

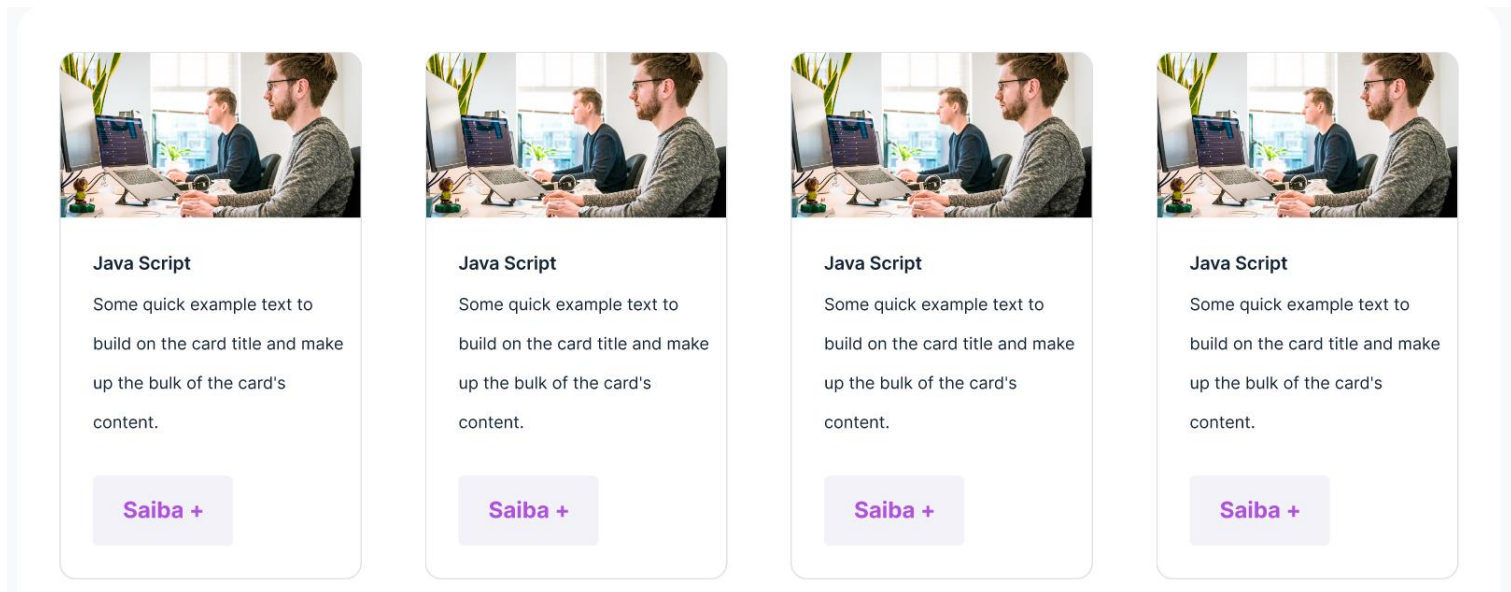
# Angular – Aula 4

Profa. Maria Janaina da Silva Ferreira



# Exercícios

- Crie a classe Courses, com os seguintes atributos: Id, IdCategory, Title, Description, Cost, Image, Duration.
- Crie o componente Courses, que deverá listar todos os cursos cadastrados, conforme apresentado na imagem abaixo:



# Correção dos exercícios

- Crie a classe courses (ng g class models/course)
- Adicione os atributos:

```
TS course.ts U X
1  export class Course {
2      id: number = 0;
3      idCategory: number = 0;
4      title: string = '';
5      description: string = '';
6      cost: number = 0;
7      image: string = '';
8      duration: number = 0;
9  }
```

# Correção dos exercícios

- Crie o componente course (ng g component course)

```
● PS C:\Users\saoavnia\Downloads\dswebII-app-master\dswebII-app-master> ng g component course
CREATE src/app/course/course.component.html (22 bytes)
CREATE src/app/course/course.component.spec.ts (619 bytes)
CREATE src/app/course/course.component.ts (246 bytes)
CREATE src/app/course/course.component.css (0 bytes)
```


# Correção dos exercícios

- Edite o arquivo `course.component.ts` e adicione a lista de cursos

```
TS course.component.ts U X
1 import { Component } from '@angular/core';
2 import { Course } from '../models/course';
3 import { CommonModule } from '@angular/common';
4
5 @Component({
6   selector: 'app-course',
7   standalone: true,
8   imports: [CommonModule],
9   templateUrl: './course.component.html',
10  styleUrls: ['./course.component.css']
11 })
12 export class CourseComponent {
13
14   courses: Course[] = [
15     {id: 1, idCategory: 1, title: 'Java para iniciantes', description: 'Aprenda Java de forma fácil: variáveis, constantes, funções, etc', cost: 100,
16      image: 'https://post.medicalnewstoday.com/wp-content/uploads/sites/3/2020/02/322868_1100-1100x628.jpg', duration: '10 meses'},
17     {id: 2, idCategory: 1, title: 'C# para iniciantes', description: 'Aprenda C# de forma fácil: variáveis, constantes, funções, etc', cost: 100,
18      image: 'https://post.medicalnewstoday.com/wp-content/uploads/sites/3/2020/02/322868_1100-1100x628.jpg', duration: '10 meses'},
19     {id: 3, idCategory: 1, title: 'Python para iniciantes', description: 'Aprenda Python de forma fácil: variáveis, constantes, funções, etc', cost: 100,
20      image: 'https://post.medicalnewstoday.com/wp-content/uploads/sites/3/2020/02/322868_1100-1100x628.jpg', duration: '10 meses'},
21     {id: 4, idCategory: 1, title: 'JavaScript para iniciantes', description: 'Aprenda JavaScript de forma fácil: variáveis, constantes, funções, etc', cost: 100,
22      image: 'https://post.medicalnewstoday.com/wp-content/uploads/sites/3/2020/02/322868_1100-1100x628.jpg', duration: '10 meses'},
23     {id: 5, idCategory: 1, title: 'SQL para iniciantes', description: 'Aprenda SQL de forma fácil: ddl, dml, procedures, funções, etc', cost: 100,
24      image: 'https://post.medicalnewstoday.com/wp-content/uploads/sites/3/2020/02/322868_1100-1100x628.jpg', duration: '10 meses'},
25     {id: 6, idCategory: 2, title: 'Inteligência Emocional', description: 'Aprenda agir de forma racional diante das diversidades. Dicas de especialistas etc', cost: 320.00,
26      image: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcSu_FDWsFa_HLgROHtgTtZJrbPktbyc1fluiw&s=', duration: '36 horas'},
27     {id: 7, idCategory: 2, title: 'Inteligência Emocional', description: 'Aprenda agir de forma racional diante das diversidades. Dicas de especialistas etc', cost: 320.00,
28      image: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcSu_FDWsFa_HLgROHtgTtZJrbPktbyc1fluiw&s=', duration: '36 horas'},
29     {id: 8, idCategory: 2, title: 'Inteligência Emocional', description: 'Aprenda agir de forma racional diante das diversidades. Dicas de especialistas etc', cost: 320.00,
30      image: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcSu_FDWsFa_HLgROHtgTtZJrbPktbyc1fluiw&s=', duration: '36 horas'},
31     {id: 9, idCategory: 2, title: 'Inteligência Emocional', description: 'Aprenda agir de forma racional diante das diversidades. Dicas de especialistas etc', cost: 320.00,
32      image: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcSu_FDWsFa_HLgROHtgTtZJrbPktbyc1fluiw&s=', duration: '36 horas'},
33   ]
34 }
```

# Correção dos exercícios

- Edite o arquivo `course.component.html` e adicione o código html abaixo para listar os cursos

```
<> course.component.html U X 
```

Go to component

```
1 <div class="container-fluid">
2 <div class="row" >
3   <div class="col-3" *ngFor="let course of courses">
4     <div class="card" style="width: 18rem">
5       
6       <div class="card-body">
7         <h5 class="card-title">{{ course.title }}</h5>
8         <p class="card-text">{{ course.description }}</p>
9         <a href="#" class="btn btn-primary">Saiba mais</a>
10      </div>
11    </div>
12  </div>
13 </div>
14 </div>
```

# Correção dos exercícios

- Edite o arquivo `course.component.css` e adicione o código css abaixo para estilizar a página

```
# course.component.css U X
1  .card-content {
2    margin-top: 65px;
3  }
4  .truncated-text {
5    max-width: 140px;
6    text-overflow: ellipsis;
7    overflow: hidden;
8    white-space: nowrap;
9  }
10 .truncated-text:hover {
11   overflow: visible;
12   white-space: normal;
13 }
```

# Correção dos exercícios

- Adicione as classes criadas no arquivo html

```
<> course.component.html U X
Go to component
1 <div class="container-fluid">
2 <div class="row" >
3   <div class="col-3 card-content" *ngFor="let course of courses">
4     <div class="card" style="width: 18rem">
5       
6       <div class="card-body">
7         <h5 class="card-title">{{ course.title }}</h5>
8         <p class="card-text truncated-text">{{ course.description }}</p>
9         <a href="#" class="btn btn-primary">Saiba mais</a>
10      </div>
11    </div>
12  </div>
13 </div>
14 </div>
```



# Rotas

- Rotas em Angular são utilizadas para redirecionamento entre componentes.
- No Angular, muita coisa é modularizada, e as rotas não são diferentes. Ao criar uma aplicação Angular, somos indagados sobre o uso das mesmas (Would you like to add Angular Routing?), e ao responder “Yes”, já temos tudo que precisamos para o roteamento.
- Na verdade, tudo que precisamos para ter rotas em nossa aplicação é um arquivo de módulo das rotas e importá-lo posteriormente no Root Module.

# Rotas

- Este arquivo de rotas, normalmente chamado app-routes.ts, tem a seguinte estrutura:

```
TS app.routes.ts X
1  import { Routes } from '@angular/router';
2
3  export const routes: Routes = [];
4
```

# Rotas

- As rotas são definidas por dois elementos principais, o Path, que é o caminho (URL) e o Componente, que é o componente a ser carregado. Ainda existem outras formas de carregar páginas mais eficientes, como carregamento assíncrono, via módulos.
- Este é o resultado da forma mais simples de rotas que podemos ter no Angular.

```
export const routes: Routes = [  
  { path: 'course', component: CourseComponent },  
  { path: 'courseDescription', component: CourseDescriptionComponent },  
];
```

# Rotas

- O roteamento permite que você exiba visualizações específicas do seu aplicativo dependendo do caminho da URL. Para adicionar essa funcionalidade ao seu aplicativo de exemplo, você precisa atualizar o arquivo `app.config.ts` para usar a função de provedores de roteador, `provideRouter`. Você importa essa função de provedor de `@angular/router`. No seu editor de código, abra o arquivo `app.config.ts`. Adicione as seguintes instruções de importação:

# Rotas

TS *app.config.ts* ✕

```
1  import { ApplicationConfig } from '@angular/core';
2  import { provideRouter } from '@angular/router';
3
4  import { routes } from './app.routes';
5  import { provideClientHydration } from '@angular/platform-browser';
6
7  export const appConfig: ApplicationConfig = {
8    providers: [provideRouter(routes), provideClientHydration()]
9  };
```

# Rotas

- Quando configuramos uma rota e navegamos até ela, o Angular recupera a URL, checa no arquivo de rotas e tenta carregar o componente na tela.
- Para que isto aconteça, precisamos de uma tag especial, chamada de Router Outlet, pois sempre que um fluxo como o descrito acima é chamado, o Angular busca esta tag para poder renderizar o componente desejado.

# Rotas - Links

- Um erro comum na navegação do Angular é utilizar o href padrão do HTML para link de páginas. O uso desta propriedade, fará com que nossa aplicação seja recarregada a cada navegação.
- Ao invés disso, utilizamos o routerLink do Angular, que fará com que apenas os componentes necessários sejam renderizados na tela.
- Exemplo:  
`<a routerLink="/login">Login</a><br>`  
`<a routerLink="/signup">Signup</a><br>`

# Rotas - Links

- Existe também, um elemento chamado Router Link Active, que nos permite deixar um link com algum CSS diferente quando sua rota está selecionada.

- Exemplo:

```
<a routerLink="/login" routerLinkActive="selected">Login</a>
```



# Rotas – Passagem de parâmetros

- Para passar parâmetros em rotas, precisamos primeiro definir o nome dele nas constantes do módulo de rotas.
- Exemplo:  
`{ path: 'home/:parametro', component: HomeComponent },`

# Rotas – Passagem de parâmetros

- Feito isto, podemos navegar novamente para página Home através da URL <https://localhost:4200/categories/1> e recuperar seu valor pelo ActivatedRoute.

- Exemplo:

```
export class HomeComponent implements OnInit {  
  constructor( private activatedRoute: ActivatedRoute ) {}  
  ngOnInit() {  
    const par =  
      this.activatedRoute.snapshot.paramMap.get('parametro')  
  }  
}
```

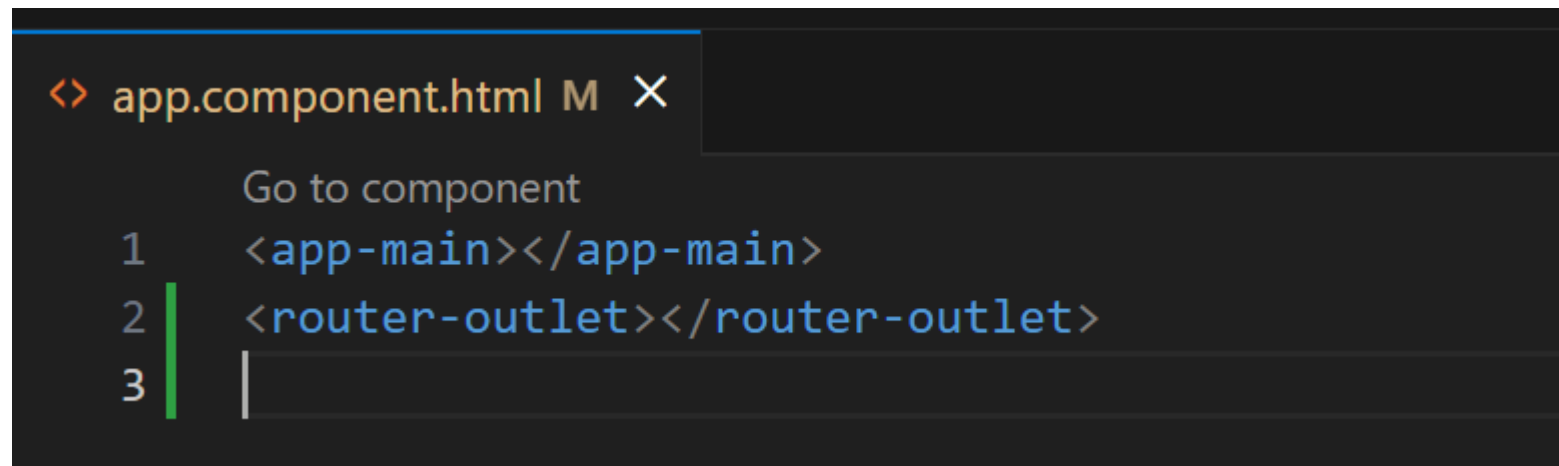
# Testando as rotas

- Adicione o componente courses-description (ng g component courses-description)
- Edite o arquivo html para listar a descrição do curso

```
<> course-description.component.html U X
Go to component
1 <div class="container-fluid">
2   <h1>Titulo do Curso</h1>
3   <h2>Categoria</h2>
4   <p>Descricao</p>
5   <p>Valor</p>
6   <p>Duracao</p>
7 </div>
8 |
```

# Criando as rotas do Projeto

- Agora vamos chamar o router-outlet que vai renderizar as nossas rotas.
- Abra o arquivo app.component.ts e adicione a tag <router-outlet>



```
<> app.component.html M X
Go to component
1 <app-main></app-main>
2 <router-outlet></router-outlet>
3 |
```

# Criando as rotas do Projeto

- Agora precisamos alterar o link Saiba Mais, para que abra o componente course-description.component.html
- Link para listar os detalhes do curso

```
<> course.component.html M X
Go to component
1 <div class="container-fluid">
2 <div class="row" >
3   <div class="col-3 card-content" *ngFor="let course of courses">
4     <div class="card" style="width: 18rem">
5       
6       <div class="card-body">
7         <h5 class="card-title">{{ course.title }}</h5>
8         <p class="card-text truncated-text">{{ course.description }}</p>
9         <a routerLink="/courseDescription" routerLinkActive="router-link-active" =="#"
10        class="btn btn-primary">Saiba mais</a>
11       </div>
12     </div>
13   </div>
14 </div>
15 </div>
16
```

# Criando as rotas do Projeto

- Adicione as rotas no arquivo de Rotas

```
TS app.routes.ts M X
1 | import { Component } from '@angular/core';
2 | import { Routes } from '@angular/router';
3 | import { CourseComponent } from '../course/course.component';
4 | import { CourseDescriptionComponent } from '../course-description/course-description.component';
5 |
6 | export const routes: Routes = [
7 |   { path: 'course', component: CourseComponent },
8 |   { path: 'courseDescription', component: CourseDescriptionComponent },
9 | ];
10 |
11 |
```

# Criando as rotas do Projeto

- Execute a aplicação e verifique se está funcionando corretamente.

# Formatando a exibição dos dados

- Para formatar a forma como os dados serão exibidos na tela, Angular utiliza Pipes
- Pipes são usados para transformar e formatar seqüências de caracteres, valores de moeda, datas e outros dados de exibição.
- Pipes são funções simples que podem ser utilizadas nas expressões de modelo para aceitar um valor de entrada e retornar um valor transformado. Por exemplo para formatar uma data, um valor numérico etc



# Pipes

- A Angular fornece pipes internos para transformações de dados típicas, incluindo transformações para internacionalização
- Funções mais utilizadas:
  - DatePipe: formata um valor de data de acordo com as regras de localidade.
  - UpperCasePipe: transforma o texto em maiúsculas.
  - LowerCasePipe: transforma o texto em minúsculas.
  - CurrencyPipe: transforma um número em uma sequência de moedas, formatada de acordo com as regras de localidade.
  - DecimalPipe: transforma um número em uma sequência de caracteres com um ponto decimal, formatado de acordo com as regras de localidade.
  - PercentPipe: transforma um número em uma string de porcentagem, formatada de acordo com as regras de localidade.

# Pipes

- Exemplo no html:

```
<p>A Data é {{ aniversario | date }}</p>
```

- Exemplo no ts:

```
export class HeroBirthdayComponent {  
  aniversario = new Date(1988, 3, 15); // 15 de Abril de 1988  
}
```