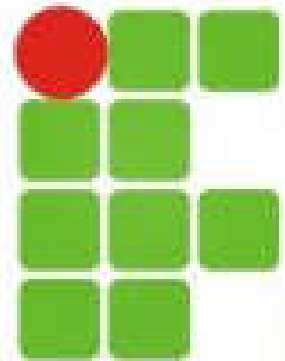


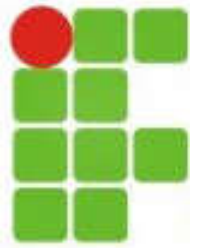
Fundamentos de Web Design 2

Professor Eng. Dr. Will Roger Pereira



Objetivos desta aula

- Conhecer os seletores jQuery.





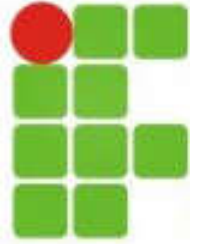
Sintaxe jQuery

- Lembre-se sempre da sintaxe jQuery:
 - `Seletor.função(argumentos);`
- O seletor definirá quais elementos HTML estarão envolvidos na função;
- Serão mostrados os seletores levando em consideração diversos aspectos → Parecido com CSS.



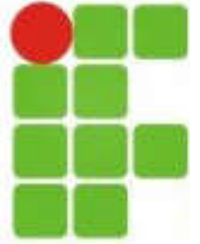
Seletores

- Os seletores jQuery sempre possuirão a seguinte estrutura:
 - `$(conteúdo)`
- O conteúdo dentro do parênteses será quase sempre uma string → Texto entre aspas (“”);
- Os elementos selecionados dependerão do conteúdo;
- Referência para todos os seletores (em inglês):
 - http://www.w3schools.com/jquery/jquery_ref_selectors.asp



Seletor tipo

- Selecionará todos os elementos de determinado **tipo**;
 - `$(“tipo”) → Ex: $(“p”), $(“td”), $(“li”).`
- Caso haja mais de um elemento do tipo, todos os elementos são selecionados;
- Mudando a cor de letra de **todos** os parágrafos:
 - `$(“p”).css(“color”, “red”);`



Seletor .classe

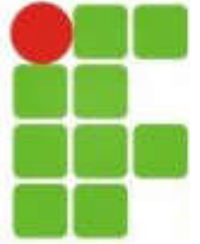
- Selecionará todos os elementos de determinada **classe**;
 - \$("classe") → Ex: \$(".links"), \$(".vermelho");
- Caso haja mais de um elemento na classe, todos os elementos são selecionados;
- Mudando a cor de letra de todos os elementos da classe links:
 - \$(".links").css("color","red");



Seletor #id

- Selecionará o elemento de determinado **id**;
 - `$("#id")` → Ex: `$("#will")`, `$("#titulo")`.
- Mudando a cor de letra do elemento com id titulo:
 - `$("#titulo").css("color","red");`

Selecionando múltiplos elementos

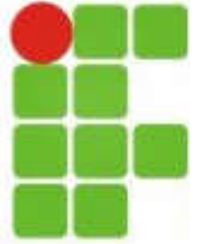


- Quando quiser realizar a mesma ação para diversos seletores, separe-os por vírgula;
 - `$(“seletor1, seletor2, etc”);`
- Mudando a cor de letra de todos os links, parágrafos, elementos de listas e elementos da classe red:
 - `$(“a, p, li, .red”).css(“color”,”red”);`



Selecionando descendentes

- Selecionará todos os elementos **filho** contidos em elementos **pai**;
 - `$(pai filho);`
- Ex: `$(“li a”)` → Todos os links dentro de elementos li;
- Ex: `$(“table .red”)` → Todos os elementos da classe red dentro de uma tabela;
- Ex: `$(“#will p”)` → Todos os parágrafos dentro de um elemento de id will;
- Nesta situação não importa se é filho direto ou não, deve apenas estar contido dentro dele.



Subseletores

- Subseletores servem para realizar uma seleção especial, baseado não somente em id, classe e tipo;
- Estes seletores sempre começam com dois pontos(:):
 - `$(:subseletor)`.





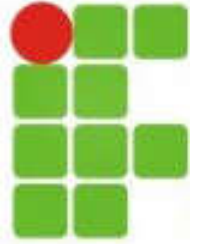
Subseletores

- Permitem também uma seleção refinada com os seletores já aprendidos, especificando ainda mais a seleção:
- Para isto, basta adicionar o seletor especial após qualquer seletor, dentro das aspas:
 - Ex: \$("seletor:subseletor").
 - Aqui será aplicado o subseletor somente nos elementos selecionados pelo seletor.
- Ou utilizá-lo sozinho, também dentro das aspas:
 - Ex: \$("subseletor").
 - Aqui será aplicado o subseletor em todos os elementos da página.



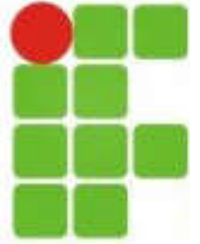
Subseletor :checked

- Este seletor especial permite selecionar todos os elementos cujo atributo **checked** possui o valor checked;
- Funciona para **checkbox**:
 - `<input type="checkbox" checked="checked">`;
 - 
- E para **radiobutton**:
 - `<input type="radio" checked="checked">`;
 - 



Subseletor :checked

- Selecionando todos os elementos checados:
 - \$(" :checked");
- Selecionando todos os elementos checados da classe will:
 - \$(".will:checked");
- Selecionando o elemento checado que possui id iftm:
 - \$("#iftm:checked").



Outros subseletores

- Há diversos outros subseletores, pesquise nas referências:
- **:input** → Todos os elementos “input”;
- **:text** → Todos os elementos “input” com tipo “text”;
- **:radio** → Todos os elementos “input” com tipo “radio”;
- **:checkbox** → Todos os elementos “input” com tipo “checkbox”;
- **:submit** → Todos os elementos “input” com tipo “submit”;
- **:button** → Todos os elementos “input” com tipo “button”;
- **:image** → Todos os elementos “input” com tipo “image”;
- **:selected** → Todos os elementos “input” selecionados (select).

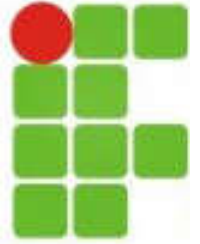


Indexação dos elementos

- Imagine uma estrutura HTML como a descrita a seguir:

```
<ul>
  <li>Primeiro</li>
  <li>Segundo</li>
  <li>Terceiro</li>
  <li>Quarto</li>
  <li>Quinto</li>
</ul>
```

- Uma característica do jQuery é que os elementos selecionados são indexados como um vetor.

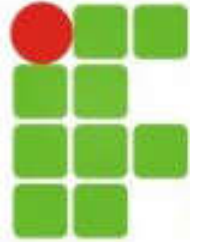


Indexação dos elementos

- Suponha um seletor \$("ul li"). Seleccionaria todos os **li** da **lista não-ordenada**;
- Seleccionando os elementos, cada um receberia um índice:

```
<ul>  
  <li>Primeiro - Índice 0</li>  
  <li>Segundo - Índice 1</li>  
  <li>Terceiro - Índice 2</li>  
  <li>Quarto - Índice 3</li>  
  <li>Quinto - Índice 4</li>  
</ul>
```

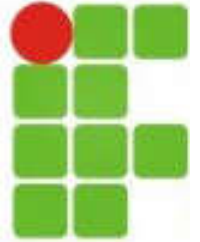
- Estes **índices** podem ser aproveitados para outros subseletores.



Subseletor :first

- Selecionará, dentre os elementos selecionados, o primeiro elemento:
 - \$("seletor:first");
- Ex: \$("ul li:first");

- Primeiro - Índice 0
- Segundo - Índice 1
- Terceiro - Índice 2
- Quarto - Índice 3
- Quinto - Índice 4



Subseletor :last

- Seleccionará, dentre os elementos seleccionados, o último elemento:
 - \$("seletor:last");
- Ex: \$("ul li:last"):

- Primeiro - Índice 0
- Segundo - Índice 1
- Terceiro - Índice 2
- Quarto - Índice 3
- Quinto - Índice 4



Subseletor :even

- Selecionará, dentre os elementos selecionados, os elementos com **índice** par:
 - \$("seletor:even");
- Ex: \$("ul li:even"):

- Primeiro - Índice 0
- Segundo - Índice 1
- Terceiro - Índice 2
- Quarto - Índice 3
- Quinto - Índice 4



Subseletor :odd

- Seleccionará, dentre os elementos selecionados, os elementos com **índice** ímpar:
 - \$("conteúdo:odd");
- Ex: \$("ul li:odd");

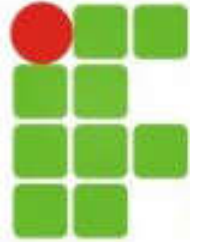
- Primeiro - Índice 0
- Segundo - Índice 1
- Terceiro - Índice 2
- Quarto - Índice 3
- Quinto - Índice 4



Subseletor :eq

- Seleccionará, dentre os elementos seleccionados, o elemento com determinado índice;
 - \$("seletor:eq(número)");
- Ex: \$("ul li:eq(3)");

- Primeiro - Índice 0
- Segundo - Índice 1
- Terceiro - Índice 2
- Quarto - Índice 3
- Quinto - Índice 4



Subseletor :gt

- Seleccionará, dentre os elementos seleccionados, os elementos com índice maior que determinado número;
 - \$("seletor:gt(número)");
- Ex: \$("ul li:gt(2)");

- Primeiro - Índice 0
- Segundo - Índice 1
- Terceiro - Índice 2
- Quarto - Índice 3
- Quinto - Índice 4



Subseletor :lt

- Seleccionará, dentre os elementos seleccionados, os elementos com índice menor que determinado número;
 - \$("seletor:lt(número)");
- Ex: \$("ul li:lt(2)");

- Primeiro - Índice 0
- Segundo - Índice 1
- Terceiro - Índice 2
- Quarto - Índice 3
- Quinto - Índice 4