

## **Módulo 1: Introdução ao Vue.js**

- Vuejs filters, formatando textos

No Vue.js, os **filters** (filtros) são usados para formatar ou transformar dados antes de exibi-los no template. Eles funcionam em conjunto com a interpolação de dados, permitindo que você aplique transformações simples, como formatação de textos, números, datas, entre outros. Embora os filtros tenham sido depreciados nas versões mais recentes do Vue (a partir da versão 3), ainda são amplamente usados em projetos legados com Vue 2, onde foram bastante populares para manipular e formatar saídas de forma mais legível e reutilizável.

## Sintaxe Básica de Filtros

A forma mais comum de usar um filtro em Vue.js é aplicá-lo diretamente na interpolação de dados no template. A sintaxe é a seguinte:

```
{{ valor | nomeDoFiltro }}
```

- **valor**: O dado original que você deseja formatar.
- **| nomeDoFiltro**: O filtro que será aplicado ao valor para transformar ou formatá-lo antes de ser exibido.

## Criando um Filtro

No Vue 2, você pode registrar filtros de forma global ou local. Um filtro global pode ser utilizado em qualquer parte da aplicação, enquanto um filtro local só estará disponível no componente onde foi definido.

Exemplo de filtro global:

```
Vue.filter('capitalize', function(valor) {  
  if (!valor) return '';  
  valor = valor.toString();  
  return valor.charAt(0).toUpperCase() + valor.slice(1);  
});
```

Aqui, o filtro **capitalize** transforma o primeiro caractere de uma string em maiúsculo. Para usá-lo no template:

```
<p>{{ nome | capitalize }}</p>
```

Se a variável `nome` for "vuejs", após aplicar o filtro, será exibido "Vuejs".

**Exemplo de filtro local:**

```
new Vue({
  el: '#app',
  data: {
    nome: 'vuejs'
  },
  filters: {
    capitalize(valor) {
      if (!valor) return '';
      valor = valor.toString();
      return valor.charAt(0).toUpperCase() + valor.slice(1);
    }
  }
});
```

Nesse exemplo, o filtro `capitalize` só estará disponível para o componente onde foi definido.

## Filtros com Parâmetros

Você pode passar parâmetros adicionais para um filtro. Esses parâmetros são especificados após o nome do filtro, separados por dois pontos `:`.

**Exemplo:**

```
Vue.filter('truncate', function(valor, limite) {
  if (!valor) return '';
  valor = valor.toString();
  return valor.length > limite ? valor.substring(0, limite) + '...' : valor;
});
```

Esse filtro `truncate` irá cortar a string até o número de caracteres especificado no `limite`. Para usá-lo no template:

```
<p>{{ descricao | truncate(10) }}</p>
```

Aqui, o filtro exibirá apenas os primeiros 10 caracteres da string `descricao` e adicionará `"..."` ao final se a string for maior do que o limite.

## Exemplo Prático de Formatação de Texto

Vamos supor que você tem um nome de usuário que deseja exibir no formato capitalizado (com a primeira letra maiúscula) e também deseja truncar a bio do usuário se ela for muito longa.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div id="app">
    <h1>{{ usuario.nome | capitalize }}</h1>
    <p>{{ usuario.bio | truncate(30) }}</p>
  </div>
</body>
</html>
```

```
<script src="https://cdn.jsdelivr.net/npm/vue@2.7.14/dist/vue.js"></script>
<script>
  new Vue({
    el: '#app',
    data: {
      usuario: {
        nome: 'joão silva',
        bio: 'João é um desenvolvedor apaixonado por tecnologias de frontend e backend, com ampla experiência.'
      }
    },
    filters: {
      capitalize(valor) {
        if (!valor) return '';
        valor = valor.toString();
        return valor.charAt(0).toUpperCase() + valor.slice(1);
      },
      truncate(valor, limite) {
        if (!valor) return '';
        valor = valor.toString();
        return valor.length > limite ? valor.substring(0, limite) + '...' : valor;
      }
    }
  });
</script>
</body>
</html>
```

Neste exemplo:

- O filtro `capitalize` faz com que o nome do usuário (`joão silva`) seja exibido como `João silva`.

- O filtro `truncate(30)` corta a bio do usuário após 30 caracteres e adiciona `...` ao final, se necessário.

# João silva

João é um desenvolvedor apaixonado...

## Uso de Filtros em Pipes

Você pode encadear vários filtros juntos, onde a saída de um filtro será a entrada do próximo:

```
<p>{{ texto | capitalize | truncate(20) }}</p>
```

## Conclusão

Os filtros no Vue.js (especialmente no Vue 2) são uma maneira prática e fácil de formatar dados diretamente no template, permitindo transformar valores antes de exibí-los. Embora essa funcionalidade tenha sido descontinuada no Vue 3, ela ainda é amplamente usada em projetos legados. Para projetos novos, é recomendável usar `computed properties` ou `methods` para realizar essas transformações de dados, mantendo o código mais flexível e legível.

Quando formos estudar Vue 3 iremos ver como contornar essa situação.