

# PROGRAMAÇÃO WEBII

1

# Data Binding

Interpolation e Property

# O que é Data Binding?

- Ligação de dados, do inglês Data Binding,
- É uma técnica geral que une duas fontes de dados/informações e as mantém em sincronia em um processo que estabelece uma conexão entre UI da aplicação e a logica de negocio.
- Possibilidade de exibir nossas variáveis dentro do html da nossa página.

# Por que aprender?

- Hoje, todos os frameworks SPA (Single Page Application) que existem como AngularJS, Ember, React e outros, fazem alguma espécie de Data Binding.
- Cada vez que esses frameworks ganham popularidade, essa prática se torna cada vez mais adorada pelos desenvolvedores front-ends.

# O que é Data Binding? resumindo...

- Possibilidade de exibir nossas variáveis dentro do html da nossa página.

# Objetivo da utilização do Data Binding



# O que é Data Binding?

- Anteriormente, criamos alguns escopos de variáveis, ficou vermelho informando que tinha algo de errado.
- Se vocês forem para o servidor, ficou vermelho e se vocês forem para a página ficou vermelho.

```
constructor() {  
  // variáveis de string com concatenação  
  // this.anuncio = 'O ' + this.nomeProduto + 'está em promoção!';  
  
  console.log('Nome do Produto', this.nomeProduto);  
  console.log('Anuncio', this.anuncio);  
  console.log('ID', this.idProduto);  
  console.log('Preço', this.precoProduto);  
  console.log('Promoção', this.promocao);  
  
  //Escopo das variáveis dentro do código  
  var variavel1;  
  let variavel2;  
  const variavel3;  
}
```

```
✓ Compiled successfully.  
✓ Browser application bundle generation complete.  
  
Initial Chunk Files | Names | Size  
main.js | main | 0 bytes  
  
5 unchanged chunks  
  
Build at: 2021-04-05T04:10:25.178Z - Hash: a6c9224dc9c347c6d52 - Time: 1436ms  
  
Error: src/app/components/home/home.component.ts:29:11 - error TS1155: 'const' declarations must be initialized.  
  
29   const variavel3;  
    ~~~~~  
  
Error: ./src/app/components/home/home.component.ts 19:23  
Module parse failed: Unexpected token (19:23)  
File was processed with these loaders:  
  * ./node_modules/@ngtools/webpack/src/ivy/index.js  
  * ./node_modules/@ngtools/webpack/src/ivy/index.js  
You may need an additional loader to handle the result of these loaders.  
  |  
  |   var variavel1;  
  |   let variavel2;  
  |   const variavel3;  
  |  
  | }  
  | ngOnInit() {  
  |  
  | }
```

```
CRUD App  
Failed to compile.  
  
src/app/components/home/home.component.ts:29:11 - error TS1155: 'const' declarations must be initialized.  
  
29   const variavel3;  
    ~~~~~  
  
CPDS  
CREATE READ UPDATE DELETE  
CRUD
```

# Vamos ver se já corrigiu?

- O que é isso?
  - ▶ Quando criamos uma constante, precisamos declarar um valor para ela, já que esse valor nunca será alterado durante a execução desse componente.
  - ▶ Vamos declarar com o valor = 1

```
//Escopo das variáveis dentro do código  
var variavel1;  
let variavel2;  
const variavel3 = 1;
```



# Vamos corrigir isso...

- Vamos ir no prompt do DOS e verificamos que voltou a compilar o projeto inteiro e já exibiu a mensagem que foi compilado com sucesso.
- Se retornarmos para a página já está tudo funcionando normal.

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **  
✓ Browser application bundle generation complete.  
  
Initial Chunk Files | Names |      Size  
main.js             | main  | 25.60 kB  
  
5 unchanged chunks  
  
Build at: 2021-08-22T21:40:29.555Z - Hash: 25a7d27327dc64b36767 - Time: 328ms  
✓ Compiled successfully.
```



Seja bem vindo ao curso de Angular!

# Deletando...

- Porém, para esse curso, iremos deletar esse escopo de variáveis e iremos criar em outro momento.
- Deletando, iremos deixar nossa tela mais limpa para que possamos trabalhar.

```
17     constructor() {  
18         // variáveis de string com concatenação  
19         // this.anuncio = 'O ' + this.nomeProduto + 'está em promoção!';  
20  
21         console.log('Nome do Produto', this.nomeProduto);  
22         console.log('Anuncio', this.anuncio);  
23         console.log('ID', this.idProduto);  
24         console.log('Preço', this.precoProduto);  
25         console.log('Promoção', this.promocao);  
26     }  
27  
28     ngOnInit(): void {  
29     }  
30 }
```

# Resumindo...

- O Data Binding é a possibilidade de manter uma sincronia entre as variáveis que temos dentro do nosso arquivo ts com o arquivo html que é exibido para o usuário.
- Toda vez que a variável mudar o seu valor dentro do typescript, automaticamente será renderizado o valor para o html e para a nossa página html.

# Resumindo...

- Vamos acessar o Angular (<https://angular.io/>) , onde explica para nós a sintaxe do binding.
- Pesquise por data e selecione Binding Syntax

The screenshot shows the Angular.io website with the search bar in the top right corner containing the text 'data'. Below the search bar, the results are organized into five columns: API (42), CLI (2), ERRORS (1), GUIDES (75), and OTHER (3). The 'GUIDES' column is highlighted, and the 'Binding syntax' link is circled in red. The 'API' column lists 'Data', 'RendererType2', 'Route', 'ActivatedRoute', and 'Resolve'. The 'CLI' column lists 'Gathering and Viewing Usage Analytics' and 'CLI Overview and Command Reference'. The 'ERRORS' column lists 'NG0100: Expression has changed after it was checked'. The 'OTHER' column lists 'Contributors page', 'Test Code Page', and 'Usage Metrics Gathering'.

API (42)	CLI (2)	ERRORS (1)	GUIDES (75)	OTHER (3)
<b>T</b> Data	Gathering and Viewing Usage Analytics	NG0100: Expression has changed after it was checked	Transforming Data Using Pipes	Contributors page
<b>I</b> RendererType2			Sharing data between child and parent directives and components	Test Code Page
<b>I</b> Route	CLI Overview and Command Reference		Introduction to forms in Angular	Usage Metrics Gathering
<b>C</b> ActivatedRoute			<b>Binding syntax</b>	
<b>I</b> Resolve				

# Tipos de binding

- Na página, o que interessa para nós é entender quais são os tipos de binding que podemos fazer.
- Nesse momento, iremos fazer o interpolation.

## Types of data binding

Angular provides three categories of data binding according to the direction of data flow:

- From the source to view
- From view to source
- In a two way sequence of view to source to view

Type	Syntax	Category
Interpolation Property Attribute Class Style	<code>{{expression}}</code> <code>[target]= expression"</code> <code>bind-target="expression"</code>	One-way from data source to view target

# Interpolation

- Interpolation, interpolação é o uso de duas chaves com a variável para exibir a nossa informação na tela para o usuário.
- Ele é categorizado como one-way, ou seja, só tem uma via. A informação só sai do typeScript e vai para o html. Não temos a possibilidade de mexer no html e enviar essa informação para o typeScript.

Type	Syntax	Category
Interpolation Property Attribute Class Style	<code>{{expression}}</code> <code>[target] "expression"</code> <code>bind-target="expression"</code>	One-way from data source to view target

ts para html

# Praticando...

- Vamos voltar para o código, no [home.component.html](#)
- Vamos apagar as divs que criamos;
- Digitar h4 <TAB>:
- <h4></h4>
- Dentro do h4, vamos criar as `{{}}` para fazer a interpolação e vamos digitar a nossa variável `anuncio` que criamos dentro do `home.component.ts`

```
1 <h1>Seja bem vindo ao curso de Angular!</h1>  
2 <h4>{{ anuncio }}</h4>
```

```
nomeProduto: string = 'Curso de Angular';  
anuncio: string = `O ${this.nomeProduto} está em promoção`;
```

→ Abrir o [home.component.ts](#)  
Aqui está a variável `anuncio`,  
lembrando que é uma variável  
template string

# Praticando...

- Vamos salvar e verificar como ficou a nossa tela.
- Nossa variável já está sendo exibida na tela.



# Praticando...

- Fizemos nossa primeira variável, primeira interpolação
- Para continuarmos, vamos arrumar nossa tela home.component
- Vamos acessar o bootstrap: <https://getbootstrap.com/>
  - ▷ Documentação 4.6
  - ▷ Utilities
  - ▷ Colors

```
2    <h4 class="text-danger">{{ anuncio }}</h4>
```

- Após o h4, vamos escrever o class = text-danger, que deixará a letra vermelha

# Praticando...

- Vamos abrir o site do bootstrap – documentação 4.6
- Selecionar components
- Vamos utilizar um card na nossa página.
- Procure por card title
- Copie o código dele:

## Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Go somewhere

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

Copy

# Praticando...

- Vamos para a nossa página home.component.html
- Vamos colar o código, após o <h4> </h4>
- Nessa imagem temos uma imagem, que iremos verificar depois.
- Temos o h5 que é o título do nosso card. Vamos substituir o título do card por duas chaves {{}} e dentro das chaves, vamos chamar nossa variável nomeProduto
- Vamos salvar e verificar como está a nossa tela.

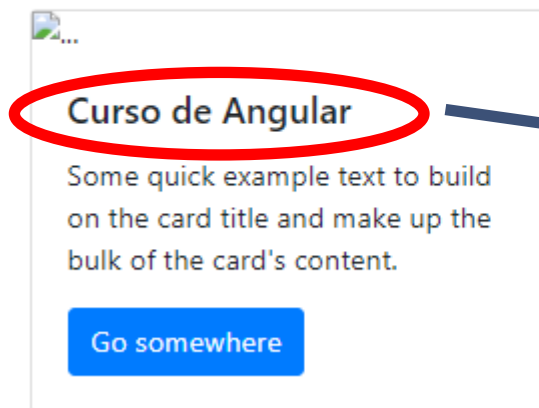
```
1 <h1>Seja bem vindo ao curso de Angular!</h1>
2
3 <h4 class="text-danger">{{ anuncio }}</h4>
4
5 <div class="card" style="width: 18rem;">
6   
7   <div class="card-body">
8     <h5 class="card-title">Card title</h5>
9     <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's
10     <a href="#" class="btn btn-primary">Go somewhere</a>
11   </div>
12 </div>
```

# Praticando...

```
5 <div class="card" style="width: 18rem;">
6   
7   <div class="card-body">
8     <h5 class="card-title">{{nomeProduto}}</h5>
9     <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's
10    <a href="#" class="btn btn-primary">Go somewhere</a>
11  </div>
12 </div>
```

## Seja bem vindo ao curso de Angular!

O Curso de Angular está em promoção



Já está exibindo Curso de Angular

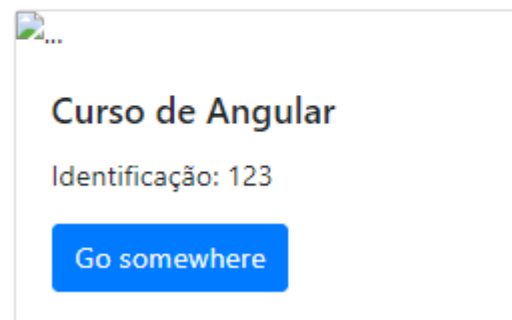
# Praticando...

- Dentro do p do card-text, vamos escrever Identificação e depois uma interpolação com o idProduto
- Identificação: {{idProduto}}
- Vamos formatar nossa tela: <ALT><SHIFT><F>

```
5 <div class="card" style="width: 18rem">
6   
7   <div class="card-body">
8     <h5 class="card-title">{{ nomeProduto }}</h5>
9     <p class="card-text">Identificação: {{ idProduto }}</p>
10    <a href="#" class="btn btn-primary">Go somewhere</a>
11  </div>
12 </div>
```

## Seja bem vindo ao curso de Angular!

O Curso de Angular está em promoção



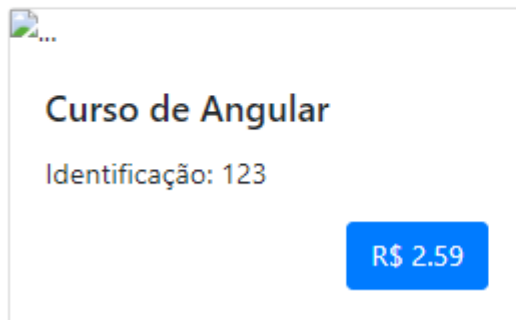
# Praticando...

- Vamos agora dentro do a, do botão e vamos colocar outra interpolação e vamos chamar o preço do produto
- Antes da interpolação, vamos inserir um R\$ para mudar um pouco
- Após o btn-primary, vamos inserir: float-right -> para o botão ir para a direita.

```
10 <a href="#" class="btn btn-primary float-right">R$ {{precoProduto}}</a>
```

Seja bem vindo ao curso de Angular!

O Curso de Angular está em promoção



# Interpolation

- Acessando a documentação: nesse momento, estamos trabalhando com o data binding de interpolação que é enviar informação da nossa variável que está dentro do nosso arquivo de typeScript para a nossa página html.
- Agora, vamos inserir a imagem. Para colocar a imagem iremos utilizar outro tipo de data binding. Utilizaremos o property.
- Property vamos colocar dentro de colchetes [] e a partir do momento que colocarmos colchetes dentro da nossa propriedade, o que estiver entre aspas vira código javascript, typescript também.

Type	Syntax	Category
Interpolation		
Property	<pre>{{expression}} [target]="expression" bind-target="expression"</pre>	One-way from data source to view target
Attribute		
Class		
Style		

# Praticando...

- Vamos ver isso na prática:
- Temos o [home.component.ts](#) e temos uma promoção, vamos criar uma nova variável chamada foto, após promoção, será do tipo string e conterá assets/img/crud.png
- Vamos salvar e já consigo utilizar essa variável na nossa tela.

```
11  nomeProduto: string = 'Curso de Angular';  
12  anuncio: string = `0 ${this.nomeProduto} está em promoção`;  
13  idProduto: number = 123;  
14  precoProduto: number = 2.59;  
15  promocao: boolean = true;  
16  foto:string = 'assets/img/crud.png'
```



# Praticando...

- Vamos abrir o arquivo [home.component.html](#)
- No campo img, clicar 2x no nome src, ficará selecionado e digitar colchetes []
- Agora, dentro das aspas "" começamos a ter código javascript...
- Ao digitar f, as nossas extensões de serviço do Angular já trouxeram para nós que podemos utilizar a variável foto que nós criamos. (nem sempre)
- Vamos salvar e verificar como ficou nossa página.

Seja bem vindo ao curso de Angular!

O Curso de Angular está em promoção



## Praticando...

- Foi bem fácil exibir nossas variáveis dentro da nossa página de internet, uma vez que temos o data binding de interpolação que exibe o conteúdo de uma variável em tempo real dentro da nossa página html.

# Diretivas...

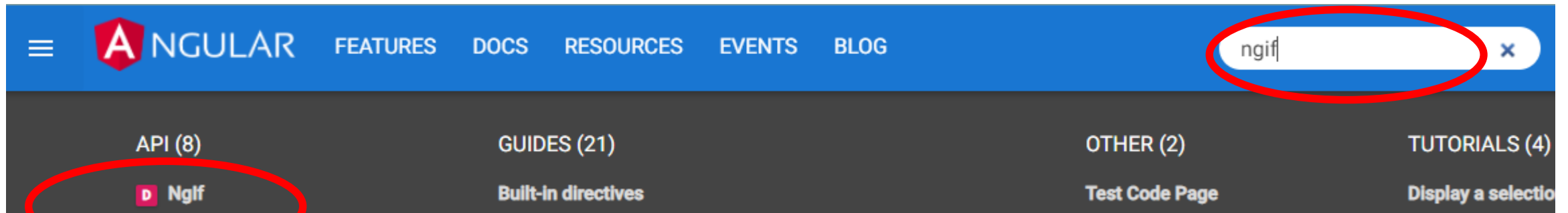
**Diretivas** são extensões da linguagem HTML, que fornecem a possibilidade de estender/ampliar o comportamento de elementos HTML.

Este recurso permite a implementação de novos comportamentos de forma declarativa.



# NgIf

- Vamos aprender agora a utilizar o NgIf.
- Entrar no site do angular e digitar NgIf



API > @angular/common

## NgIf

DIRECTIVE

A structural directive that conditionally includes a template based on the value of an expression coerced to Boolean. When the expression evaluates to true, Angular renders the template provided in a `then` clause, and when false or null, Angular renders the template provided in an optional `else` clause. The default template for the `else` clause is blank.

[See more...](#)

# NgIf

- NgIf é uma diretiva que muda a estrutura da nossa página.
- Vamos até a descrição, onde está escrito `*NgIf =` e uma condição

## Description

A [shorthand form](#) of the directive, `*ngIf="condition"`, is generally used, provided as an attribute of the anchor element for the inserted template. Angular expands this into a more explicit version, in which the anchor element is contained in an `<ng-template>` element.

## Praticando...

- Nós já temos uma variável que criamos aqui, denominada promoção – verificar no [home.component.ts](#)
- Agora iremos colocar dentro da nossa página html essa promoção.
- Vamos na tag <h5>, antes de finalizar a tag </h5>, vamos dar um <ENTER>
- Vou trazer o nomeProduto também para a linha de baixo.

```
8      <h5 class="card-title">
9          {{ nomeProduto }}
10
11      </h5>
```

## Praticando...

- Após a linha nomeProduto, vamos digitar span <TAB>
- Dentro do span, vamos colocar um class
- Acessando documentação do bootstrap
  - ▶ Versão 4.6 / Components / badge
- E dentro do class, digitar: badge badge-success (para fazer um badge verde)
- O conteúdo, vamos escrever promoção
- Vamos salvar.

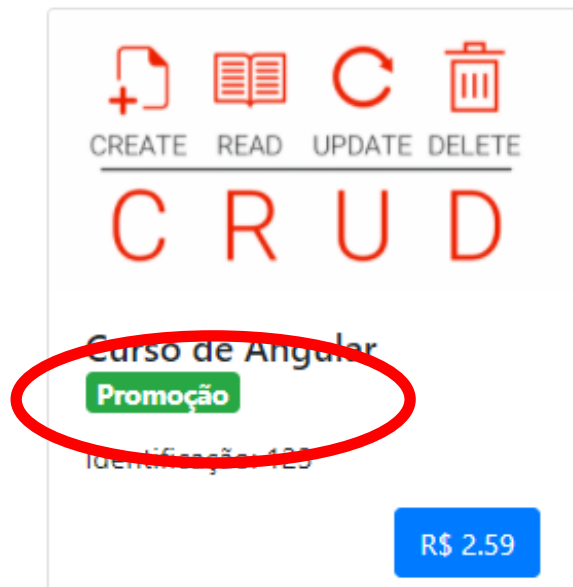
```
7      <div class="card-body">
8        <h5 class="card-title">
9          {{ nomeProduto }}
10         <span class="badge badge-success">Promoção</span>
11        </h5>
12      </div>
```

# Praticando...

Quando viermos na nossa página, iremos verificar que temos um badge informando que está em promoção:

**Seja bem vindo ao curso de Angular!**

O Curso de Angular está em promoção





## Praticando...

- Como fazemos para exibir somente se o produto estiver em promoção?
- Após o span, vamos digitar:
  - ▷ `*nglf=""` -> entre aspas iremos informar a condição
  - ▷ `*nglf=""` -> uma convenção de nomenclatura sempre que uma palavra muda, colocamos em letra maiúscula, por isso o `l` é maiúsculo.

## Praticando...

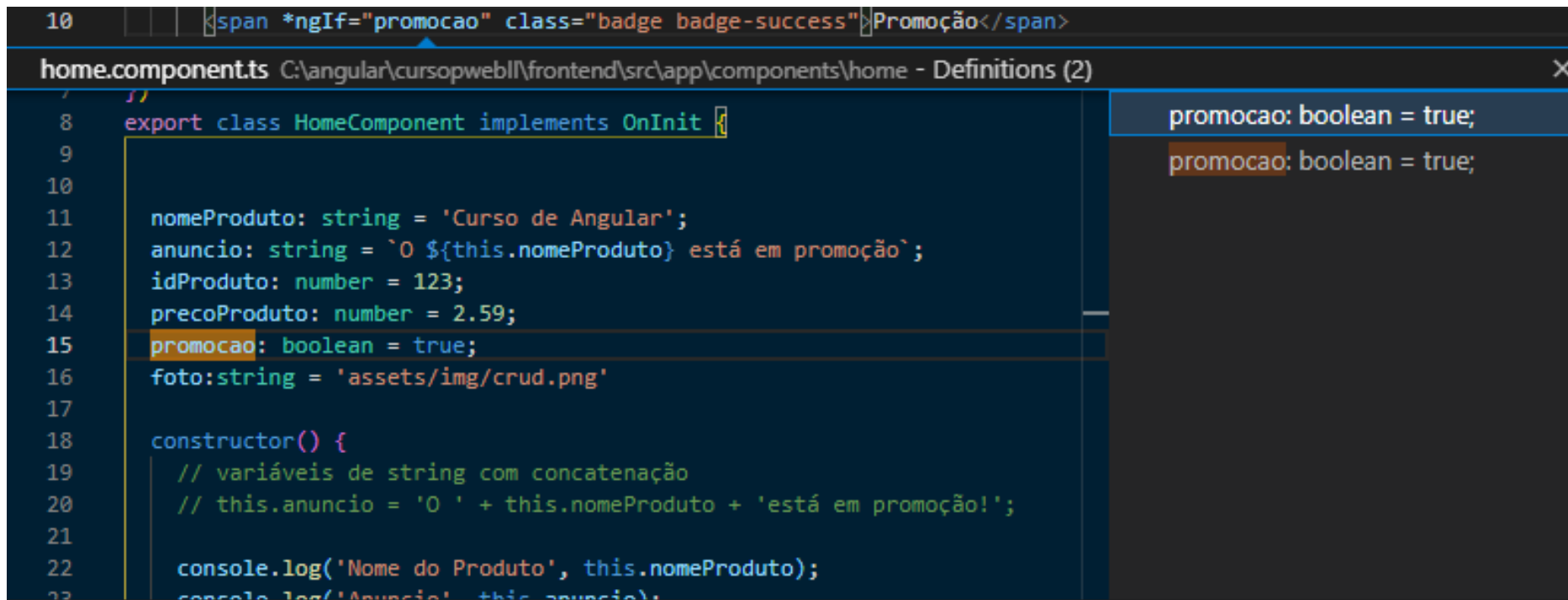
- A condição é se nosso produto está em promoção. Lembrando que nossa variável está aqui e tem um valor booleano, que é true.

```
<span *ngIf="promocao" class="badge badge-success">Promoção</span>
```

- Vamos salvar..

# Praticando...

■ Segurando o <CTRL> na variável `promocao`, podemos verificar que a variável está aqui e é do tipo `boolean` que é igual a `true`.



The screenshot shows an IDE with a TypeScript file named `home.component.ts` open. The file contains the following code:

```
10 <span *ngIf="promocao" class="badge badge-success">Promoção</span>
home.component.ts C:\angular\cursoweb\frontend\src\app\components\home - Definitions (2)
7
8 export class HomeComponent implements OnInit {
9
10
11     nomeProduto: string = 'Curso de Angular';
12     anuncio: string = `O ${this.nomeProduto} está em promoção`;
13     idProduto: number = 123;
14     precoProduto: number = 2.59;
15     promocao: boolean = true;
16     foto:string = 'assets/img/crud.png'
17
18     constructor() {
19         // variáveis de string com concatenação
20         // this.anuncio = 'O ' + this.nomeProduto + 'está em promoção!';
21
22         console.log('Nome do Produto', this.nomeProduto);
23         console.log('Anuncio', this.anuncio);
24     }
25 }
```

On the right side, a "Definitions (2)" panel is open, showing the definition of the `promocao` variable:

```
promocao: boolean = true;
promocao: boolean = true;
```

# Praticando...

- Na nossa página, a promoção continua sendo exibida.

Seja bem vindo ao curso de Angular!

O Curso de Angular está em promoção



CREATE READ UPDATE DELETE

C R U D

Curso de Angular

Promoção

Identificação: 123

R\$ 2.59

## Praticando...

- Como devemos fazer para não aparecer mais?
- Vamos no [home.component.ts](#)
- Onde está promoção: boolean = true, vamos substituir para false

```
promocao: boolean = false;
```

## Praticando...

- Quando salvar, a página é recarregada e não temos mais a promoção aqui na página.
- Lembrando que... Só precisou ser recarregada nossa página porque alteramos nosso código fonte manualmente. Se essa variável fosse alterada através de código ou método que foi executado posteriormente, a nossa página iria tirar essa promoção em tempo real, sem precisar recarregar tudo. Lembrando que estamos trabalhando com SPA, aplicações de página única.

Seja bem vindo ao curso de Angular!  
O Curso de Angular está em promoção



## Praticando...

- Se quisermos fazer um else, como ficaria?
- Vamos na documentação para consultar...
- <https://angular.io/api/common/NgIf>

## Praticando...

- Vamos ter nossa div com o \*ngIf, onde teremos a nossa condição e depois temos que passar para o else, um elseBlock
- Para inserir o else, nós precisamos criar um elemento chamado ng-template
- Vamos copiar essa linha do ngtemplate

Form with an "else" block:

```
<div *ngIf="condition; else elseBlock">Content to render when condition is true.  
</div>  
<ng-template #elseBlock>Content to render when condition is false.</ng-template>
```





## Praticando...

- Na linha abaixo do span, vamos colar o ng-template

```
10 <span *ngIf="promocao" class="badge badge-success">Promoção</span>  
11 <ng-template #elseBlock>Content to render when condition is false.</ng-template>
```

- Após a promoção, vamos digitar ; else elseBlock “
- Apaga o que está dentro do elseBlock porque iremos ter que exibir quando a condição for falsa
- Vamos fazer um outro span caso a promoção seja false

```
10 <span *ngIf="promocao;else elseBlock " class="badge badge-success">Promoção</span>  
11 <ng-template #elseBlock>  
12 | <span class="badge badge-primary">Aproveite</span>  
13 </ng-template>
```

## Praticando...

■ O nome `elseBlock` pode ser substituído por outro qualquer, mas vamos deixar o padrão.

```
10 <span *ngIf="promocao;else elseBlock " class="badge badge-success">Promoção</span>  
11 <ng-template #elseBlock>  
12   <span class="badge badge-primary">Aproveite</span>  
13 </ng-template>
```

## Praticando...

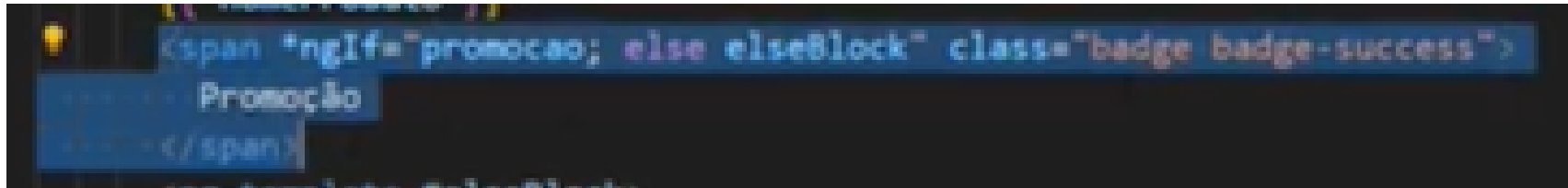
- Já temos aqui o primeiro span que é o if (Se promoção for verdadeiro). Na condição do \*ngIf, podemos fazer qualquer comparação.... Como o campo é booleano não precisa, mas poderíamos fazer, por exemplo:

\*ngIf= "promoção == true, mas ficaria redundante nesse caso. Aqui poderia fazer um teste, qualquer teste, utilizar >, <, =. Vamos apagar porque promoção já retorna um booleano.

```
<span *ngIf="promocao == true; else elseBlock" class="badge badge-success">
```

## Praticando...

- Se promoção for verdadeiro, irá exibir o span abaixo, que é o badge de success que ele irá exibir na tela Promoção:



```
<span *ngIf="promocao; else elseBlock" class="badge badge-success">  
  Promoção  
</span>
```

- Senão, exibirá o bloco nosso de else. O bloco de else tem que estar dentro de um ng-template, depois iremos colocar um jogo da velha #elseBlock. Novamente, esse nome elseBlock vocês podem trocar, vocês podem colocar o nome que for mais fácil de entender, mas como padrão, vamos deixar elseBlock
- Depois, dentro do ng-template podemos ter todo o novo código que só será exibido se a condição for falsa e então colocamos um span com outro badge que estará escrito Aproveite.

# Praticando...

- Vamos salvar e ver a nossa página.
- Apareceu Aproveite, porque promoção é false. Vamos alterar para true e verificar se o banner é alterado.

## Seja bem vindo ao curso de Angular!

O Curso de Angular está em promoção



CREATE READ UPDATE DELETE

C R U D

Curso de Angular [Aproveite](#)

Identificação: 123

R\$ 2.59

