iscte
INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

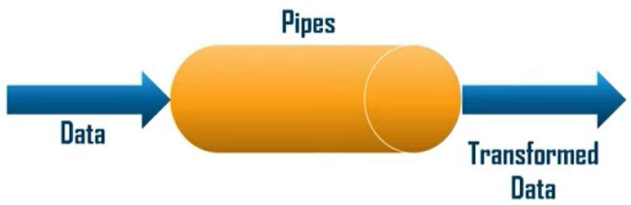
Módulo 9: Introdução ao Angular

Aula 05
Pipes e Navegação

UPskill

Pipes

- São funções simples que têm o objetivo de transformar a informação de visualização.
- O símbolo que define um pipe é a barra vertical “|”
- Apesar de podermos costumizar um pipe, o Angular traz consigo já pipes predefinidos para as transformações de dados mais comuns



Exemplos

Utilizando pipes transforme a seguinte informação:

A. Coloque uma palavra em MAIÚSCULAS e outra em minúsculas
B. Elimine as primeiras 3 letras da mesma palavra
C. Coloque um número decimal em percentagem
D. Transforme 5$ em 5EUR.
E. Escreva a data de hoje

```
<h1 [ngClass]="isGreen ? 'text-green' : 'text-red'">{{ text }}</h1>
```

```
<h2>{{ name | lowercase }}</h2>
<h2>{{ name | uppercase }}</h2>
<h2>{{ message | titlecase}}</h2>
<h2>{{ name | slice:3:5}}</h2>
<h2>{{ person | json }}</h2>
```

```
<h2>{{ 0.25 | percent }}</h2>

<h2>{{ 0.25 | currency }}</h2>
<h2>{{ 0.25 | currency: 'EUR': 'code' }}</h2>

<h2>{{ date }}</h2>
<h2>{{ date | date:'short' }}</h2>
<h2>{{ date | date:'shortDate' }}</h2>
<h2>{{ date | date:'shortTime' }}</h2>
```

```
<h2>{{5.678 | number:'1.2-3'}}</h2>
<h2>{{5.678 | number:'3.4-5'}}</h2>
<h2>{{5.678 | number:'3.1-2'}}</h2>
```

Routing

- São fundamentais para criarmos uma navegabilidade na nossa aplicação Web.
- Para termos esta funcionalidade disponível em Angular é obrigatório incluir o módulo routing no projeto.

```
ng generate module app-routing --flat --module=app
```

Routing

- Dentro do ficheiro **`app-routing.module.ts`** agora criado bastará associar dentro da sua lista **`Routes`** os diferentes caminhos para cada página.

Nota: Uma página deve também ser vista como um componente

Ex:

```
const routes: Routes = [
    { path: ‘ ’ , redirectTo: ‘/home’, pathMatch: ‘full’ }
    { path: ‘home’ , component: DasboardComponent }
    { path: ‘profile’ , component: ProfileComponent }
    { path: ‘**’ , component: PageNotFoundComponent }
]
```

Routing

- Configuração (ver projeto):
  - https://we.tl/t-FjH8DdQmP2

E se quiséssemos enviar informação para a página que estamos a abrir, como poderíamos fazê-lo?

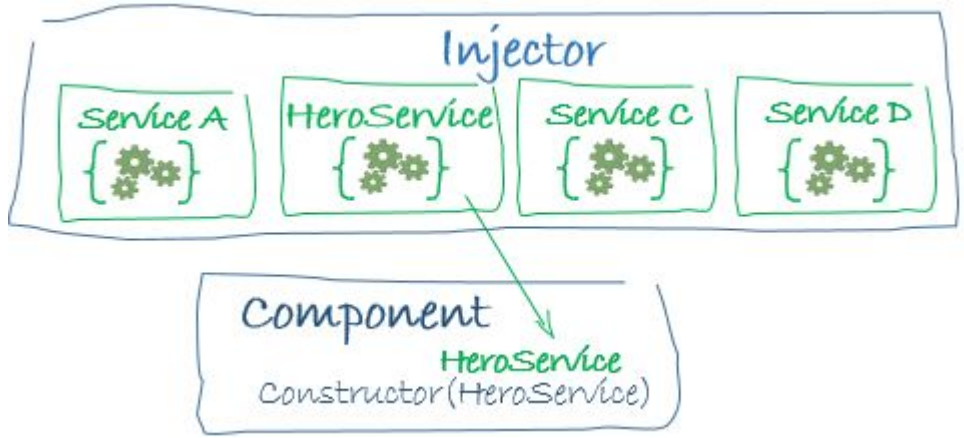
iscte INSTITUTO UNIVERSITÁRIO DE LISBOA emprego digital

Injeção de dependências

- Conceito fundamental do Angular que permite às classes configurar as dependências que necessitam.
- Existem duas funções principais no sistema de injeção de dependências:
    - Consumir Dependencias
    - Fornecer Dependências
- O Angular facilita a interação entre consumidores dependentes e fornecedores de dependência, utilizando uma abstração chamada **Injector**.
- Na maioria dos casos não é necessário criar injetores manualmente, mas é importante saber que existe uma camada que liga fornecedores e consumidores.

Injeção de dependências

- A forma mais comum de injectar uma dependência é **declará-la num construtor de classes**.
    - Quando o Angular cria uma nova instância de um componente, diretiva, ou classe de Pipe, ele determina quais os serviços ou outras dependências de que a classe necessita, analisando os tipos de parâmetros do construtor.



Ex:

```
constructor(private _router: Router) { }
```

Routing e Parâmetros

- Para termos acesso ao objeto numa nova página devemos passar a sua referência como parâmetro do URL e fazer uso da biblioteca @angular/router: **Router** e **ActivatedRoute**

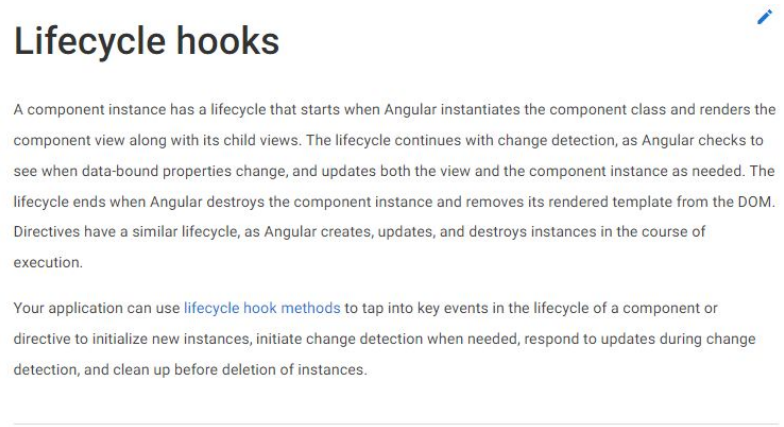
```
const routes: Routes = [
    { path: ‘ piada/:id ’ , component: PiadaDetailedComponent }
]
```

```
» this._router.navigate([‘/piada-detailed’ , piada.id])
```

```
» this.piadaID = parseInt(this._route.snapshot.paramMap.get(‘id’))
```

Lifecycle Hooks

- É a forma como a camada da lógica e a



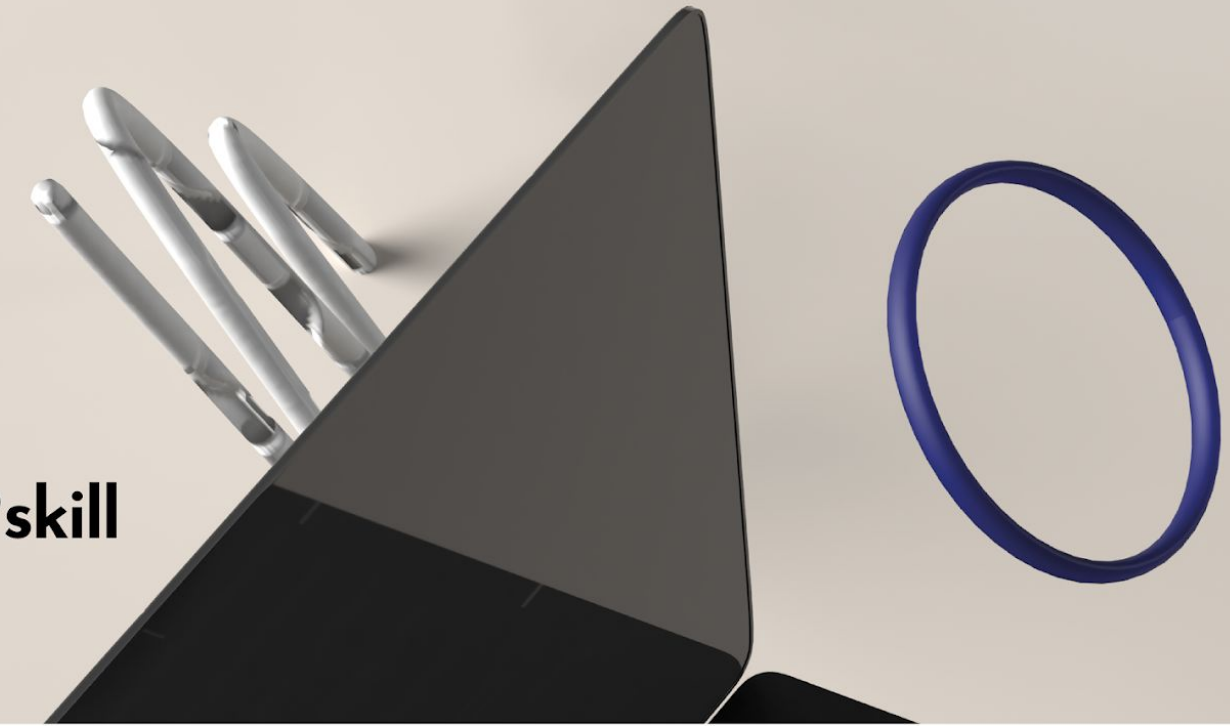
Lifecycle hooks

A component instance has a lifecycle that starts when Angular instantiates the component class and renders the component view along with its child views. The lifecycle continues with change detection, as Angular checks to see when data-bound properties change, and updates both the view and the component instance as needed. The lifecycle ends when Angular destroys the component instance and removes its rendered template from the DOM. Directives have a similar lifecycle, as Angular creates, updates, and destroys instances in the course of execution.

Your application can use lifecycle hook methods to tap into key events in the lifecycle of a component or directive to initialize new instances, initiate change detection when needed, respond to updates during change detection, and clean up before deletion of instances.

iscte INSTITUTO UNIVERSITÁRIO DE LISBOA
emprego digital

O futuro profissional começa aqui



iscte INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

UPskill