

Programação de Sistemas para Internet

Prof. Diego Cirilo

Aula 14: jQuery

jQuery

- Biblioteca desenvolvida pra simplificar a manipulação do DOM, eventos, requisições, etc;
- Garantia o funcionamento do script em diferentes navegadores/*engines*/versões do JS;
- Possui várias funções de efeitos, como *fade*, *slide*, etc;
- Usa uma linguagem menos *verbosa* que o JavaScript puro (*Vanilla*);

jQuery

- Já foi quase "obrigatória", hoje não mais;
- O CSS/HTML/JS evoluíram, absorvendo funcionalidades do jQuery;
- Mesmo assim, ainda é cômodo utilizá-la em algumas aplicações;
- *You might not need jQuery.*

jQuery

- Pode ser baixada e incluída no projeto localmente;
- Também pode ser importada online via CDN (*Content Delivery Network*);
- Versões:
 - `uncompressed` : código normal;
 - `minified` : código comprimido;
 - `slim` : versão com menos recursos;
- Normalmente utilizamos a versão `minified` ou `min` ;
- ```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
```

# Conceitos

- As funções *jQuery* iniciam com `$` ;
- `$() == jQuery()` ;
- A sintaxe básica é:
  - `$("seletor").ação()`
- O `seletor` seleciona qual/quais elementos serão utilizados;
- A `ação()` executa alguma funcionalidade;

## `$(document).ready()`

- Normalmente os códigos *jQuery* ficam dentro de um:

```
$(document).ready(function(){
 //...codigo...
})
```

- Essa função garante que a página já foi toda carregada.

# Detalhe

- Se usarmos o `defer` no carregamento do *jQuery* o `$` ainda não vai ter sido carregado em scripts no *body*;
- Ex.: o `$(document).ready()` não vai ser reconhecido;
- Nesse caso é necessário usar o `window.onload` antes;

# Exemplo

```
...
<head>
...
<script defer src="https://code.jquery.com/jquery-3.7.1.min.js"></script>`
<script defer src="meuCodigo.js">//se usa o jquery, tem que vir depois</script>`
<body>
...
 <script>
 window.onload = function() {
 // codigo que usa o jQuery($)
 }
 </script>
...
</body>
...
```

# Seletores *jQuery*

- O seletor básico é `$()` :
  - `$("p")`
  - `$(".classe")`
  - `$("#id")`
- Equivalente aos `document.querySelector()`, `document.querySelectorAll()` e `document.getElementById()` do JS *Vanilla*;
- É possível compor seletores mais específicos

# Seletores *jQuery*

- Exemplos:

```
$("*") // tudo
$(this) // o elemento atual
$("p.intro") // todos <p> com classe intro
$("p:first") // o primeiro <p>
$("ul li:first") // o primeiro do primeiro
$("ul li:first-child") // o primeiro de todos
$("a[href]") // todos elementos com atributo href
$("a[target='_blank']") // todos <a> com target="_blank"
$("a[target!='_blank']") // todos <a> com target diferente de "_blank"
$("button") // todos <button> e <input> com type="button"
$("tr:even") // <tr> par
$("tr:odd") // <tr> impar
```

# Eventos *jQuery*

- click , dblclick , hover , keypress , focus , change , etc;
- Ex.:

```
$("#meuBotao").click(function() {
 $(this).css("color", "red");
});
```

# Manipulação do DOM com *jQuery*

- CSS

```
$("#id").css({
 "color": "blue",
 "background-color": "red"
});
```

- Class

```
$("#id").addClass("btn btn-primary");
```

# Manipulação do DOM com *jQuery*

- Hide/Show:

```
$("#id").hide();
$("#id").show();
```

- Get/Set:

```
let conteudo = $(sel).text(); // get
$(sel).text("conteúdo de texto"); // set
let conteudo = $(sel).html(); //get
$(sel).html("conteudo HTML interno"); //set
let conteudo = $(sel).val(); //get
$(sel).val("valor de form"); //set
```

# Manipulação do DOM com *jQuery*

- Atributos:

```
let atributo = $("#id").attr("href"); //get
$("#id").attr("href", "https://site.com"); //set
```

- Propriedades:

```
$("#id").prop("checked", false);
```

# Manipulação do DOM com *jQuery*

- Adicionar elementos:

```
$("#lista").append("Item no final");
$("#lista").prepend("Item no início");
$("#lista").after("<p>Texto depois da lista</p>");
$("#lista").before("<p>Texto antes da lista</p>");
```

- Remover:

```
$("#id").remove(); // Remove o elemento completamente
$("#id").empty(); // Remove apenas o conteúdo interno
```

# Manipulação do DOM com *jQuery*

- Clone:

```
let clone = $("#meuElemento").clone();
$("#outroElemento").append(clone);
```

- Substituir:

```
$("#meuElemento").replaceWith("<p>Novo Parágrafo</p>");
```

# Manipulação do DOM com *jQuery*

- Criar elemento

```
let botao = $("<button>", {
 text: "Clique Aqui",
 id: "meuBotao",
 class: "btn btn-primary"
});

$("#container").append(botao);
```

# Manipulação do DOM com *jQuery*

- Acessar elementos *pais*:

```
$("span").parent();
$("span").parents();
$("span").parents("ul"); //filtro
$("span").parentsUntil("div");
```

- *Filhos/Irmãos*

```
$("div").children();
$("div").siblings();
$("div").prev();
$("div").next();
$("div").find();
```

# Exemplo

- teste.html :

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width">
 <title>Teste</title>
 <script defer src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
 <script defer src="meuScript.js"></script>
 </head>
 <body>
 <div class="conteudo outra-classe">
 <h1>Teste</h1>
 <button id="meuBotao">Clique aqui!</button>
 <p>Meu conteúdo relevante.</p>
 </div>
 </body>
</html>
```

# Exemplo

- meuScript.js

```
const conteudo = $(".conteudo p");
let contador = 0;
botao.click(() => {
 contador++;
 if (contador < 10) {
 conteudo.html(`<p>0 botão foi clicado ${contador} vezes!</p>`);
 } else if (contador < 16){
 conteudo.html(`<p>0 botão foi clicado ${contador} vezes! Por favor, pare!</p>`);
 $(".conteudo h1").text("TESTE");
 $("#meuBotao").text("Não clique aqui!");
 $("#meuBotao").css({
 "position": "absolute",
 "top": `${contador**2}px`
 });
 } else {
 let aviso = $("<h1>", {
 text: "PARE!!!",
 css: {
 "font-size": "20em",
 "color": "yellow"
 }
 });
 $("body").css("background-color", "red");
 $("body").html(aviso);
 }
});
```

# Referências

- <https://api.jquery.com/>
- <https://www.w3schools.com/jquery/default.asp>

# Dúvidas?

