaque personalizado no componente app</p>

<p highlight defaultColor="blue" highlightColor="green">Texto com destaque personalizado no outro componente</p>
```