



Data Binding (Ligação de Dados)

O que é?

Data Binding é o mecanismo que permite que os dados do componente (TypeScript) apareçam na tela (HTML), e que ações do usuário na tela sejam enviadas de volta para o componente.

É como um "fio invisível" que conecta o cérebro (lógica) com o rosto (interface visual) da sua aplicação.



Interpolação – {{ nome }}

O que é? Permite mostrar valores do componente diretamente no HTML.

É caracterizado com chaves duplas no HTML.

Ts:

```
nome = 'João';
```

Html:

```
<p>Olá, {{ nome }}!</p>
```

Resultado:

```
Olá, João!
```



Property Binding – `[src]="imageUrl"`

O que é? Você conecta uma propriedade HTML a uma variável do componente.

Ts:

Html:

```
imageUrl = 'https://exemplo.com/imagem.png';
```

```
<img [src]="imageUrl">
```

Conclusão:

Sem o Angular, você teria que atualizar manualmente essa URL. Com o Angular, ela se atualiza automaticamente se mudar no componente.



Event Binding – (click)="mudarTexto()"

Definição: Aqui você liga um evento do DOM (como clique, digitação, etc.) a uma função no componente.

Ts:

```
mensagem = 'Olá!';

mudarTexto() {
  this.mensagem = 'Você clicou!';
}
```

Html:

```
<p>{{ mensagem }}</p>
<button (click)="mudarTexto()">Clique aqui</button>
```

É como apertar um botão do controle remoto: você aperta, e algo acontece.



Two-way Binding – [(ngModel)]="nome"

Definição: Permite que uma variável seja atualizada automaticamente tanto na tela quanto no código.

Ts:

```
nome = '';
```

Html:

```
<input [(ngModel)]="nome">  
<p>Você digitou: {{ nome }}</p>
```

À medida que você digita, a variável nome no componente muda, e o texto abaixo mostra isso em tempo real.

É como um espelho: o que acontece de um lado, reflete no outro imediatamente.



Por que data binding é importante?

Porque sem o data binding, seria como ter dois mundos separados:
o que o usuário vê e o que o sistema faz.

O data binding une esses dois mundos, fazendo com que a aplicação reaja aos dados e ações do usuário automaticamente.

Dando ao usuário feedback do que suas ações estão fazendo na aplicação web.



Diretivas são instruções especiais no HTML que dizem ao Angular como mostrar ou comportar-se com certos elementos.

Vamos ver os dois tipos principais:

Diretivas Estruturais

Elas mudam a estrutura do DOM, ou seja, adicionam, removem ou repetem elementos.

Diretivas de Atributo

Elas mudam o estilo ou comportamento de um elemento já existente.



Diretivas Estruturais - @If

Serve para: Mostrar ou esconder algo com base em **uma condição**.

Ts:

```
estaLogado : boolean = true;
```

HTML:

```
@if (estaLogado) {  
    <p>Bem Vindo!</p>  
}
```

Se *estaLogado* for **false**, o <p> nem aparece no HTML.

É como acender uma luz só se alguém estiver no quarto.



Diretivas Estruturais - @for

Serve para: Repetir um elemento para **cada item de uma lista**.

Ts:

```
produtos : string[] = [  
    'teclado',  
    'mouse',  
    'monitor',  
]
```

HTML:

```
<ul>  
    @for (produto of produtos; track produto){  
        <li> {{ produto }}</li>  
    }  
</ul>
```

Resultado:

Dentro de uma lista não-ordenada (ul) são gerados os elementos (li) dinamicamente.

- teclado
- mouse
- monitor



Diretivas Estruturais - @switch

Serve para: Mostrar **apenas uma** das opções com base em um valor.

Ts:

```
status : string = 'pendente';
```

HTML:

```
@switch (status){  
    @case ('aprovado'){  
        <p>Aprovado</p>  
    }  
    @case ('rejeitado'){  
        <p>Rejeitado</p>  
    }  
    @default {  
        <p>Pendente</p>  
    }  
}
```

É como menu muda o prato servido
dependendo da escolha do cliente.



Diretivas de Atributo

Elas mudam o **estilo** ou **comportamento** de um elemento já existente.

No exemplo abaixo (**classbinding**) **[class]='variável'** temos 3 formas diferentes.

Ts:

HTML:

```
listClasses = 'full-width outlined';  
sectionClasses = ['expandable', 'elevated'];  
buttonClasses = {  
  highlighted: true,  
  embiggened: false,  
};
```

```
<ul [class]="listClasses"> ... </ul>  
<section [class]="sectionClasses"> ... </section>  
<button [class]="buttonClasses"> ... </button>
```



```
<ul class="full-width outlined"> ... </ul>  
<section class="expandable elevated"> ... </section>  
<button class="highlighted"> ... </button>
```



Diretivas de Atributo – CSS class binding

Muda o **classe** baseado numa condição.

Serve para: Aplicar classes CSS de forma dinâmica.

Ts:

```
emPromocao : boolean = true;
```

HTML:

```
<p [class.destaque]="emPromocao">Produto</p>
```

Só aplica a classe **destaque** no elemento <p> “Produto”, somente se *emPromocao* for verdadeiro.



Diretivas de Atributo – CSS style property binding

Você também pode usar para colocar propriedades CSS diretamente nos elementos. No exemplo usaremos **display:flex; / display:block;**

Ts:

```
boolFlexbox : boolean = true;
```

HTML:

```
<div [style.display]="boolFlexbox ? 'flex' : 'block'">...</div>
```

Ternário: **CONDIÇÃO** ? **Caso verdadeiro** : **caso falso**



Exercício juntos

1. Crie um componente chamado “**Produtos**”.
2. Crie os valores dos produtos no Ts.
3. Crie o estilo no CSS.
4. Crie o HTML
5. Rode e veja

```
produtos : ({...})[] = [  
  { nome: 'Notebook', promocao: true },  
  { nome: 'Mouse', promocao: false },  
  { nome: 'Teclado', promocao: true }  
];
```

```
<ul>  
  @for (p of produtos; track p) {  
    <li [class.promocao]="p.promocao">  
      {{p.nome}}  
      @if (p.promocao){<span> - Em promoção!</span>}  
    </li>  
  }  
</ul>
```

```
.promocao {  
  color: lightgreen;  
  font-weight: bold;  
}
```

- Notebook - Em promoção!
- Mouse
- Teclado - Em promoção!

Desafio: Lista Dinâmica de Tarefas

Objetivo: Criar uma lista onde o usuário possa:

Ver todas as tarefas

Clicar para marcar como concluída (opcional, para os avançados)

Tarefas concluídas aparecem com um estilo diferente

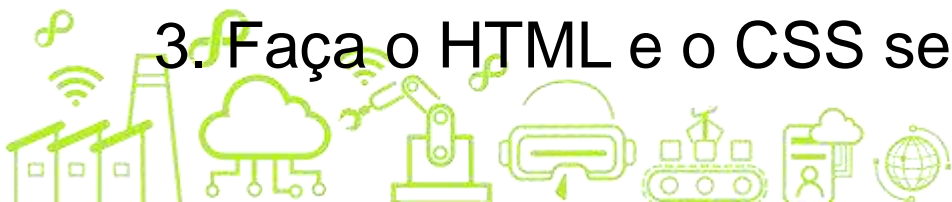
1. Crie uma lista com 3 tarefas:

Estudar Angular, Fazer Exercício, Fazer Atividade

2. (opcional) Faça uma função que faça o Toggle de Tarefa Concluída.

Toggle é **true=>false** e **false=>>true**

3. Faça o HTML e o CSS serem dinâmicos.



Obrigado!



    @fpftech.educacional