

Introdução às tecnologias Web - ITW

Aula 10 – knockoutJS **Knockout**.

Sumário

Revisões – jQuery, JSON

A biblioteca knockoutJS

Integração com a biblioteca Bootstrap

Comparação knockoutJS vs jQuery

Utilização de JSON para suporte aos dados



Revisões:

O que é o jQuery

jQuery é uma biblioteca JavaScript multi-plataforma projetada para simplificar a programação (*scripting*) do lado do cliente de HTML.

A sintaxe do jQuery foi projetada para tornar mais fácil a navegação nos elementos de um documento. Exemplos:

- * selecionar elementos DOM
- * criar animações,
- * manipular eventos e
- * desenvolver aplicações Ajax.

Revisões:

Vantagens da utilização de jQuery

Separação entre o Javascript e o HTML

Ao invés de usar atributos HTML para identificar as funções para manipulação de eventos, o jQuery lida com eventos puramente em JavaScript. **Deste modo, as tags HTML e o código Javascript são completamente separados.**

Elimina incompatibilidades entre navegadores:

Os motores de Javascript dos diferentes navegadores diferem ligeiramente, de modo que o código Javascript que funciona para um navegador pode não funcionar em outro.

O jQuery lida com todas essas inconsistências entre browsers e fornece uma interface consistente que funciona nos diferentes navegadores.

Extensível:

O jQuery é muito extensível – através da adição de novas livrarias ao projeto.

Novos eventos, elementos e métodos podem ser facilmente adicionados e depois reutilizados como um plugin.

Revisões:

Sintaxe jQuery

A sintaxe jQuery foi feita a pensar especialmente na selecão de elemento(s) HTML e na execução de alguma ação sobre o(s) mesmo(s).

A sintaxe básica é: `$(selector).action()`

Um sinal \$ para definir / aceder à biblioteca jQuery

um (seletor) para "consultar/encontrar" elementos HTML no documento

Uma ação jQuery () a ser executada no(s) elemento(s)

Seletores:

`<form ...></form> → $("form")`

`id="myId" → $("#myId")`

`class="myClass" → $(".myClass")`

`<input name="myName"> → $("input[name*='Nam']")`

Revisões:

JSON - JavaScript Object Notation

JSON é um formato leve de armazenamento e intercâmbio de dados que é independente da linguagem de programação utilizada e é auto-descritivo, sendo, por isso, fácil de entender.

Usa a sintaxe JavaScript, mas o formato JSON é somente texto, por isso pode ser lido e usado como formato de dados por qualquer linguagem de programação.

Revisões:

JSON Objects & Arrays

Os objetos JSON são escritos dentro de chavetas {} e podem conter vários pares nome / valor, separados por vírgulas:

Saber!

```
{
  'name': 'Noé Elisabete Ferreiro',
  'email' : 'noe.ferreira@nowhere.com',
  'address' : 'Street name & number\nCounty\nState',
  'birthDate' : '1990/11/24',
  'sex' : 'Male',
  'course' : {
    'id' : 1234,
    'name' : 'Course name'
  }
}
```

Nota: Os valores do tipo texto são escritos entre aspas (simples '...' ou duplas "..."). Os valores lógicos ou numéricos são escritos diretamente.

Os objetos JSON podem ser agrupados em arrays que são escritos entre colchetes [] e separados por vírgulas:

```
"employees": [
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}]
```

Knockout.JS

Knockout.

Simplify dynamic JavaScript UIs with the Model-View-View Model (MVVM) pattern

Download
v3.5.1 - 25kb min+gz



[release notes](#)

Key concepts



Declarative Bindings

Easily associate DOM elements with model data using a concise, readable syntax



Automatic UI Refresh

When your data model's state changes, your UI updates automatically



Dependency Tracking

Implicitly set up chains of relationships between model data, to transform and combine it



Templating

Quickly generate sophisticated, nested UIs as a function of your model data

More features

- ✓ Free, open source ([MIT license](#))
- ✓ Pure JavaScript — works with any web framework
- ✓ Small & lightweight — 66kb minified
... reduces to 25kb when using HTTP compression

New: Interactive tutorials

Get started with knockout.js quickly, learning to build *single-page applications*, *custom bindings* and more with [these interactive tutorials](#).

A livraria KnockoutJS

Knockout é uma biblioteca JavaScript que ajuda a criar interfaces de utilizador de exibição e edição ricas e responsivas com um modelo de dados subjacente limpo.

Sempre que há seções da interface de utilizador que necessitam de atualização dinâmica (por exemplo, devido às ações do utilizador ou quando uma fonte de dados externa é alterada), o KO, acrónimo do Knockout, pode ajudar nessa implementação de forma mais simples e mais eficiente que utilizando apenas javascript ou mesmo jQuery.

A livraria KnockoutJS

Principais características:



Vinculações declarativas

Associa elementos do DOM a um modelo de dados através de uma sintaxe concisa e legível

Atualização automática da interface com o utilizador

Quando o estado do modelo de dados é alterado, a interface com o utilizador é atualizada automaticamente

Acompanhamento de dependências

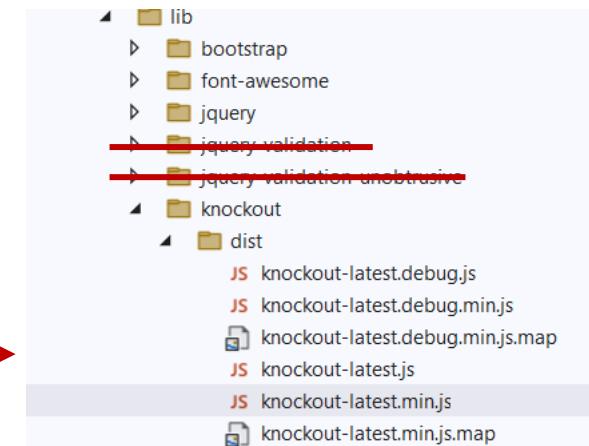
Implicitamente estabelece cadeias de relações entre os dados do modelo de modo a transformá-los e combiná-los

Templating

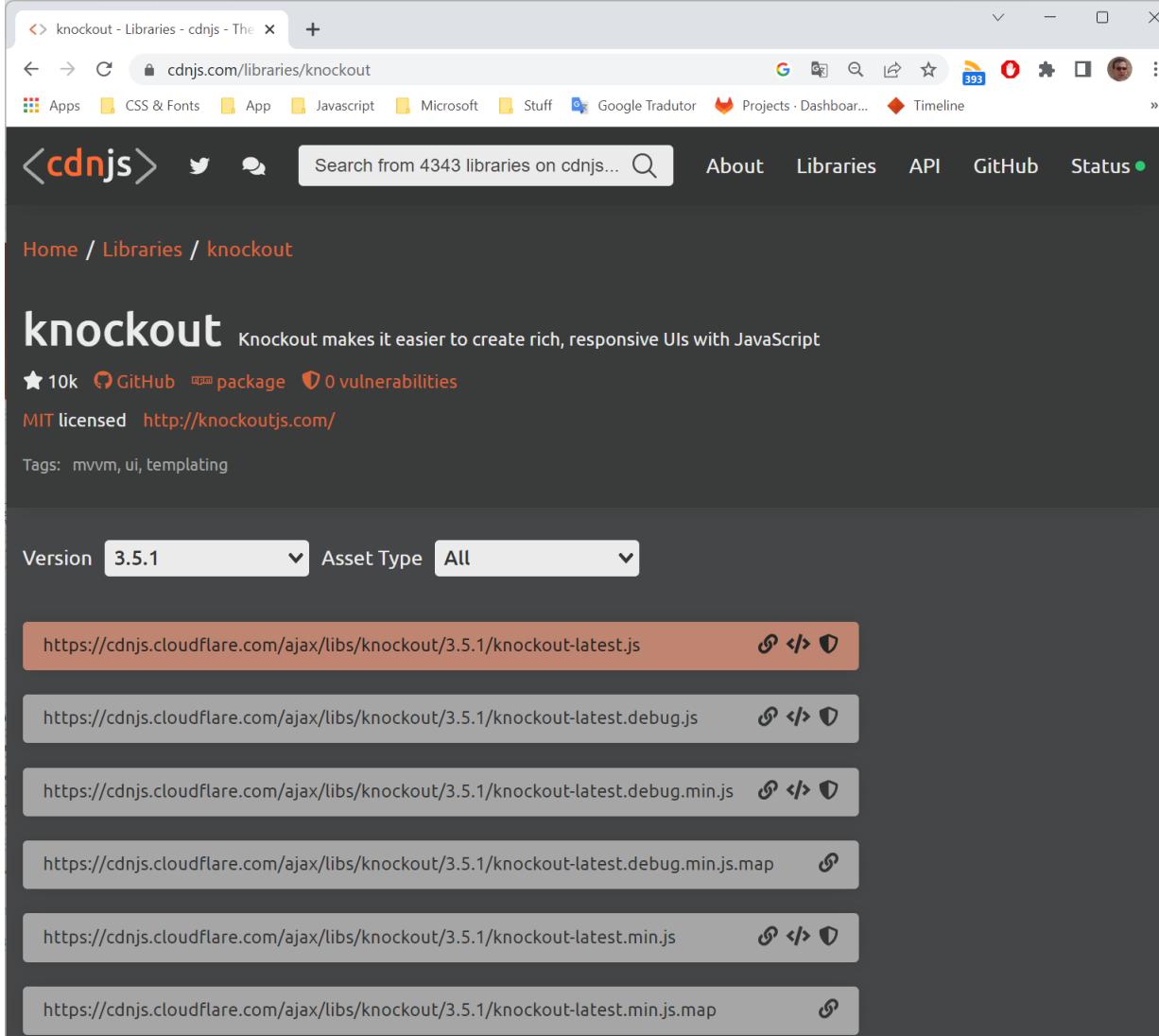
Gera rapidamente interfaces de utilizador sofisticadas como uma função dos dados do modelo

Instalação do knockout no Visual Studio

```
{  
  "version": "1.0",  
  "defaultProvider": "cdnjs",  
  "libraries": [  
    {  
      "destination": "wwwroot/lib/bootstrap/dist/",  
      "library": "bootstrap@5.2.2"  
    },  
    {  
      "destination": "wwwroot/lib/font-awesome/dist",  
      "library": "font-awesome@4.7.0"  
    },  
    {  
      "destination": "wwwroot/lib/jquery/dist/",  
      "library": "jquery@3.6.1"  
    },  
    {  
      "destination": "wwwroot/lib/knockout/dist",  
      "library": "knockout@3.5.1"  
    }  
  ]  
}
```



Instalação do knockout através de CDN



A livraria KnockoutJS

Outras características:

Livre, código aberto (licença MIT)

JavaScript puro - funciona com qualquer framework web

Sem dependências

Pequeno e leve - 67kb minified (@3.5.1 – 11/05/2019)

Suporta todos os navegadores habituais, mesmo os antigos
IE 6+, Firefox 3.5+, Chrome, Opera, Safari (desktop / mobile)

Totalmente documentado

Há documentos da API, exemplos e tutoriais interativos (até livros!)

Como usar o knockout? (1)

Para criar um viewmodel com KO, basta declarar qualquer objeto JavaScript (JSON). Por exemplo:

```
var myViewModel = {  
    personName: 'Zé Maria',  
    personAge: 45  
};
```

Pode criar-se uma view deste viewmodel usando uma vinculação declarativa.

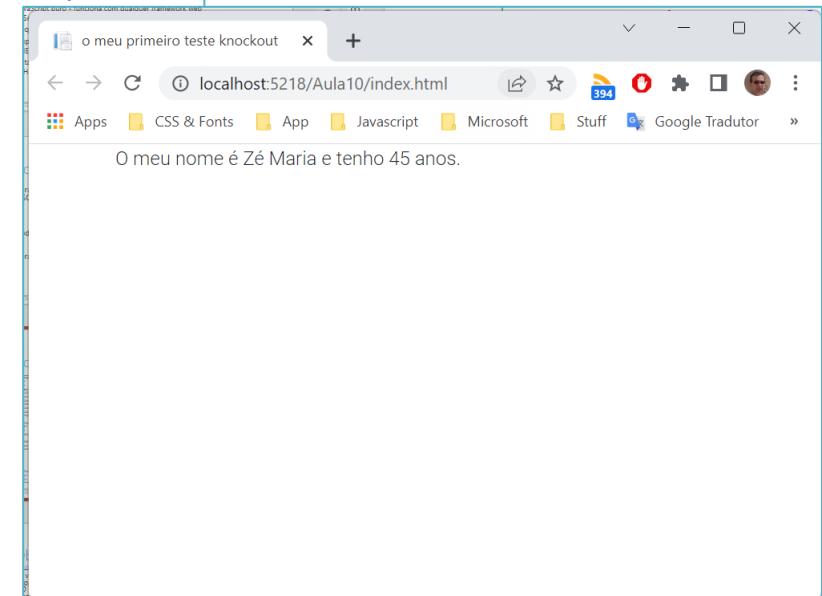
```
O meu nome é <span data-bind="text: personName"></span>
```

Para que tudo funcione, é preciso ativar o knockout:

```
ko.applyBindings(myViewModel);
```

Como usar o knockout? (2)

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>o meu primeiro teste knockout</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href='https://fonts.googleapis.com/css?family=Roboto:300' rel='stylesheet' type='text/css'>
    <link href="../../lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />
    <link href="../../lib/font-awesome/dist/css/font-awesome.min.css" rel="stylesheet" />
    <style>
        body, * { font-family: 'Roboto', sans-serif; }
    </style>
</head>
<body>
    <div class="container">
        O meu nome é <span data-bind="text: personName"></span>
        e tenho <span data-bind="text: personAge"></span> anos.
    </div>
    <script src="../../lib/knockout/dist/knockout-latest.min.js"></script>
    <script>
        var myViewModel = {
            personName: 'Zé Maria',
            personAge: 45
        };
        ko.applyBindings(myViewModel);
    </script>
</body>
</html>
```



Observáveis e dependências (ko.observable())(1)

<http://knockoutjs.com/documentation/observables.html>

Já vimos como criar um viewmodel básico e como exibir uma das suas propriedades (text) usando uma ligação mas um dos principais benefícios do KO é que ele atualiza a interface (view) do utilizador automaticamente quando o viewmodel muda.

Pergunta: Como é que o KO pode saber quando as partes do viewmodel mudam?

Resposta: é preciso declarar as propriedades do seu modelo como observáveis!

Os observáveis são objetos JavaScript especiais que podem notificar os assinantes sobre as alterações e podem detectar dependências automaticamente.

Observáveis e dependências (`ko.observable()`)(2)

Para utilizar variáveis observáveis, reescreve-se o.viewmodel anterior da seguinte maneira:

```
var myViewModel = {  
    personName: ko.observable('Zé Maria'),  
    personAge: ko.observable(45)  
};
```



Não é preciso alterar a view - a sintaxe de ligação de dados é a mesma.

A diferença é que agora a view é capaz de detectar alterações da.viewmodel e, quando isso acontecer, atualizará a informação na view automaticamente.

Observáveis e dependências (`ko.observable()`)(3)

Problema: Nem todos os browser suportam operações de leitura (get) e escrita (set) de JavaScript (incompatibilidades entre implementações do JavaScript), portanto, por questões de compatibilidade, os objetos `ko.observable` são funções.

- Para ler o valor atual do observável, basta chamar o observável sem parâmetros.

Do exemplo, `myViewModel.personName()` retornará 'Zé Maria' e `myViewModel.personAge()` retornará 45.

- Para escrever um novo valor no observável, invoca-se o observável e passa-se o novo valor como parâmetro.

Por exemplo, `myViewModel.personName('Maria')` irá alterar o valor de nome para 'Maria'.

Arrays de observáveis (`ko.observableArray([])`)

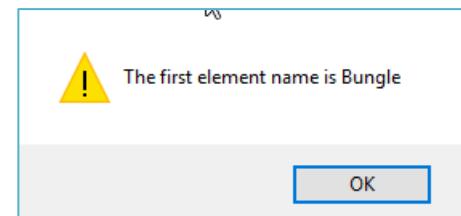
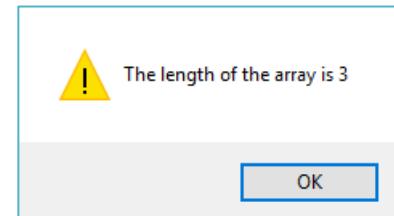
<http://knockoutjs.com/documentation/observableArrays.html>

Já vimos que, caso se pretenda detectar e responder a alterações num objeto, usamos observáveis. 🤔

Se pretendermos detectar e responder a alterações **numa coleção de objetos**, deveremos utilizar um `observableArray`.

Esta possibilidade é particularmente útil em cenários em que se exibem ou editam vários valores e são necessárias seções repetidas da interface para fazer aparecer e desaparecer à medida que os itens são adicionados e/ou removidos.

```
// This observable array initially contains three objects
var myObservableArray = ko.observableArray([
    { name: "Bungle", type: "Bear" },
    { name: "George", type: "Hippo" },
    { name: "Zippy", type: "Unknown" }
]);
alert('The length of the array is ' + myObservableArray().length);
alert('The first element name is ' + myObservableArray()[0].name);
```



Observáveis calculados(ko.computed)

Suponha que já tem um observável para firstName, e outro para lastName, e deseja exibir o nome completo?

É aí que os **observáveis calculados** são úteis - são funções que dependem de um ou mais observáveis e serão atualizados automaticamente sempre que alguma das suas dependências mudarem.

```
O meu nome é <span data-bind="text: fullName"></span>  
  
function AppViewModel() {  
    var self = this;  
  
    self.firstName = ko.observable('Bob');  
    self.lastName = ko.observable('Smith');  
    self.fullName = ko.computed(function () {  
        return self.firstName() + " " + self.lastName();  
    });  
}
```

! Sem os espacos (" ")

KO bindings (1)

text: ~
`text()` – o binding com `text()` faz com que o elemento DOM associado exiba o valor de texto do seu parâmetro.

Normalmente, esta propriedade é útil com elementos que tradicionalmente exibem texto, como por exemplo o `` ou o ``, mas tecnicamente pode usá-lo com qualquer elemento.

html: ~
`html()` – o binding com `html()` faz com que o elemento DOM associado exiba o html do seu parâmetro.

Normalmente, isso é útil quando os valores no viewmodel são sequências de marcação HTML.

KO bindings (2)

css:

css() – o binding css adiciona ou remove uma ou mais classes CSS ao elemento DOM associado.

(Nota: Se não quiser aplicar uma classe CSS, mas preferir atribuir um valor de atributo de estilo diretamente, consulte o binding style.)

```
<div data-bind="css: profitStatus">Profit Information</div>
```

style:

style() – o binding style adiciona ou remove um ou mais valores de estilo ao elemento DOM associado.

```
<div data-bind="style: { color: currentProfit() < 0 ? 'red' : 'black' }>Profit Information</div>
```

Não esquecer !!!

► Neste caso, se o número for menor que 0 a cor fica vermelha, caso contrário a cor fica preta

KO bindings (3)

`attr()` – O binding attr fornece uma maneira genérica de definir o valor de qualquer atributo para o elemento DOM associado.

Isso é útil, por exemplo, quando precisa definir o atributo de título de um elemento, o `src` de uma tag `img` ou o `href` de um link com base em valores no seu viewmodel, com o valor do atributo sendo atualizado automaticamente sempre que a propriedade correspondente no viewmodel muda.

```
<a data-bind="attr: { href: url, title: details }">Relatório</a>

<script type="text/javascript">
    var viewModel = {
        url: ko.observable("http://somesite.com/yearReport.html"),
        details: ko.observable("Relatório e contas referente ao corrente ano")
    };
</script>
```

KO bindings (4)

`visible()` – permite fazer o binding da propriedade visível a um elemento Dom que ficará visivel sempre que a variável de controlo do.viewmodel tomar um valor true.

KO – controlo de fluxo

<http://knockoutjs.com/documentation/foreach-binding.html>

"Corre todos os valores na data"

- ! **foreach()** – o binding **foreach** duplica a “lista de entidades” para cada entrada numa matriz e vincula cada cópia dessa lista ao item de matriz correspondente.

Isso é especialmente útil para representar listas ou tabelas.

Assumindo que a matriz é um array de observáveis, sempre que se adicionar, remover, alterar ou reordenar as entradas da matriz, a ligação atualizará eficientemente a UI mantendo o sincronismo entre elas – inserindo ou removendo elementos da lista ou reordenando os elementos existentes, sem afetar quaisquer outros elementos DOM.

Pode encapsular-se qualquer número de bindings **foreach** e utilizá-los em conjunto com outras ligações de controle-fluxo, como **if** ou **with**.

Próximos slides

Exemplo de binding com foreach

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>knockout exemplo foreach</title>
    <link href="../lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <table class="table table-striped table-condensed">
        <thead>
            <tr><th>First name</th><th>Last name</th></tr>
        </thead>
        <tbody data-bind="foreach: people">
            <tr>
                <td data-bind="text: firstName"></td>
                <td data-bind="text: lastName"></td>
            </tr>
        </tbody>
    </table>
    <script src="../lib/knockout/dist/knockout-latest.min.js"></script>
    <script type="text/javascript">
        ko.applyBindings({
            people: [
                { firstName: 'Bert', lastName: 'Bertington' },
                { firstName: 'Charles', lastName: 'Charlesforth' },
                { firstName: 'Denise', lastName: 'Dentiste' }
            ]
        });
    </script>
</body>
</html>
```

View model simple

Super útil!

| First name | Last name |
|------------|--------------|
| Bert | Bertington |
| Charles | Charlesforth |
| Denise | Dentiste |

KO – controlo de fluxo

<http://knockoutjs.com/documentation/if-binding.html>

<http://knockoutjs.com/documentation/ifnot-binding.html>

<http://knockoutjs.com/documentation/with-binding.html>

if() – o binding **if** faz com que uma seção de marcação apareça no documento somente se a variável de controlo especificada for avaliada como verdadeira.

ifnot() – é igual ao binding **if** somente inverte o valor da expressão de avaliação especificada – isto porque não existe um “else binding”

with() - o binding com **with** cria um novo contexto de vinculação, de modo que os elementos descendentes são vinculados no contexto de um objeto especificado.

Exemplo de binding com with

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>knockout Exemplo with</title>
  <link href="../../lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
  <h1 data-bind="text: city"> </h1>
  <p data-bind="with: coord">
    Latitude(°): <span data-bind="text: lat"> </span>,
    Longitude(°): <span data-bind="text: lon"> </span>
  </p>
  <script src="../../lib/knockout/dist/knockout-latest.min.js"></script>
  <script>
    ko.applyBindings({
      city: "London",
      coord: {
        lat: 51.5001524,
        lon: -0.1262362
      }
    });
  </script>
</body>
</html>
```

Usando o jQuery no exemplo da aula anterior...

```
$("#coordinates").html('Lon (°): ' + data.coord.lon + ' |  
Lat (°): ' + data.coord.lat);
```

London

Latitude: 51.5001524, Longitude: -0.1262362

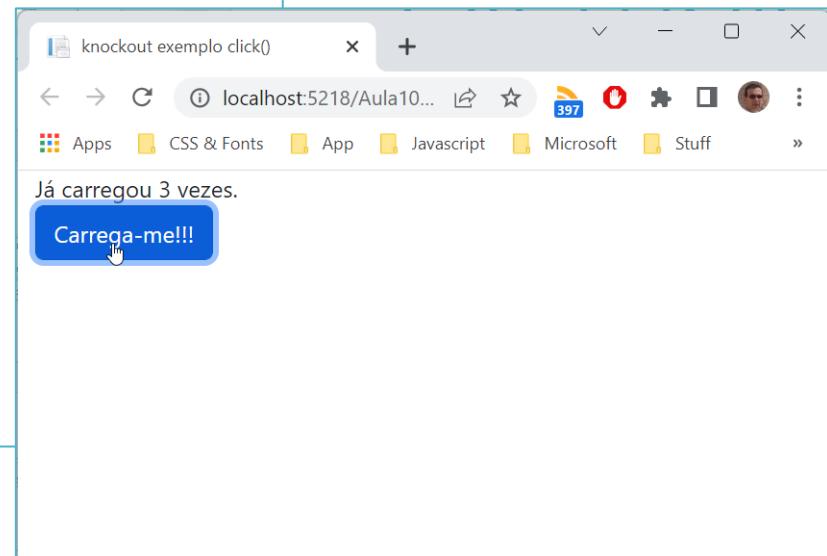
KO – binding eventos

click() – O binding do evento `click` permite associar um gestor de eventos cuja função JavaScript é chamada quando o elemento DOM associado for clicado.

Isso é mais comumente usado com elementos como botões, input e hiperligações, mas na verdade funciona com qualquer elemento DOM visível.

Exemplo de binding do evento click

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>knockout exemplo click()</title>
    <link href="../../lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <div class="container">
        Já carregou...<span data-bind="text: numberOfClicks"></span> vezes.<br />
        <button data-bind="click: incrementClickCounter" class="btn btn-primary">Carrega-me!!!</button>
    </div>
    <script src="../../lib/knockout/dist/knockout-latest.min.js"></script>
    <script>
        var viewModel = {
            numberOfClicks: ko.observable(0),
            incrementClickCounter: function () {
                var previousCount = this.numberOfClicks();
                this.numberOfClicks(previousCount + 1);
            }
        };
        ko.applyBindings(viewModel);
    </script>
</body>
</html>
```

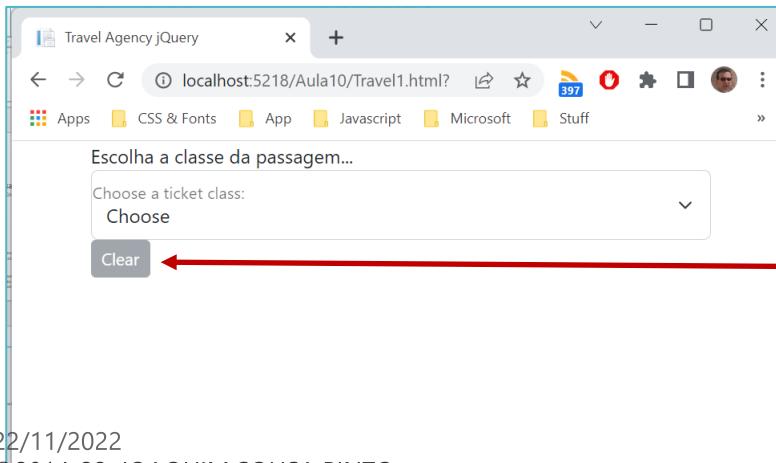


Desafio:

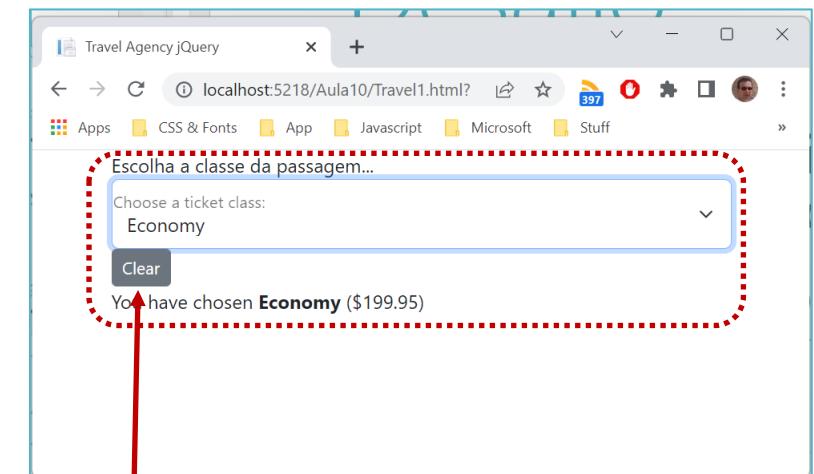
Fazer um formulário para a gestão da classe de uma passagem de avião e do seu respetivo preço – Cenário 1: usando jQuery; Cenário 2 : usando Knockout.

Dados para controlo do formulário:

```
tickets = [  
    { name: "Economy", price: 199.95 },  
    { name: "Business", price: 449.22 },  
    { name: "First Class", price: 1199.99 }  
];
```



Enquanto não há uma escolha, o botão está desativado



Quando há uma escolha, o botão fica ativo e é apresentada uma mensagem com a classe escolhida e o preço.

Cenário 1: usando jQuery

travel1.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Travel Agency jQuery</title>
    <link href="../lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <div class="container">
        <div class="page-header">Escolha a classe da passagem...</div>
        <form class="row">
            <div class="form-floating">
                <select id="flightClasses" class="form-select"></select>
                <label for="flightClasses" class="control-label">Choose a ticket class:</label>
            </div>
            <div class="form-floating">
                <button id="clearBtn" class="btn btn-sm btn-secondary">Clear</button>
            </div>
            <div class="form-floating">
                <p id="chooseTicket" class="form-control-static">
                    You have chosen <b id="chooseClass"></b>
                    ($<span id="choosePrice"></span>)
                </p>
            </div>
        </form>
    </div>
    <script src="../lib/jquery/dist/jquery.min.js"></script>
    <script src="travel1.js"></script>
</body>
</html>

```

Cenário 1: usando jQuery (2)

travel1.js

```

$(document).ready(function () {
    tickets = [
        { name: "Economy", price: 199.95 },
        { name: "Business", price: 449.22 },
        { name: "First Class", price: 1199.99 }
    ];
    console.log("document ready");
    //--- Inicialização dos elementos html
    console.log("adding <select> options")
    //--- Lista de opções - elemento em branco (a pedir para selecionar ...)
    $('#flightClasses').append($('<option>', {
        value: '',
        text: 'Choose'
    }));
    //--- Lista de opções - inicialização dos elementos da lista
    $.each(tickets, function (i, ticket) {
        $('#flightClasses').append($('<option>', {
            value: ticket.price,
            text: ticket.name
        }));
    });
    //--- Disable do botão
    $("#clearBtn").prop("disabled", true);
    //--- Esconder a mensagem
    $("#chooseTicket").addClass("d-none");
}

//--- Inicialização terminada.
//--- Gestão de eventos ...
$("#flightClasses").change(function () {
    if ($("#flightClasses").val() == "") {
        //--- Disable do botão
        $("#clearBtn").prop("disabled", true);
        //--- Esconder a mensagem
        $("#chooseTicket").addClass("d-none");
    }
    else {
        //--- Enable do botão
        $("#clearBtn").prop("disabled", false);
        //--- Mostrar a mensagem
        $("#chooseTicket").removeClass("d-none");
        $("#chooseClass").text($("#flightClasses
option:selected").text());
        $("#choosePrice").text($("#flightClasses").val());
    }
});
}
)

```

Cenário 2: usando knockout.js

travel2.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Travel Agency jQuery</title>
    <link href="../lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />
</head>
<body>
    <div class="container">
        <div class="page-header">Escolha a classe da passagem...</div>
        <form class="row">
            <div class="form-floating">
                <select data-bind="options: tickets,
                                optionsCaption: 'Choose...',
                                optionsText: 'name',
                                value: chosenTicket" class="form-control"></select>
                <label for="flightClasses" class="control-label">Choose a ticket class:</label>
            </div>
            <div class="form-floating">
                <button data-bind="enable: chosenTicket,
                                click: resetTicket" class="btn btn-sm btn-secondary">Clear</button>
            </div>
            <div class="form-floating">
                <p data-bind="with: chosenTicket" class="form-control-static">
                    You have chosen <b data-bind="text: name"></b>
                    ($<span data-bind="text: price"></span>)
                </p>
            </div>
        </form>
    </div>
    <script src="../lib/knockout/dist/knockout-latest.min.js"></script>
    <script src="travel2.js"></script>
</body>
</html>

```

Cenário 2: usando knockoutJs(2)

travel2.js

```
function TicketsViewModel() {
    this.tickets = [
        { name: "Economy", price: 199.95 },
        { name: "Business", price: 449.22 },
        { name: "First Class", price: 1199.99 }
    ];
    this.chosenTicket = ko.observable();
    this.resetTicket = function () { this.chosenTicket(null) }
}
ko.applyBindings(new TicketsViewModel());
```

Só isto ... e mais nada. Descodificando...

A variável `this.chosenTicket`

fica com o valor escolhido na interface pelo `<select></select>` através da propriedade `value: chosenTicket`
 O `<button></button>` é controlado também por este valor através da propriedade `enable: chosenTicket`

A função `this.resetTicket`

é atuada na interface pelo `<button></button>` ativa no código o método `click: resetTicket"` que coloca o valor da variável `this.chosenTicket` em `null`
 em consequência dessa alteração na parte do código, na interface, o `<select></select>`, o `<button></button>` e o `<p></p>` são alterados

Bibliografia

knockoutjs.com, "Knockout ", em linha : <http://knockoutjs.com/index.html>, visitado em 13/12/2020

1-16 of 52 results for "knockoutjs"

Sort by: Featured

Department

- Movies & TV
- Movies
- Movies & Films
- TV
- TV Shows
- Books
- Home & Kitchen
- Sports & Fitness
- Kindle Store
- Kindle eBooks
- See All 9 Departments

Avg. Customer Review

- ★★★★★ & Up
- ★★★★☆ & Up
- ★★★☆☆ & Up
- ★★☆☆☆ & Up

TV Show

- Comanche Moon
- Nora Roberts
- The Wind in the Willows
- WrestleMania
- Elmo
- Saddle Club
- Wolverine and the X-Men
- See more

Actor

- Sacha Baron Cohen
- Alan Rickman
- Emma Watson
- Robbie Coltrane
- Daniel Radcliffe
- Rupert Grint
- Julie Walters
- Dame Maggie Smith
- See more

Director

- Jason Woliner
- Larry Charles
- Chris Columbus
- Tim Burton
- David Yates
- Bill Melendez
- Peter Jackson
- See more

Price

- Free
- \$0 to \$1
- \$1 to \$3

**Mastering KnockoutJS**

by Timothy Moran | Nov 26, 2014

7

Kindle\$29⁹⁹ \$49.99

Available instantly

Paperback\$49⁹⁹

Ships to Portugal

New and used options from
\$35.80**Knockout.js: Building Dynamic Client-Side Web Applications**

by Jamie Munro | Jan 3, 2015

11

Paperback\$19⁹⁹

Ships to Portugal

New and used options from
\$3.97**Kindle**\$16⁹⁹ \$19.99

Available instantly

**Fundamentals and Essentials of KnockoutJS: Standalone JavaScript implementation of the Model-View-ViewModel pattern with templates**

by Kenald Primewood Publications | Dec 8, 2020

Kindle Edition\$0⁰⁰

Free with Kindle Unlimited membership Learn More

Or \$9.99 to buy

**Deep Dive into ASP.NET MVC 5 with Bootstrap and Knockout.js: Best Practices, Optimization Techniques, Tips & Tricks from Real life Projects**

by Kragoselt Publications | Nov 30, 2020

Kindle Edition\$0⁰⁰

Free with Kindle Unlimited membership Learn More

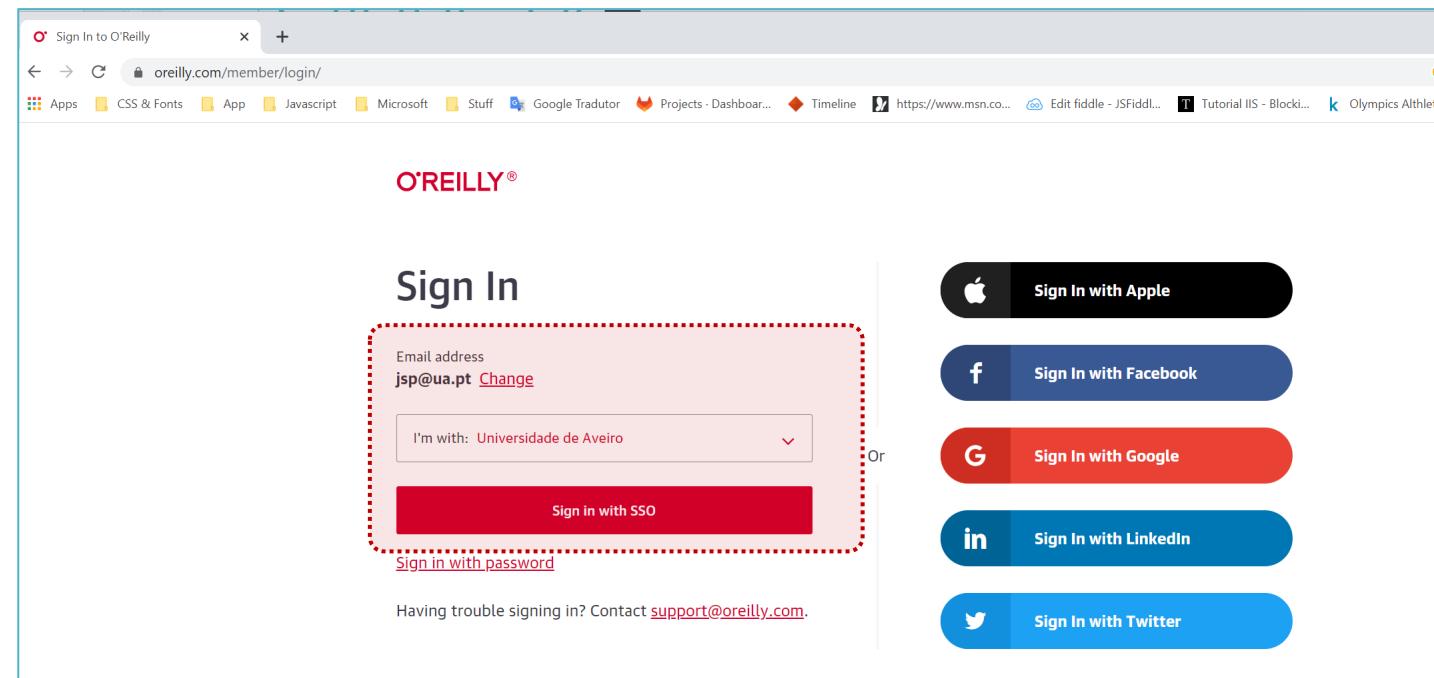
Or \$9.99 to buy

Notícias de 14-12-20:

<https://www.ua.pt/pt/noticias/8/65113>.

Alunos, docentes, investigadores e pessoal técnico, auxiliar e de gestão passam a ter acesso integral aos conteúdos da O'Reilly, mediante autenticação e através da página principal (login) ou, diretamente, através do seguinte link:

<https://go.oreilly.com/universidade-de-aveiro>



Search +

learning.oreilly.com/search/?query=knockoutJS&extended_publisher_data=true&highlight=true&include_assessments=false&include_case_s... Apps CSS & Fonts App Javascript Microsoft Stuff Google Tradutor Projects · Dashboard Timeline https://www.msn.co... Edit fiddle - JSFiddle Tutorial IIS - Blocki...

knockoutJS

1 - 10 of 147 search results for "knockoutJS"

All Books Videos Courses Playlists Topics Sort By Relevance

 **BOOK**
Knockout.js
By Jamie Munro
KNOCKOUT.JS
★★★★★ 7 reviews
O'Reilly Media, Inc. December 2014
Use **Knockout.js** to design and build dynamic client-side web applications that are extremely responsive and easy to maintain. This example-driven book shows you how to use this lightweight JavaScript framework and its Model-View-ViewModel (MVVM) pattern.

 **BOOK**
KnockoutJS Essentials
By Jorge Ferrando
KNOCKOUT.JS
★★★★☆ 1 review
Packt Publishing February 2015
Implement a successful JavaScript-rich application with **KnockoutJS**, jQuery, and Bootstrap In Detail **KnockoutJS** is a Model View ViewModel (MVVM) framework that allows you to simplify the design of complex user. ... **KnockoutJS** Essentials gives you an insight into the best practices when designing and building...

Search

learning.oreilly.com/search/?query=bootstrap%205&extended_publisher_data=true&highlight=true&include_assessments=false&include_ca... 398

Apps CSS & Fonts App Javascript Microsoft Stuff Google Tradutor Projects · Dashboard Timeline https://www.msn.co... Edit fiddle - JSFiddle Tutorial IIS - Blocki...

bootstrap 5

1 - 10 of 5732 search results for "bootstrap 5"

All Books Videos Courses Learning Paths Audiobooks

Topics Sort By Relevance

 **BOOK**
The Missing Bootstrap 5 Guide
By [Jeppe Schaumburg Jensen](#)
BOOTSTRAP

Write the first review
Packt Publishing August 2022

The Missing **Bootstrap 5** Guide will help you customize **Bootstrap 5** with Sass to achieve a layout that stands out from the crowd, enabling you to create something unique that doesn't look like it was created with **Bootstrap**. Make your website stand out by unlocking the full potential of **Bootstrap 5** and creating effi...

 **COURSE**
Bootstrap 5 Course: Build Responsive Websites Like a Pro
By [Meta Brains](#)
BOOTSTRAP

Write the first review
Packt Publishing June 2022

After that, you will learn about the **Bootstrap 5** Grid system, look at the different breakpoints in **Bootstrap 5**, and learn about **Bootstrap 5** components such as Badge, Breadcrumb, Buttons, Button group, Card, and much more. **Bootstrap 5**, the most recent version of the world's most popular front-end framework, wil...

Search x +

learning.oreilly.com/search/?query=javascript&extended_publisher_data=true&highlight=true&include_assessments=false&include_case_stu... 399

Apps CSS & Fonts App Javascript Microsoft Stuff Google Tradutor Projects · Dashboard Timeline https://www.msn.co... Edit fiddle - JSFiddle Tutorial IIS - Block...

javascript

1 - 10 of 19564 search results for "javascript"

All Books Videos Courses Learning Paths Playlists Audiobooks

Topics Sort By Relevance

 BOOK
JavaScript: The Definitive Guide, 7th Edition
By David Flanagan
JAVASCRIPT

★★★★★ 35 reviews
O'Reilly Media, Inc. May 2020

JavaScript is the programming language of the web and is used by more software developers today than any other programming language. For nearly 25 years this best seller has been the go-to guide for **JavaScript** programmers. ... This book is for programmers who want to learn **JavaScript** and for web developers...

 PLAYLIST
JavaScript Essentials
By Elisabeth Robson
JAVASCRIPT

March 2020

JavaScript is a key component of the modern web. Elisabeth Robson's picks help you build a solid understanding of JS and apply it to your projects.

Search + ×

learning.oreilly.com/search/?query=jquery&extended_publisher_data=true&highlight=true&include_assessments=false&include_case_studies=false

Apps CSS & Fonts App Javascript Microsoft Stuff Google Tradutor Projects - Dashboard Timeline https://www.msn.co... Edit fiddle - JSFiddle Tutorial IIS - Blocker

jquery

1 - 10 of 2937 search results for "jquery"

All Books Videos Courses Learning Paths Playlists Audiobooks Topics Sort By Relevance

 BOOK
jQuery Recipes: Find Ready-Made Solutions to All Your jQuery Problems
By [Bintu Harwani](#)
JQUERY

Write the first review
Apress October 2021

It is an invaluable reference to keep by your side as you progress in your **jQuery** development. Start becoming a **jQuery** pro with **jQuery** Recipes today. What You'll Learn Understand the basics of **jQuery** and how this library works Learn how to traverse and manipulate the DOM to your own requirements Explore...

 COURSE
The Complete jQuery Course - Beginner to Professional
By [Laurence Svekis](#)
JQUERY

Write the first review
Packt Publishing June 2022

You will see class and styling with **jQuery**, forms with **jQuery**, and look at **jQuery** page animations. Finally, you will see AJAX and JSON integrating with **jQuery**. By the end of this course, you will have learned about **jQuery** from scratch and be confident to start creating amazing web applications using **jQuery**. Best of al...