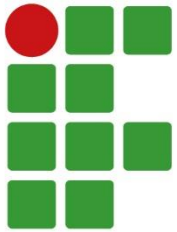


INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Aula 06

jQuery

Prof. Tiago Henrique Trojahn

tiagotrojahn@ifsp.edu.br

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

O que é?

- “**jQuery** é uma Biblioteca JavaScript que visa a simplificação na manipulação do DOM , nas chamadas de AJAX e no gerenciamento de Eventos . É muito utilizado por desenvolvedores de Javascript.” – MDN, 2019

- Para usá-lo, basta adicionar o seguinte comando no HEAD de seu site

```
<script type="text/javascript" src="https://code.jquery.com/jquery-3.4.1.slim.min.js">
</script>
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Princípio básico

- Basicamente, cada comando jQuery possui o seguinte formato:

```
$(seletor).ação();
```

- Onde o **seletor** indica QUAL elemento HTML/CSS se refere e a **ação** refere-se a O QUE deve acontecer.

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Seletores

- O jQuery suporta uma ampla gama de seletores, similar ao CSS em si
- Exemplos:

```
$("div"); // Todos os divs da página
$(".azul"); // Todos os elementos com classe 'azul'
$("#primeiro"); // Elemento com id 'primeiro'
$("div p"); // Todos os 'p' filhos de algum 'div'
$("div[id]"); // Todos os 'div' que possuam o atributo 'id'
$("[name=teste]"); // Todos os elementos com a atributo 'name' que tenha valor 'teste'
$("div p:first-child"); // Todos os primeiros 'p' que sejam filhos de um 'div'
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Window.onload?

- O *window.onload,* comando padrão do JavaScript, é capaz de forçar o navegador a executar um trecho de código após a página estar carregada.

- Porém, tal função possui seus vícios:
  - A função é executada apenas após o conteúdo estiver totalmente carregado
    - O que acontece se a página “demorar” pra estar pronta?
  - Cada navegador pode implementar a operação da forma que quiser.
  - Em alguns casos, window.onload é desnecessariamente lento...

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Window.onload?

```
$(document).ready(function() {
    /* Código a ser executado após
       o carregamento da página
    */
});
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Código base

```
<!DOCTYPE html>
<html lang='pt-br'>
<head>
    <meta charset='utf-8'>
    <title>Título</title>
    <script type="text/javascript" src="https://code.jquery.com/jquery-3.4.1.slim.min.js"></script>
    <script>
        $(document).ready(function () {
            /* Código a ser executado após
                o carregamento da página
            */
        });
    </script>
</head>

<body>
</body>
</html>
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Eventos

- O JavaScript suporta a especificação de eventos sobre os elementos

```
$("#submeter").click(function(e) {

});

$("body").on("click", "#submeter", function(e) {

});
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Eventos

- Todos os eventos são suportados...

```
$("#submeter").mouseover(function(e) {...});
$("#submeter").dblclick(function(e) {...});
$("#submeter").keypress(function(e) {...});
$("#submeter").mouseenter(function(e) {...});
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Eventos

- E como identificar qual o elemento específico causou o “evento” executado?

```
$("button").click(function(e) {
    ???
});
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Eventos

- Temos duas opções!
  - Usando JavaScript puro...

```
$("button").click(function(e) {
    e.target; // Objeto que causou o evento...
});
```

  - Ou jQuery

```
$("button").click(function(e) {
    $(this); // Objeto que causou o evento...
});
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Exercício 1

- Considere o seguinte comando em jQuery, capaz de ocultar um elemento HTML

```
$(seletor).hide();
```

- Dado o código HTML de exemplo, faça com que as células que sejam clicadas “desapareçam” da página.
  - Use apenas uma única captura de evento.

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Exercício 1 - Resolução

```
$(document).ready(function() {
    $("tbody td").click(function(e) {
        $(this).hide();
    });
});
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Caminhar na árvore DOM

```
$("p").not("#ifsp"); // Seleciona todos os 'p', exceto id='ifsp'
$("p").eq(3); // Seleciona o 'p' com índice
$("p").eq(-2); // Seleciona o penúltimo 'p'
$("p").first(); // Seleciona o primeiro 'p'
$("p").last(); // Seleciona o último 'p'
$("p").children(); // Seleciona todos os filhos de todos 'p'
$("p").next(); // Seleciona os próximos irmãos de 'p'
$("p").prev(); // Seleciona irmãos anteriores de 'p'
$("p").parent(); // Seleciona o pai de todos os 'p'
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Caminhar na árvore DOM

- Métodos como **children**, **next** e **prev** aceitam um parâmetro usado como filtro.

```
// Seleciona todos os filhos de 'p' que tenham a classe ’azul’
$("p").children(".azul");
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Modificando elementos DOM

```
// Retorna o conteúdo html do elemento com id='teste' selecionado
var conteudo = $("#teste").html();
// Modifica o conteúdo html do elemento selecionado (id='teste')
$("#teste").html("<p>Teste</p>");
// Retorna o conteúdo text do PRIMEIRO elemento com classe 'verde'
var conteudo = $(".verde").text();
// Modifica o conteúdo text DE TODOS os elementos com classe 'verde'
$(".verde").text("novo conteúdo de exemplo");
// Retorna o valor digitado de um campo input com nome 'ifsp'
var idade = $("input[name=ifsp]").val();
// Modifica o valor de um campo input com nome 'ifsp'
$("input[name=ifsp]").val("13.125");
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Modificando elementos DOM

- Os métodos **text** e **html** (e **val**), modificam ou acessam o *conteúdo* de uma tag

  <tag>*conteúdo*</tag>

- Além do conteúdo, pode-se modificar os atributos da tag

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Modificando elementos DOM

```
/*
    Retorna o valor do campo 'id' do primeiro 'div'
    --- caso não exista o atributo? undefined ---
*/
var idDiv = $("div").eq(0).attr("id");
/*
    Modifica o valor do atributo 'id' do último 'p'
    --- caso não exista o atributo? adiciona o atributo! ---

*/
$("p").last().attr("id", "ifsp");
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Modificando elementos DOM

- Um caso particular é o atributo multivalorado *class*...
  - Existem três métodos específicos: *addClass*, *hasClass*, *removeClass*

```
// Verifica se o elemento possui a classe 'azul'
if($("#conteudo").hasClass("azul")) {
    // Se existir, remove a classe 'azul' do elemento
    $("#conteudo").removeClass("azul");
} else {
    // Se não existir, adiciona a classe 'azul' ao elemento
    $("#conteudo").addClass("azul");
}
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Modificando elementos DOM

- As informações presentes no CSS dos elementos podem ser lidas ou modificadas usando o método *.css()*

```
// Obtém o valor da propriedade 'color' do primeiro p encontrado
var cor = $("p").eq(0).css("color");

// Modifica a propriedade 'background-color' de todos os div para 'red'
$("div").css("background-color", "red");

// Calcula o tamanho da fonte dos elementos com classe '.classe'
// aumentando o tamanho da fonte em 5px.
var tamanhoFonte = parseInt($(".classe").css("font-size"));
$(".classe").css("font-size", (tamanhoFonte + 5) + "px");
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Exercício 2

- Em muitos sites, existe a opção de alterar o tamanho da fonte para facilitar a leitura em diferentes dispositivos.
- Outro recurso interessante é o “modo noturno”, onde o texto é exibido na cor branca e o fundo é renderizado na cor preta (ou outra cor considerada escura).
- Desenvolva um website contendo quatro botões e diversos parágrafos de texto que realizam as operações de
  - Aumentar ou diminuir a fonte do texto
  - Modo noturno e diurno.
- Para esse exercício, utilize apenas um único método de gerenciamento do evento de clique.

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Exercício 2 - Gabarito

```
$(document).ready(function() {
    $("button").click(function() {
        switch ($(this).attr("id")) {
            case "fMais":
                var tamFonte = parseInt($("p").css("font-size")) + 3;
                $("p").css("font-size", tamFonte + "px");
                break;
            case "fMenos":
                var tamFonte = parseInt($("p").css("font-size")) - 3;
                $("p").css("font-size", tamFonte + "px");
                break;
            case "mNoturno":
                $("body").css("background-color", "black");
                $("p").css("color", "white");
                break;
            case "mDiurno":
                $("body").css("background-color", "white");
                $("p").css("color", "black");
                break;
        }
    });
});
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Modificando a DOM

- Além de modificar os elementos da árvore DOM (elementos HTML e/ou CSS), pode-se ainda modificar a estrutura da árvore por meio da inserção ou remoção de elementos HTML ao site.

- É importante ressalvar que, embora haja um método “puramente jQuery”, é possível adicionar elementos simplesmente usando uma string. Tal método, porém, provê pouca flexibilidade para o desenvolvedor.

```
var elemento = "<p id='primeiro' class='verde'>Texto a ser adicionado</p>";
$("body").append(elemento);
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Modificando a DOM

```
// Cria um novo elemento com a tag "input"
var novoElemento = $("<input>");
// Adiciona o atributo 'type' com valor number
novoElemento.attr("type", "number");
// Adiciona o atributo 'name' com valor idade
novoElemento.attr("name", "idade");
// Adiciona a classe 'azul' ao elemento
novoElemento.addClass("azul");
// Modifica o css do elemento para ter texto vermelho
novoElemento.css("color", "red");
// Modifica o conteúdo padrão do input para '10';
novoElemento.val("10");
/* Adiciona o elemento como PRIMEIRO FILHO do primeiro form do site.
Para 'último filho', use o método APPEND */
$("form").eq(0).prepend(novoElemento);
// Remove o elemento com id=‘teste’
$("#teste").remove()
```

INSTITUTO FEDERAL
São Paulo
Câmpus
São Carlos

Exercício 3

```
<button>Adicionar</button>
<button>Remover</button>
<table>
    <thead>
        <tr>
            <th>Coluna 1</th>
            <th>Coluna 2</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Linha 1</td>
            <td>Linha 1</td>
        </tr>
        <tr>
            <td>Linha 2</td>
            <td>Linha 2</td>
        </tr>
    </tbody>
</table>
```

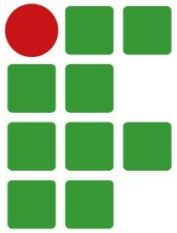
- Dado o trecho HTML ao lado, implemente a funcionalidade de adicionar ou remover linhas do **tbody** usando jQuery.

- Garanta que ao menos 2 e no máximo 17 linhas estejam sempre no **tbody**.

Obrigado!

Prof. Tiago Henrique Trojahn

tiagotrojahn@ifsp.edu.br



INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos