



JQUERY

O que é jQuery?

O objetivo do jQuery (<https://jquery.com/>) é **facilitar o uso do JavaScript** e manter **compatibilidade com diversos browsers**. O jQuery leva muitas tarefas comuns que exigem muitas linhas de código JavaScript para serem realizadas e as envolve em métodos que você pode **chamar com uma única linha de código**. O jQuery também **simplifica** muitas das coisas complicadas do JavaScript, como as chamadas do **AJAX** e a **manipulação do DOM**.

A biblioteca jQuery contém os seguintes recursos:

- Manipulação de HTML / DOM
- Manipulação de CSS
- Métodos de evento HTML
- Efeitos e animações
- AJAX

Adicionar jQuery à páginas:

É possível fazer o download em jquery.com ou utilizar um CDN (Content Delivery Network) como o google.

```
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery
.min.js"></script>
</head>
```

Sintaxe do jQuery

A sintaxe do jQuery é feita sob medida para selecionar elementos HTML e executar alguma ação no (s) elemento (s).

A sintaxe básica é: **\$ (seletor) .action ();**

- Um sinal \$ para definir / acessar jQuery (no seu lugar pode utilizar *jQuery*)
- Um (seletor) para "consultar (ou localizar)" elementos HTML
- Uma ação jQuery () a ser executada no (s) elemento (s)

Exemplos:

<p id='p1' class='teste'>Texto Original</p>

\$("#p1").text('Novo texto JQuery');

\$("p").hide() – oculta todos os elementos <p>

\$(".teste").hide() – oculta todos os elementos com a **classe**="test".

\$("#p1").hide() – oculta todos os elementos com o **id**="test".

O Evento “Document Ready”

Este evento deve ser utilizado para evitar que o código jQuery seja executado antes do carregamento completo do documento

```
$(document).ready(function() {  
    // jQuery métodos aqui  
});
```

Seletores jQuery

Sintaxe	Descrição
\$('#elemento_html')	Seleciona todos os elementos com a tag especificada
\$("#id")	Seleciona o elemento com o atributo id especificado
\$(".class")	Seleciona o(s) elemento(s) com o atributo class especificado
\$("*")	Seleciona todos os elementos
\$(this)	Seleciona o elemento HTML atual
\$(p.intro")	Seleciona todos os elementos <p> com class = "intro"
\$(p:first")	Seleciona o primeiro elemento <p>
\$(ul li:first")	Seleciona o primeiro elemento do primeiro
\$(ul li:first-child")	Seleciona o primeiro elemento de cada
\$("[href]")	Seleciona todos os elementos com um atributo href

<code>\$("a[target='_blank']")</code>	Seleciona todos os elementos <a> com um valor de atributo de destino igual a "_blank"
<code>\$("a[target!='_blank']")</code>	Seleciona todos os elementos <a> com um valor de atributo de destino diferente de "_blank"
<code>\$(":button")</code>	Seleciona todos os elementos <button> e elementos <input> do tipo = "botão"
<code>\$("tr:even")</code>	Seleciona todos os elementos <tr> pares
<code>\$("tr:odd")</code>	Seleciona todos os elementos <tr> ímpares

Ver ferramenta : <https://www.w3schools.com/jquery/trysel.asp>

Seletores Customizados

- `$('div > p');` // <p>s imediatamente filhos de <div>
- `$('p + p');` // <p>s imediatamente precedidos por outro <p>
- `$('div > *:first-child');` // um elemento que seja o primeiro filho direto de uma <div>
- `$('div > *:last-child');` // um elemento que seja o ultimo filho direto de uma <div>
- `$('div p:nth(0)');` // o primeiro elemento <p> filho de uma <div>
- `$('div:empty');` // <div>s vazios

Seletores de Formulário

- **:button** Seleciona elementos do tipo <button> e elementos com type="button"
- **:checkbox** Seleciona inputs com type="checkbox"
- **:checked** Seleciona inputs selecionados
- **:disabled** Seleciona elementos de formulário desabilitados
- **:enabled** Seleciona elementos de formulário habilitados
- **:file** Seleciona inputs com type="file"
- **:image** Seleciona inputs com type="image"
- **:input** Seleciona <input>, <textarea>, e elementos <select>
- **:password** Selecionam inputs com type="password"
- **:radio** Selecionam inputs com type="radio"
- **:reset** Selecionam inputs com type="reset"
- **:selected** Seleciona inputs que estão selecionados
- **:submit** Seleciona inputs com type="submit"
- **:text** Seleciona inputs com type="text"

Exemplo:

`$("#myForm :input"); // seleciona todos os elementos que aceitam entrada de dados`

HTML

```
<div id="turno">
  <input type="radio" name='turno' value="M">Manhã
  <input type="radio" name='turno' value="T">Tarde
  <input type="radio" name='turno' value="T">Noite
</div>
```

CSS

```
.erro{
  border:2px solid red;
}
```

JS

```
function verturno(){
  if($("#turno input:checked").length==0){
    $('#turno').attr('class','erro');
  } else {
    $('#turno').attr('class','');
  }
  ;
}
```

Eventos jQuery

No jQuery, a maioria dos eventos DOM tem um método jQuery equivalente. Para atribuir um evento de clique a todos os parágrafos em uma página, você pode fazer isso:

```
$("#p").click(<nome da função>);
```

Ou

```
$("#p").click(function(){
  // código aqui
});
```

Principais eventos jQuery

Mouse Events	Teclado Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Exemplo

HTML

```
<!DOCTYPE html>

<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery
.min.js">
  <title>jQuery Introdução</title>

</head>

<body>

  <h1> jQuery Sintaxe</h1>
  <p id='p1'> teste de parágrafo 1 </p>
  <div>
    <p id='p2'> teste de parágrafo 2 </p>
  </div>

  <input type="button" value="Ocultar" id='btn1'>
  <script type='text/javascript'
src='js/jqueryIntro.js'></script>
</body>
```

JavaScript

```
$(document).ready(function () {  
    $("#btn1").click(ocultar);  
});  
  
function ocultar() {  
    $("p:first").hide();  
}
```

Trabalhando com Seleções

O jQuery traz também entre suas diversas funcionalidades, uma função que facilita a iteração em elementos de um Array com uma sintaxe mais agradável:

```
<style>  
    .verm {  
        color: red;  
    }  
</style>
```

```
$('h1').attr('class', 'verm'); //altera todas as tags h1 para  
a classe verm
```

Adicionar novo conteúdo HTML

Vamos ver quatro métodos jQuery que são usados para adicionar novos conteúdos:

append () - Insere conteúdo no final dos elementos selecionados

prepend () - Insere conteúdo no início dos elementos selecionados

after () - Insere conteúdo após os elementos selecionados

before () - Insere conteúdo antes dos elementos selecionados

JavaScript

```
$('h1').append(' novo texto');  
$('h1').prepend(' novo texto');
```

Efeitos jQuery

Ver os principais efeitos em (https://www.w3schools.com/jquery/jquery_hide_show.asp)

Setando conteúdo- text(), html(), and val()

text () - Define ou retorna o conteúdo de texto dos elementos selecionados

html () - Define ou retorna o conteúdo de elementos selecionados (incluindo marcação HTML)

val () - Define ou retorna o valor dos **campos de formulário**

O exemplo a seguir demonstra como definir o conteúdo com os métodos text (), html () e val () do jQuery:

HTML

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js"></script>
</head>
<body>

<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>

<p>Input field: <input type="text" id="test3" value="Mickey
Mouse"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>
</body>
</html>
```

JavaScript

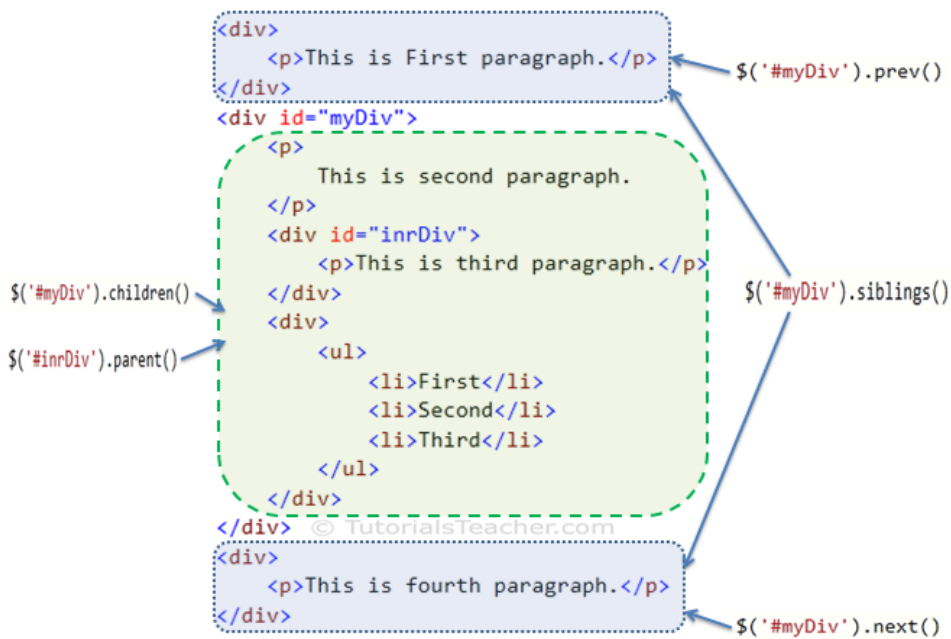
```
<script>
$(document).ready(function() {
    $("#btn1").click(function() {
        $("#test1").text("Hello world!");
    });
    $("#btn2").click(function() {
        $("#test2").html("<b>Hello world!</b>");
    });
    $("#btn3").click(function() {
        $("#test3").val("Dolly Duck");
    });
});
```

</script>

Navegação nos elementos DOM usando jQuery

A biblioteca jQuery inclui vários métodos para percorrer elementos DOM em uma hierarquia DOM. Os principais métodos são listados na tabela abaixo.

jQuery Método	Descrição
children()	Retorna todos os elementos filho do (s) elemento (s) especificado (s)
each()	Iterar sobre elementos especificados e executar a função de retorno de chamada especificada para cada elemento.
eq(indice)	O método eq () retorna um elemento com um número de índice específico dos elementos selecionados.
find()	Retorna todos os elementos filho especificados de cada elemento especificado.
first()	Retorna a primeira ocorrência do elemento especificado.
next()	Retorna o irmão imediatamente seguinte do elemento especificado.
parent()	Retorna o pai do (s) elemento (s) especificado (s).
prev()	Retorna o irmão imediatamente anterior do elemento especificado.
siblings()	Retorna os irmãos de cada elemento especificado (s)



A função **each** chamada logo após um seletor executa a função que passamos como argumento para cada um dos itens encontrados. Essa função precisa de dois argumentos. O primeiro será o “índice” do elemento atual na coleção (0 para o primeiro, 1 para o segundo e assim por diante), e o segundo será o próprio elemento.

HTML

```

<table id='notas'>
  <tr>
    <th>Aluno</th>
    <th>Média Semestre</th>
    <th>Status</th>
  </tr>
  <tr>
    <td>Paula</td>
    <td>6.99</td>
    <td></td>
  </tr>
  <tr>
    <td>Joana</td>
    <td>10</td>
    <td></td>
  </tr>
  <tr>
    <td>Laura</td>
    <td>0.9</td>
    <td></td>
  </tr>

```

```

        <tr>
            <td>Eliane</td>
            <td>7</td>
            <td></td>
        </tr>

</table>

<ul id='lista1'> Lista de Alunos
    <li>Pedro</li>
    <li>Maria</li>
    <li>Julia</li>
</ul>
<p id='msg'></p>

```

JavaScript

```

$("#notas tr").next().each(function (i, r) {
    //$ (this).next();
    nota = parseFloat($(this).children().eq(1).text());
    if (nota >= 7) {
        $(this).children().eq(2).text('Aprovado');
        $(this).addClass('azul');
    } else if (nota < 1.8) {
        $(this).children().eq(2).text('Reprovado');
        $(this).addClass('verm');
    } else {
        $(this).children().eq(2).text('Em exame');
        $(this).addClass('ama');
    }
});

```

CSS

```

body {
    background-color: aquamarine;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
}

```

```

}

.verm {
    color: red;
}

.azul {
    color: blue;
}

.ama {
    color: darkgoldenrod;
}

table {
    border-collapse: separate;
}

td {
    border: 1px solid black;
    border-radius: 10px;
    text-align: center;
}

```

No jQuery existe o elemento **\$(event.target)** que identifica o elemento que gerou o evento

```

$('#lista1').click(function () {
    $('#msg').html('Clicou no item ' +
$(event.target).text());
    $(event.target).remove();
    //$(event.target).css('text-decoration', 'line-through');
});

```

Excluir linhas das tabela

```

$('#notas tbody').click(function () {
    if ($(event.target).is('td')) {
        $(event.target).parent().remove();
    }
});

```

Incluir Linhas na Tabela

```

$("#notas tbody").append('<tr> <td>Aluno teste</td> <td>8</td>
<td> </td> > </tr>');

```

jQuery Manipulando CSS

O jQuery possui vários métodos para manipulação de CSS. Vamos olhar para os seguintes métodos:

`addClass ()` - Adiciona uma ou mais classes aos elementos selecionados

`removeClass ()` - Remove uma ou mais classes dos elementos selecionados

`toggleClass ()` - Alterna entre adicionar / remover classes dos elementos selecionados

`css ()` - define ou retorna o atributo de estilo

jQuery Mask Plugin

Além do jquery Incluir CDN do Plugin

```
<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.1
5/jquery.mask.js'></script>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery.mask/1.14.1
5/jquery.mask.min.js'></script>
```

'0': um número obrigatório

'9': um número opcional

'#': n números opcionais

'A': todos os caracteres, incluindo números

'S': somente caracteres não numéricos

Exemplos

```
$(document).ready(function() {
  $('.date').mask('00/00/0000');
  $('.time').mask('00:00:00');
  $('.date_time').mask('00/00/0000 00:00:00');
  $('.cep').mask('00000-000',{clearIfNotMatch: true});
  $('.cpf').mask('000.000.000-00');
  $('.cnpj').mask('00.000.000/0000-00');
  $('.placa').mask('SSS-0A00');
  $('.money2').mask("#.##0,00", {reverse: true});
  $('.percent').mask('##0,00%', {reverse: true});
  $('.clear-if-not-match').mask("00/00/0000",
{clearIfNotMatch: true});
});
```


Entrada de múltiplas linhas em uma tabela:

HTML

```
<!DOCTYPE html>
<html lang="pt-BR">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="description" content="">
  <meta name="author" content="">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTElPi6
jizo" crossorigin="anonymous"></script>
  <title>Entrada múltiplas linhas</title>

  <link rel="shortcut icon" href="" type="image/x-icon">
  <link rel="apple-touch-icon" href="">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2M
ZwlT" crossorigin="anonymous">

  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHND
z0W1" crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/njGzIxFDs4x0xIM+B0
7jRM" crossorigin="anonymous"></script>

  <!-- Link to your css file -->
  <link rel="stylesheet" href="">
  <script type="text/javascript" src=""></script>
</head>

<body>
  <div class="container">
    <table class="table table-bordered" id='tabela1'>
      <thead>
        <tr>
          <th scope="col">Matricula</th>
          <th scope="col">Nome</th>
        </tr>
      </thead>
    </table>
  </div>
</body>
</html>
```

```

        <th scope="col">Sobrenome</th>
        <th scope="col">Nota</th>
    </tr>
</thead>
<tbody>
    <tr>
        <th scope="row">1</th>
        <td>Mark</td>
        <td>Otto</td>
        <td><input type="text"></td>
    </tr>
    <tr>
        <th scope="row">2</th>
        <td>Jacob</td>
        <td>Thornton</td>
        <td><input type="text"></td>
    </tr>
    <tr>
        <th scope="row">3</th>
        <td>Larry</td>
        <td>the Bird</td>
        <td><input type="text"></td>
    </tr>
</tbody>
</table>
    <button type="button" class="btn btn-primary" id='salvar'>Salvar</button>
</div>
    <p id='resultado'></p>
</body></html>

```

Script

```

function lertab() {
    var tab = $("#tabela1 tr");
    var notas = []; //array para armazenar notas digitadas na tabela
    for (i = 1; i < tab.length; i++) {
        notas.push([parseInt(tab[i].cells[0].innerHTML),
parseFloat(tab[i].cells[3].childNodes[0].value)]);
    }
    //document.write(activities);
    console.table(notas);
    $("#resultado").text(notas);
}
$(document).ready(function() {
    $("#salvar").click(lertab);
})

```