Pontos chave:

- **O que são templates?** Templates são arquivos HTML que definem a estrutura visual de um componente Angular. Eles contêm elementos HTML como divs, paragraphs e headings, além de diretivas Angular que permitem interagir com dados e lógica de programação.
- **Criando templates:** Templates podem ser criados em arquivos separados com a extensão `.html` ou diretamente dentro do código do componente utilizando a sintaxe `template`.
- **Interpolação:** A interpolação permite inserir valores de variáveis dentro do template utilizando as chaves `{{ }}`. Isso permite que o template seja dinâmico e se atualize automaticamente conforme os dados mudam.
- **Diretivas:** Diretivas Angular são atributos HTML especiais que fornecem funcionalidades adicionais aos templates. Existem diversas diretivas disponíveis, como `ngFor`, `ngIf` e `ngClass`, que permitem iterar sobre listas, exibir ou ocultar elementos e aplicar classes CSS dinamicamente.
- **Componentes:** Templates são utilizados em conjunto com componentes Angular para criar interfaces de usuário reutilizáveis e modulares. Um componente pode ter um ou mais templates associados a ele.

Exemplos de como estruturar o template no Angular

1. Modularidade:

- Divida seus templates em componentes menores e reutilizáveis. Isso facilita a manutenção e o código fica mais organizado.
- Utilize componentes para encapsular funcionalidades específicas da interface.
- Crie diretivas personalizadas para abstrair código repetitivo.

2. Clareza:

- Utilize código HTML limpo e bem formatado.
- Adicione comentários para explicar o que cada parte do código faz.
- Utilize indentação adequada para melhorar a legibilidade.

3. Concisão:

- Evite código HTML redundante e desnecessário.
- Utilize diretivas e componentes para reduzir a quantidade de código HTML repetitivo.

- Utilize expressões Angular para simplificar a manipulação de dados.

- **4. Reutilização:**
- Crie componentes e diretivas reutilizáveis para evitar código repetitivo.
- Utilize bibliotecas de componentes de terceiros para aproveitar código já pronto.
- Compartilhe seus componentes e diretivas com a comunidade.

5. Acessibilidade:

- Torne seus templates acessíveis a todos os usuários, incluindo pessoas com deficiências.
- Utilize tags HTML semânticas para estruturar o conteúdo.
- Forneça alternativas de texto para imagens e outros elementos não textuais.
- Utilize cores e contrastes adequados para facilitar a leitura.

Exemplos práticos:

- **Componente para exibir uma lista de produtos:**

HTML
```
<div class="produto">
  <h2>{{ produto.nome }}</h2>
  <img src="{{ produto.imagem }}" alt="{{ produto.nome }}">
  <p>{{ produto.descricao }}</p>
  <button (click)="adicionarCarrinho()">Adicionar ao
carrinho</button>
</div>
```

- **Diretiva para formatar números como moeda:**

HTML
```
<p>{{ preco | currency:'BRL' }}</p>
```

- **Componente para exibir um formulário de contato:**

HTML
```
<form [formGroup]="contatoForm">
  <input type="text" formControlName="nome" placeholder="Nome">
  <input type="email" formControlName="email" placeholder="Email">
  <textarea formControlName="mensagem"
placeholder="Mensagem"></textarea>
  <button type="submit">Enviar</button>
</form>
```

Recursos adicionais:

- Documentação oficial do Angular sobre templates: https://angular.io/guide/templates
- Curso Angular da Loiane Groner: https://www.youtube.com/playlist?list=PLn9d9zI0I-y5ZbU7tCMx4Fz04n4Ei4o3F