APLICAÇÕES INTERATIVAS

ANGULAR
BINDINGS (LIGAÇÕES) DE VIA ÚNICA

Bindings

- Bindings (“ligações” do inglês) são formas de conectar o template de um componente (página HTML) a algum código (propriedade ou método) presente no arquivo de comportamento do mesmo (arquivo TypeScript do componente)

Bindings

- Bindings podem ser feitos entre elementos de texto ou atributos de componentes
- Há diferentes tipos de bindings, utilizados em ocasiões específicas, que estudaremos a seguir

Interpolação

- Interpolação é o tipo de binding que utilizamos anteriormente
- Capaz de resolver expressões de propriedades e métodos do TypeScript do componente
- Usado nos templates dos componentes (HTML)

Interpolação

- Para realizar a interpolação de uma propriedade ou método, declaramos o mesmo entre {{ e }}
- O angular substituirá a ocorrência pelo valor da propriedade ou do retorno do método

Interpolação

- Podem ser utilizados tanto no texto da página (antes ou depois) quanto em atributos HTML e dentro de tags
- É necessário que a propriedade ou o método exista no arquivo .ts do mesmo componente

Interpolação

{{ propriedade }}

Resolução de Interpolações

- As propriedades interpoladas serão resolvidas com base nos valores do objeto da classe “.ts” do componente
- Não é possível utilizar propriedades de outros componentes diretamente (apenas do próprio componente)

Resolução de Interpolações

test.component.ts

```
@Component({
  selector: 'app-test',
  templateUrl: './test.component.html',
  styleUrls: ['./test.component.css']
})
export class TestComponent implements OnInit
 {

  propriedade = 'Teste';

  constructor() { }

  ngOnInit() {

  }

}
```

test.component.html

```
<p>{{propriedade}}</p>
```

Quando o Angular processar e exibir o HTML do componente, o valor de {{propriedade}} será resolvido para “Teste”

```
<p>Teste</p>
```

Formas de Uso de Interpolação

- Uso de interpolação no interior de tags ou junto a outro tipo de texto de conteúdo

{{ propriedade }}

Resultado: {{ propriedade }}

{{ propriedade }}

Valor é {{ propriedade }}

Formas de Uso de Interpolação

- Uso de interpolação em valores de atributos, integral ou parcialmente (junto a mais texto)

```
<img src="{{imageUrl}}">

<img src="images/{{url}}">

<p class="{{value}}">

<p class="p.{{highlightClass}}">
```

Chamada a Métodos

- É possível efetuar a chamada de métodos no interior de interpolações
- O método será substituído pelo valor de seu retorno e é possível fornecer-lhe parâmetros

```
{{ obterValor() }}

<p>{{ somarValores(1, 1) }}</p>
```

Interpolação e Expressões

- Interpolações são capazes de resolver expressões simples, como cálculos e concatenações

```
<p>A soma de 1 + 1 é {{1 + 1}}</p>
```

```
<p>Soma: {{ 1 + 1 + getValue() }}</p>
```

Tipos de Dados de Interpolações

- As interpolações podem ocorrer em quaisquer tipos de dados, basta que o Angular seja capaz de obter uma representação de string do mesmo
- string, number, boolean, Array, etc

Boas Práticas de Interpolações

- Simplicidade
  - As expressões de interpolação devem ser simples e de fácil entendimento
  - É possível escrever expressões bem complexas utilizando interpolações, **mas não é recomendado**
  - Focar em propriedades e métodos

Boas Práticas de Interpolações

- Execução Rápida
  - Não devem ser utilizadas expressões que realizem tarefas pesadas, já que o Angular precisará reprocessar a expressão sempre que algo atualizar
  - Não realizar consultas remotas e a bancos de dados em expressões de interpolação

Exercícios de Interpolação

Crie um componente novo para cada exercício e adicione-o em app-component

1. Crie uma propriedade com seu nome e outra com seu sobrenome. Faça com que ambas apareçam juntas, separadas por um espaço, na página

Exercícios de Interpolação

2. Crie um método que aceite uma string e seja capaz de devolvê-la com todos os caracteres maiúsculos. Chame esse método na página através de interpolação, fornecendo-lhe como parâmetro a palavra “corona”

Exercícios de Interpolação

3. Defina uma propriedade com o nome de classe CSS “petroleo” e:
    1. Crie esta classe no CSS com a fonte verde escuro, negrito, fundo amarelo
    2. Faça com que a classe seja atribuída através de interpolação num h1 com a palavra “Petrobrás”
    3. Utilize a função do exercício anterior para deixar também a palavra em maiúsculas

Binding Unidirecional

- A interpolação é uma forma de ligação unidirecional (*one-way binding*)
- Representa que o valor é modificado em um lado (componente) e o efeito é propagado para o outro (template/HTML)

Binding Unidirecional

- O binding unidirecional, nesse caso, ocorre do componente em direção ao HTML
- Sempre que o componente tiver a propriedade atualizada, o HTML será atualizado pelo Angular

Binding Unidirecional

- Dizemos que o binding via interpolação é um tipo de binding que ocorre em source-to-view (código para a tela)

Property Binding

- Outra forma de realizar binding unidirecional (além da interpolação) é através da utilização de *property binding*
- Permite ligar um atributo de um elemento HTML diretamente a uma propriedade/método no *.ts* do componente

Property Binding

- Para realizar property binding, inserimos colchetes “[ ]” ao redor de um atributo HTML e, em seu valor, apontamos para o nome da propriedade ou do método que deve ser consultado para resolução do atributo
- Se for um método, deve retornar o valor

Property Binding

```
<p [class]="styleClass">Teste</p>
```

Property Binding

.html

```
<p [class]="styleClass">Teste</p>
```

.ts

```
...
export class TestComponent implements OnInit {

  styleClass = 'pHighlight';

  constructor() { }

  ...

}
```

.css

```
.pHighlight {
    color: yellow;
    font-weight: bold;
}
```

Property Binding

- É possível utilizar *property bindings* para controlar o estado de componentes
- Se o valor for alterado no .ts, as alterações serão refletidas automática e instantaneamente no HTML

Property Binding

.html

```
<input [disabled]="disableInput">
```

.ts

```
...
export class TestComponent implements OnInit {

  disableInput = false;

  constructor() { }

  ...

}
```

Property Binding

- *Property binding* também é um tipo de binding que ocorre em source-to-view (do código para a tela)
- Ou seja, as propriedades alteram o HTML conforme são modificadas no .ts

Exercícios de Property Binding

4. Insira três imagens na pasta assets
    a) Crie três tags img capazes de resolver qual imagem mostrar através de property binding (3 propriedades diferentes)
    b) Crie também 3 classes CSS com configurações diferentes para imagens (tamanho, borda, etc) e atribua-as através de property binding às imagens

Binding de Eventos

- O binding de eventos ou “declarações de template” (*template statements*) são utilizados no Angular para permitir que eventos comuns de elementos HTML (clique, pressionamento de teclas, apontar com o mouse, etc) acionem métodos no .ts do componente

Binding de Eventos

- O binding unidirecional, nesse caso, ocorre da página em direção ao Componente
- Dizemos que o binding de eventos é um tipo de binding que ocorre em view-to-source (da tela para o código), o contrário dos anteriores

Binding de Eventos

- Sempre que um evento ocorrer, o HTML acionará o código .ts e este poderá alterar o estado das propriedades do componente
- Essas alterações nas propriedades serão, por consequência, refletidas automaticamente na página

Binding de Eventos

- Para realizar binding de eventos, colocamos o atributo do evento do elemento HTML sem o “on” (click, mouseover, keypress, etc) entre parênteses “( )” e, no valor do atributo, informamos o método que será executado quando o evento ocorrer

Binding de Eventos

- O método deve ser informado como uma chamada (com “( )” no final) e parâmetros podem ser fornecidos
- O método deve estar no código .ts do próprio componente (não é possível chamar métodos de outros componentes diretamente)

Binding de Eventos

```
<button (click)="tratarClick()">Clique aqui!</button>
```

Binding de Eventos

.html

```
<button (click)="tratarClick()">Clique aqui!</button>
```

.ts

```
...
export class TestComponent implements OnInit {

  constructor() { }

  ...

  tratarClick():void {
     alert('Olá, mundo!');
  }
}
```

Binding de Eventos

.html

```
<input [disabled]="disableInput">
<button (click)="tratarClick()">Clique aqui!</button>
```

.ts

```
...
export class TestComponent implements OnInit {

  disableInput = false;

  ...

  tratarClick():void {
    this.disableInput = !this.disableInput;
  }
}
```

Exercícios de Event Binding

5. Crie um componente que contenha uma imagem e um botão. Ao clicar no botão, uma segunda imagem deve ser exibida. Se o botão for novamente clicado, alternar para a imagem original (toggle)

Exercícios de Event Binding

6. Crie 3 botões. Quando clicados, cada um deles deve atribuir um texto específico para uma propriedade. Use a propriedade num texto simples para exibir seu valor conforme os botões são clicados

Exercícios de Event Binding

7. Crie três botões e 3 divs. Ao serem clicados, os botões devem controlar quais classes cada div possui, alternando entre classes em que uma div está visível e as outras duas estão invisíveis, para cada uma das divs (alternando entre divs 1, 2 e 3)


"That's all Folks!"