

1. O que são componentes Angular (criação e uso)

- **Conceito:** Componentes são a base da construção de aplicações em Angular. Eles representam partes da interface de usuário (UI). Cada componente tem um HTML, um CSS, e uma lógica em TypeScript.
- **Criação:**

```
arduino  
Copiar código  
ng generate component nome-do-componente
```

Isso cria uma pasta com os arquivos `.html`, `.css`, `.ts`, e `.spec.ts`.

- **Uso:** Para usar um componente, adicione sua tag no HTML de outro componente:

```
html  
Copiar código  
<app-nome-do-componente></app-nome-do-componente>
```

2. O que são serviços (criação e uso)

- **Conceito:** Serviços são usados para compartilhar dados ou lógica entre diferentes componentes. Exemplos incluem chamadas HTTP e armazenamento de estados.
- **Criação:**

```
arduino  
Copiar código  
ng generate service nome-do-servico
```

- **Uso:** Importe o serviço no componente desejado e injete no construtor:

```
typescript  
Copiar código  
constructor(private nomeDoServico: NomeDoServico) { }
```

3. Uso de grids com Ionic

- **Conceito:** O Ionic usa o sistema de grid baseado no Flexbox para layouts responsivos.
- **Exemplo:**

```
html  
Copiar código  
<ion-grid>  
  <ion-row>  
    <ion-col size="6">Coluna 1</ion-col>  
    <ion-col size="6">Coluna 2</ion-col>  
  </ion-row>
```

```
</ion-grid>
```

4. Como utilizar componentes em páginas

- Crie o componente e inclua-o no módulo correto.
- Insira o seletor no HTML da página onde deseja usá-lo, por exemplo:

```
html
Copiar código
<app-header></app-header>
```

5. Como atender a eventos com Angular

- Utilize **event binding** para responder a eventos do DOM, como cliques:

```
html
Copiar código
<button (click)="acao()">Clique</button>
```

- No TypeScript:

```
typescript
Copiar código
acao() {
  console.log('Botão clicado');
}
```

6. Como apresentar valores em tela

- **Interpolação:**

```
html
Copiar código
<p>{{ mensagem }}></p>
typescript
Copiar código
mensagem = "Olá, mundo!";
```

7. Como coletar dados de formulários

- Use formulários reativos ou baseados em templates:

```
html
Copiar código
<form (ngSubmit)="submitForm()" #meuForm="ngForm">
  <input name="nome" ngModel required>
  <button type="submit">Enviar</button>
</form>
```

- No TypeScript:

```
typescript
Copiar código
submitForm() {
  console.log('Formulário enviado');
}
```

8. Como gerar notificações utilizando Ionic no Angular

- Instale o plugin de notificações push:

```
perl
Copiar código
npm install @capacitor/push-notifications
```

- Configure e use o plugin para enviar notificações.

EXEMPLOS:

1. Componentes Angular (Criação e Uso)

Exemplo: Criar um componente de cabeçalho e usá-lo.

1. Criar o componente:

```
css
Copiar código
ng generate component header
```

2. No arquivo header.component.html:

```
html
Copiar código
<header>
  <h1>{{ titulo }}</h1>
</header>
```

3. No arquivo header.component.ts:

```
typescript
Copiar código
import { Component } from '@angular/core';

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.css']
})
export class HeaderComponent {
  titulo = 'Meu Cabeçalho';
}
```

```
}
```

4. Usar no HTML de outra página:

```
html
Copiar código
<app-header></app-header>
```

2. Serviços (Criação e Uso)

Exemplo: Criar um serviço que retorna uma lista de usuários.

1. Criar o serviço:

```
sql
Copiar código
ng generate service user
```

2. No arquivo user.service.ts:

```
typescript
Copiar código
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class UserService {
  getUsers() {
    return ['Alice', 'Bob', 'Charlie'];
  }
}
```

3. Usar no componente:

```
typescript
Copiar código
import { Component } from '@angular/core';
import { UserService } from '../user.service';

@Component({
  selector: 'app-user-list',
  template: `
    <ul>
      <li *ngFor="let user of users">{{ user }}</li>
    </ul>
  `,
})
export class UserListComponent {
  users: string[] = [];

  constructor(private userService: UserService) {
    this.users = this.userService.getUsers();
  }
}
```

3. Uso de grids com Ionic

Exemplo: Criar um layout com 3 colunas.

```
html
Copiar código
<ion-grid>
  <ion-row>
    <ion-col size="4">Coluna 1</ion-col>
    <ion-col size="4">Coluna 2</ion-col>
    <ion-col size="4">Coluna 3</ion-col>
  </ion-row>
</ion-grid>
```

4. Como utilizar componentes em páginas

Exemplo: Reutilizar um componente de rodapé.

1. Criar o componente:

```
css
Copiar código
ng generate component footer
```

2. No arquivo footer.component.html:

```
html
Copiar código
<footer>
  <p>© 2024 Meu Aplicativo</p>
</footer>
```

3. Usar o componente em outra página:

```
html
Copiar código
<app-footer></app-footer>
```

5. Como atender a eventos com Angular

Exemplo: Capturar o clique de um botão.

1. No HTML:

```
html
Copiar código
<button (click)="saudar()">Clique aqui</button>
```

2. No TypeScript:

```
typescript
Copiar código
```

```
saudar() {  
  console.log('Olá, mundo!');  
}
```

6. Como apresentar valores em tela

Exemplo: Mostrar a data atual.

1. No HTML:

```
html  
Copiar código  
<p>A data atual é: {{ dataAtual | date:'dd/MM/yyyy' }}</p>
```

2. No TypeScript:

```
typescript  
Copiar código  
dataAtual = new Date();
```

7. Como coletar dados de formulários

Exemplo: Criar um formulário de login.

1. No HTML:

```
html  
Copiar código  
<form (ngSubmit)="logar()" #formLogin="ngForm">  
  <label for="email">Email:</label>  
  <input type="email" name="email" ngModel>  
  
  <label for="senha">Senha:</label>  
  <input type="password" name="senha" ngModel>  
  
  <button type="submit">Entrar</button>  
</form>
```

2. No TypeScript:

```
typescript  
Copiar código  
logar() {  
  console.log('Formulário enviado!');  
}
```

8. Como gerar notificações utilizando Ionic no Angular

Exemplo: Notificação local.

1. Instalar o plugin de notificações:

```
perl
Copiar código
npm install @capacitor/local-notifications
```

2. Configurar o plugin no TypeScript:

```
typescript
Copiar código
import { LocalNotifications } from '@capacitor/local-notifications';

async notificar() {
  await LocalNotifications.schedule({
    notifications: [
      {
        title: 'Lembrete',
        body: 'Essa é sua notificação!',
        id: 1,
        schedule: { at: new Date(new Date().getTime() + 1000 *
5) },
      },
    ],
  });
}
```

3. No HTML:

```
html
Copiar código
<button (click)="notificar()">Enviar Notificação</button>
```