

## 1 Introdução

O Angular possui diversos mecanismos que possibilitam a especificação de estilos para componentes usando CSS. Neste material estudaremos os principais.

## 2 Desenvolvimento

**2.1 (Novo projeto)** Comece abrindo um novo terminal e navegando até o seu workspace, ou seja, a pasta que você está usando para abrigar os seus projetos Angular. Tome o cuidado de não acessar a pasta de nenhum projeto já existente. Use os comandos a seguir para criar um novo projeto Angular, navegar até o seu diretório e abrir uma instância do VS Code vinculada a ele.

```
ng new nome-do-projeto
cd nome-do-projeto
code .
```

- Quando perguntado, escolha não adicionar um módulo para roteamento e use o modelo CSS simples.

- Coloque o servidor de testes em execução com

```
ng serve --open
```

**2.2 (Novo componente)** Crie um novo componente chamado formulario com o seguinte comando

```
ng g c formulario --skipTests
```

**2.3 (Código para o template)** Vamos criar uma página HTML simples para então aplicar a ela algumas configurações básicas de CSS. Apague o conteúdo do arquivo **formulario.component.html** e adicione a ele o conteúdo da Listagem 2.3.1.

Listagem 2.3.1

```
<div class="principal">
  <article class="conteudo">
    <header class="titulo">
      Aplicando CSS com Angular
    </header>
    <div class="corpo">
      Há diferentes formas para se aplicar estilos com mecanismos do Angular
    </div>
  </article>
</div>
```

**2.4 (Aplicando CSS no arquivo vinculado ao componente)** O objeto JSON que especificamos na anotação `@Component` possui uma propriedade esperada chamada **styleUrls**. Caso especificada, ela deve ser associada a um vetor JSON cujos elementos são nomes de arquivos css. Por padrão, o componente criado pelo Angular CLI inclui essa propriedade e a vincula a um vetor cujo único elemento é o nome do arquivo css que fica na mesma pasta do componente. Abra, portanto, o arquivo **formulario.component.css** e adicione a ele o conteúdo da Listagem 2.4.1.

#### Listagem 2.4.1

```
.principal {
  width: 1280px;
  margin: auto;
  text-align: center;
}

.conteudo {
  padding: 12px;
  margin-top: 12px;
  border: 1px solid #CCC;
  border-radius: 8px;
  display: inline-block;
}

.titulo {
  border-bottom: 1px solid black;
  padding: 8px;
  margin-bottom: 12px;
  background-color: #EEE;
  border-radius: 8px 8px 0 0;
}
```

- Note que esse conteúdo CSS se aplica somente ao componente que referencia o arquivo em seus metadados.

- Para ver o resultado, apague todo o conteúdo do arquivo **app.component.html** e adicione a tag (especificada na propriedade **selector**) do componente. Por padrão, ela se chama **app-formulario**.

**2.5 (A propriedade styles)** O objeto JSON especificado nos metadados do componente (anotação `@Component`) possui também uma propriedade chamada `styles`. Com ela, podemos especificar estilos sem a necessidade de utilizar outros arquivos. Veja a Listagem 2.5.1.

**Dica:** Use a crase para definir a string, assim será possível pular linhas.

**Nota:** As propriedades **styles** e **styleUrls** podem ser especificadas simultaneamente. Têm precedência as regras dos arquivos associados a `styleUrls`.

### Listagem 2.5.1

```
@Component({
  selector: 'app-formulario',
  templateUrl: './formulario.component.html',
  styles: [
    .titulo {
      font-size: 1.5rem;
    }
    .corpo{
      padding: 8px;
      font-size: 1.1rem;
    }
  ],
  styleUrls: ['./formulario.component.css'],
})
```

**2.6 (A diretiva ngStyle)** A diretiva ngStyle é um tipo de diretiva que permite alterar os atributos de um elemento. Para utilizá-la, basta fazer um property binding e especificar um objeto JSON com as propriedades desejadas. Adicione o botão da Listagem 2.6.1 para testá-la.

### Listagem 2.6.1

```
<div class="principal">
  <article class="conteudo">
    <header class="titulo">
      Aplicando CSS com Angular
    </header>
    <div class="corpo">
      Há diferentes formas para se aplicar estilos com mecanismos do Angular
    </div>
  </article>
  <div [ngStyle]="{ 'margin-top': '4px' }">
    <button [ngStyle]="{ 'padding': '8px', 'width': '20%' }">OK</button>
  </div>
</div>
```

- Utilizada desta forma, a diretiva ngStyle não oferece muitas vantagens quando comparada com estilos CSS inline. O fato é que o objeto JSON associado a ela pode ser gerado em tempo de execução. Para ilustrar, implemente o método da Listagem 2.6.2 no arquivo **formulario.component.ts** e faça a sua associação com a diretiva no arquivo **formulario.component.html**.

## Listagem 2.6.2

```
export class FormularioComponent {  
  obterEstilosParaOBotao() {  
    return {  
      backgroundColor: '#eee6ff',  
      padding: '8px',  
      width: '20%',  
      border: 'none',  
      borderRadius: '4px'  
    }  
  }  
}  
  
<div [ngStyle]="{ 'margin-top': '4px' }">  
  <button [ngStyle]="obterEstilosParaOBotao()">OK</button>  
</div>
```

**Nota:** É possível especificar os nomes das propriedades CSS usando o padrão camelCase (sem aspas) ou o padrão kebab-case (com aspas).

**2.7 (A diretiva `ngClass`)** A diretiva **`ngClass`** pode ser associada a uma lista de nomes de classes. Para vê-la em funcionamento, crie a classe da Listagem 2.7.1 no arquivo **formulario.component.html** e aplique-a ao rodapé criado na Listagem 2.7.2, usando a diretiva **`ngClass`**.

## Listagem 2.7.1

```
.rodape {  
  border-top: 1px solid #CCC;  
  border-bottom: 1px solid #CCC;  
  padding: 8px;  
  margin-top: 12px;  
  width: 50%;  
  display: inline-block;  
}
```

### Listagem 2.7.2

```
<div class="principal">
  <article class="conteudo">
    <header class="titulo">
      Aplicando CSS com Angular
    </header>
    <div class="corpo">
      Há diferentes formas para se aplicar estilos com mecanismos do Angular
    </div>
  </article>
  <div [ngStyle]="{ 'margin-top': '4px' }">
    <button [ngStyle]="obterEstilosParaOBotao()">OK</button>
  </div>
  <footer [ngClass]="['rodape']">
    Angular. 2020. Google Inc.
  </footer>
</div>
```

- Nada impede que as classes a serem vinculadas à diretiva ngClass sejam devolvidas por um método, assim como ocorre com a diretiva ngStyle. Veja a Listagem 2.7.3.

### Listagem 2.7.3

```
<footer [ngClass]="obterClassesRodape()">
  Angular. 2020. Google Inc.
</footer>

obterClassesRodape() {
  return ['rodape']
}
```

- A diretiva ngClass também pode ser vinculada a um objeto JSON cujas chaves são nomes de classes e cujos valores associados são valores booleanos que indicam se as respectivas classes devem ser aplicadas ou não. Para ilustrar, vamos fazer um event binding no botão. O método chamado alterará o valor de uma variável declarada no componente, gerando um valor aleatório. Caso o valor seja par, aplicaremos uma classe. Caso seja ímpar, aplicaremos uma segunda classe. No arquivo **formulario.component.ts**, defina o método e a variável da Listagem 2.7.4.

#### Listagem 2.7.4

```
valor;  
gerarValor() {  
  this.valor = Math.round(Math.random() * 100) + 1;  
}
```

- No arquivo **formulario.component.css**, defina as classes da Listagem 2.7.5.

#### Listagem 2.7.5

```
.numero-par {  
  background-color: rgba(0, 0, 255, 0.349);  
}  
  
.numero-impar {  
  background-color: rgba(255, 0, 0, 0.534);  
}
```

- Faça o event binding no botão, como na Listagem 2.7.6.

#### Listagem 2.7.6

```
<button [ngStyle]="obterEstilosParaOBotao()" (click)=gerarValor()>OK</button>
```

- Adicione a div da Listagem 2.7.7 logo acima do botão.

#### Listagem 2.7.7

```
<div class="principal">  
  <article class="conteudo">  
    <header class="titulo">  
      Aplicando CSS com Angular  
    </header>  
    <div class="corpo">  
      Há diferentes formas para se aplicar estilos com mecanismos do Angular  
    </div>  
  </article>  
  <div class="valor">  
    {{valor}}  
  </div>  
  <div [ngStyle]="{ 'margin-top': '4px' }">  
    <button [ngStyle]="obterEstilosParaOBotao()" (click)=gerarValor()>OK</button>  
  </div>  
  <footer [ngClass]="obterClassesRodape()">  
    Angular. 2020. Google Inc.  
  </footer>  
</div>
```

- Defina a classe CSS **valor** como na Listagem 2.7.8.

#### Listagem 2.7.8

```
.valor {  
  margin: 24px;  
  padding: 50px;  
  border: 1px solid #EEE;  
  border-radius: 8px;  
  width: 20%;  
  display: inline-block  
}
```



## ***Referências***

Angular. 2020. Disponível em <<https://angular.io>>. Acesso em maio de 2020.