



Utilizando a primeira diretiva

Transcrição

Temos dois componentes em exibição na página do navegador, cada qual usando seus próprios dados, como gostaríamos. Mas e se tivéssemos cem, duzentas imagens, o que teríamos que fazer? Seria muito trabalhoso. Além disso, sabemos que, quando trabalhamos com uma grande quantidade de dados, no universo da programação, usualmente lidamos com **listas**.

Vamos abrir `app.component.ts` e substituir as propriedades de `AppComponent`, que não estão sendo utilizadas, pelo seguinte trecho:

```
photos = [  
  {  
    url: 'https://upload.wikimedia.org/wikipedia/commons/thu  
    description: 'Leão'  
  },  
  {  
    url: 'https://upload.wikimedia.org/wikipedia/commons/thu  
    description: 'Leoa'  
  }  
];
```

[COPIAR CÓDIGO](#)

Com isso, criamos a propriedade `photos`, em que cada item do *array* é um objeto, com aquelas mesmas informações vistas anteriormente, de `url` e `description` das imagens do leão e da leoa. A partir destes dados listados, queremos criar um `<ap-photo>` para cada dado proveniente do *array*.

Não queremos utilizar o `forEach()`, passando pelos itens e criando um `<ap-photo>` individualmente, fazendo loops. A boa notícia é que o Angular possui um recurso chamado **diretiva**, capaz de alterar o comportamento de um componente já existente. A diretiva `ngFor` receberá `photos`, o mesmo que aparece em `app.component.ts`. Entretanto, a cada passada em que se queira referenciar a este `photos`, usaremos o nome `photo`.

Isso significa que iteramos `photos` e, a cada passada, o valor da imagem correspondente ficará na variável `photo`. Não podemos nos esquecer de declarar a variável usando `let`. E, se `photo` é uma variável, em `url` colocamos `photo.url` e, em `description` usamos `photo.description`.

```
<ap-photo
  *ngFor="let photo of photos"
  url="photo.url"
  description="photo.description"
>
</ap-photo>
```

[COPIAR CÓDIGO](#)

No entanto, desta forma não fazemos *Data binding*, e isto é um problema, pois o Angular entenderá que estamos passando a *string* `photo.url`, e não o valor dessa expressão. Então, o código ficará assim:

```
<ap-photo
  *ngFor="let photo of photos"
  [url]="photo.url"
  [description]="photo.description"
>
</ap-photo>
```

[COPIAR CÓDIGO](#)

Salvaremos o arquivo, voltaremos ao navegador, e teremos as duas imagens sendo exibidas corretamente. Se adicionarmos mais uma imagem no *array*, em `app.component.ts`, uma terceira imagem aparecerá na página. O Angular possui esta diretiva que se encarrega de iterar uma lista e repetir o componente para nós. Isso é muito interessante, já que, em vez de termos uma lista de imagens fixa, temos uma lista cujas imagens vêm do back end.

Assim, sabemos que a partir do momento em que colocarmos um novo valor em `photos`, o *Data binding* do Angular disparará a renderização da nossa tela, e então teremos acesso a todas as imagens que vêm do back end, com o qual faremos a integração em breve.