**Conceitos do Angular**

**Componentes**

Ao criar um novo componente no Angular, você recebe 3 arquivos, o arquivo HTML, CSS e TS. No arquivo TS é onde vai estar a referência dos arquivos CSS e HTML.

Também estará o **seletor** do componente. O **seletor** é o nome que vai ser usado caso esse componente precise ser chamado em outros componentes.

As frameworks de estilização tem que ser importadas sempre no arquivo principal da aplicação “AppModule”.

Só é possível usar os componentes que estão carregados dentro do “AppModule”.

**Scripts**

**ng new nomeProjeto --minimal** - cria um projeto angular. O --*minimal* tira os arquivos e pastas de testes do projeto.

**DETALHE:** quando criamos o projeto usando o -- minimal, ele coloca os arquivos de estilizacao, módulo e html em um arquivo só. Para modificar isso e separar os arquivos, basta ir até o arquivo **angular.json**, procurar por **inlineTemplate\*\*** e **inlineStyle** e modificar o atributo para **false.**

**ng add @angular/material** - instala a lib do MaterialDesign

**ng generate componente nomePasta/nomeArquivo**- cria um novo componente

**Elementos do Angular**

**Diretiva de Atributos**

Altera a **aparência** ou o **comportamento** de um elemento, componente ou outra diretiva.

**Exemplo:**



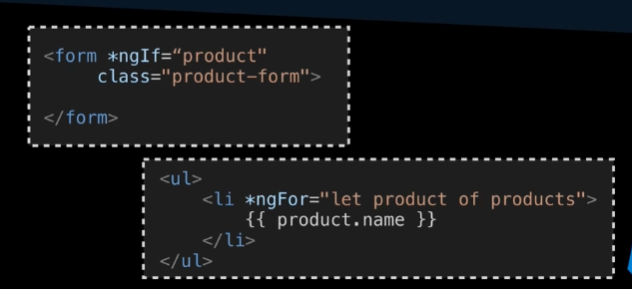


**Diretiva Estrutural**

Altera o layout **adicionando** e **removendo** elementos da **DOM.**

As diretivas estruturais vem com o **\*** na frente.

**Exemplo:**



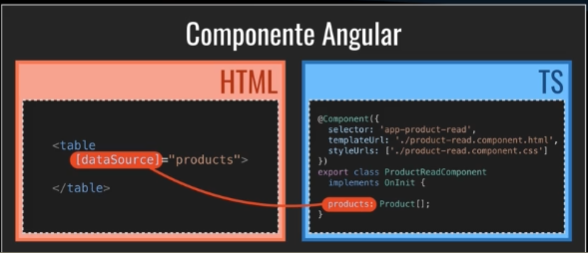
O **\*ngIf** significa que o elemento **form** vai ser exibido se **product** for **true.**

O **\*ngFor** significa que ele a quantidade de produtos que o **product** tem, ou seja, se o **product** têm 10 produtos, ele vai rederizar 10 <li>.

**Property Binding**

Com os **Property Binding** você pode acessar atributos do arquivo TypeScript no HTML.

**Exemplo:**

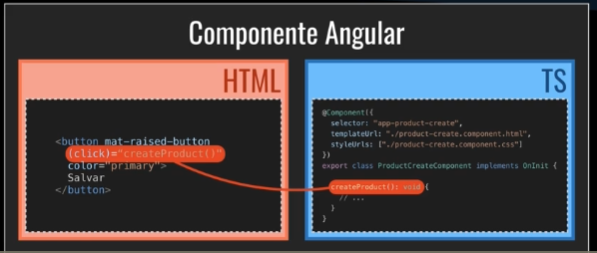


O atributo **[dataSource]** precisa estar entre colchetes pois isto significa que ele vai verificar se no arquivo TS existe um atributo com o mesmo conteúdo passado, se tiver, ele vai retornar o conteúdo do atributo.

**Event Binding**

Os eventos são referenciados por **()** dentro de uma tag HTML.

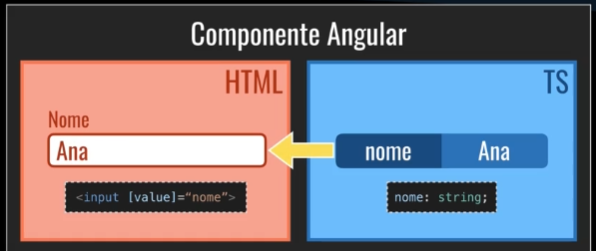
**Exemplo:**



**One Way Data Binding**

O One Way Data Binding é quando você altera o valor de algum atributo no arquivo TS e o valor novo reflete no arquivo HTML. Ou seja, você só consegue alterar o valor do atributo no arquivo TS.

**Exemplo:**

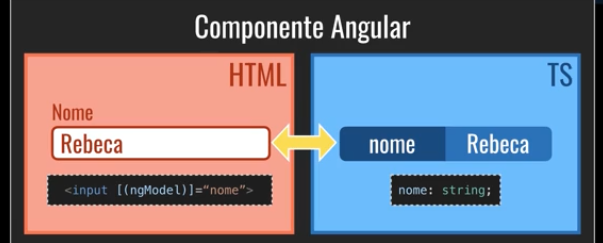


**[value] –** esta é a sintaxe para usar o One Way Data Binding

**Two Way Data Binding**

Significa que se você alterar o atributo no TS ele vai atualizar no HTML e se você alterar o atributo no HTML ele vai atualizar no TS. Ou seja, o sentido de alteração do atributo é tanto do TS quanto do HTML.

**Exemplo:**



**[(ngModel)] –** esta é a sintaxe para se usar o Two Way Data Binding.

**Angular Router**

Faz o mapeamento das rotas dos componentes da aplicação.

**Angular Pipes**

**Programação Reativa**

O conceito de programação Reativa significa que o código só vai ser chamado se acontecer alguma coisa externa(evento).

**Padrão Observer**

Padrão orientado a **Evento!**

**Exemplo:**



Está imagem está dizendo que:

O Subject detecta o evento que foi realizado e nisso ele notifica os observer(funções) que aquele evento aconteceu, para que aí então os observer possa fazer o que eles precisam fazer. Este é um exemplo de **Programação Reativa.**

**Services**

São classes que têm como principal objetivo **organizar** e **compartilhar** métodos e dados entre componentes.

O ideal, é que o componente tenha responsabilidades que diz respeito a parte visual, e caso ele precise utilizar um método para acessar a API, o ideal é fazer dentro do **Services.**

As responsabilidades dentro de um Service devem estar coerentes, ou seja, se você criou um Service para fazer um CRUD de produtos, só deve estar neste Service o que está relacionado ao CRUD.

**ng g s services/nomeArquivo –** cria um Service no Angular.

**Exemplo de arquivo Service:**



**Injeção de Dependência**

É um padrão no qual a classe recebe as **dependências** de uma **fonte externa** ao invés de criar por conta própria.