

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
CURSO TÉCNICO INTEGRADO EM INFORMÁTICA
CAMPUS CURRAIS NOVOS

Desenvolvimento Web

jQuery – aula I –

Introdução, inserção e seletores

Professor: Bruno E. G. Gomes

2013

INTRODUÇÃO

- *jQuery* é uma biblioteca feita para facilitar o trabalho em *Javascript*
 - faça mais, escreva menos
 - tarefas que requerem muitas linhas de código em Javascript puro podem ser feitas de modo ágil em jQuery
- A própria biblioteca é um conjunto de classes e funções *Javascript*
- Simplifica a manipulação do *Document Object Model* (DOM)



JQUERY

○ Principais usos

- Manipulação HTML/DOM
- Manipulação CSS
- Eventos e métodos HTML
- Efeitos e animações
- Integração com AJAX
- Diversos *plugins* para tarefas comuns (calendário, *sliders*, menus, etc.)

○ Quem usa?

- Wikipedia, Google, Netflix, Wordpress, dentre outros



INSTALANDO

○ Duas maneiras

1. Baixar a biblioteca e salvar o arquivo no seu diretório de desenvolvimento

<http://jquery.com/download/>

2. Usando um CDN que contenha uma versão do jQuery.

```
<script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
```

○ No caso do CDN

- É preciso uma conexão ativa com a Internet na primeira vez que for acessar o arquivo
- Uma vez baixado, o servidor armazena o arquivo, não necessitando baixá-lo outra vez



JQUERY VS. \$()

- Essas duas notações são utilizadas para acessar o objeto *jQuery*
 - `$()` é mais conveniente por ser mais curta
- A partir desta notação é possível acessar elementos do DOM e chamar métodos para operar sobre eles.
- Ex.: `$("p")` ou `jQuery ("p")` seleciona todos os parágrafos do documento



\$ (DOCUMENT).READY()

- Garante que a página seja manipulada de forma segura
 - *somente quando o DOM estiver carregado*
- É uma boa prática usar o código *jQuery* dentro desta função

```
$( document ).ready(function() {  
    $("#div1").html ("Pronto!" );  
});
```



\$ (WINDOW).LOAD()

- Semelhante a *#document.ready()*, exceto que:
 - O código será executado apenas quando todo o documento estiver carregado, incluindo imagens e frames.

```
$( window ).load(function) {  
    $("#div1").html ("Pronto!" );  
});
```



PASSANDO UMA FUNÇÃO NOMEADA

- É possível passar uma função nomeada a $\$(document)$ ou $\$(window)$ ao invés de uma função anônima

```
function fn1 ( jQuery ) {  
    // Código a ser executado quando o documento estiver pronto  
}  
  
$( document ).ready( fn1 );  
// ou:  
$( window ).load( fn1 );
```

SELETORES

- O conceito mais básico de *jQuery* é "*selecionar um elemento e fazer algo com ele*"
- *jQuery* suporta a maioria dos seletores CSS e alguns próprios

| Seleção por | Exemplo |
|----------------|--|
| ID | <code>\$("#myId");</code> |
| Nome da classe | <code>\$(".myClass");</code> |
| Atributo | <code>\$("input[name='identidade']");</code> |
| CSS composto | <code>\$("#conteudo ul.pessoas li");</code> |



PSEUDO-SELETORES

- Inseridos com **:** (dois pontos) seguido do nome do pseudo-seletor

| Exemplo | Significado |
|--|---|
| <code>\$("a.external:first");</code> | O primeiro elemento dos <i>links</i> que tem a classe “.external” |
| <code>\$("tr:odd");</code> | Todas as linhas que são ímpares |
| <code>\$("#myForm:input")</code> | Todos os elementos <i>input</i> em um formulário |
| <code>\$("div:visible");</code> | Todas as divisões que são visíveis. <i>visible</i> : “height” e “width” são maiores que 0 (zero). No caso de “tr” é quando “display” é diferente de “none” |
| <code>\$("div:gt(2)");</code> | Todas as divisões, exceto as duas primeiras <i>gt</i> = maior que |



TESTANDO SE UMA SELEÇÃO CONTÉM ELEMENTOS

- Para realizar o teste, use código semelhante ao exemplo
 - O tamanho (*length*) 0 (zero) equivale ao booleano **false**. Diferente de zero é igual a **true**

```
if ( $( "div.foo" ).length ) {  
    ...  
}
```



OBTER E MODIFICAR INFORMAÇÕES DE UM ELEMENTO

| Método | Significado |
|-------------|---|
| .html() | Obter ou inserir conteúdo HTML. O <i>mesmo que .innerHTML em JS.</i> |
| .text() | Obter ou inserir conteúdo de texto |
| .val() | Obter ou inserir o valor de um elemento de formulário. O <i>mesmo que .value em JS.</i> |
| .attr() | Obter ou modificar o valor de um atributo |
| .width() | Obter ou modificar a largura em pixels do primeiro elemento da seleção |
| .height() | Obter ou modificar a altura em pixels do primeiro elemento da seleção |
| .position() | Obter a posição de um elemento relativa ao seu ancestral |

MANIPULANDO ATRIBUTOS

- É possível modificar ou consultar o valor de um atributo de *tag*

```
$( "a" ).attr( "href",  
"allMyHrefsAreTheSameNow.html" );
```

```
$( "a" ).attr( "href" ); //apenas consulta
```

```
$("#btn").attr("value", "imprimir");
```



CRIANDO ELEMENTOS DINAMICAMENTE

- A mesma notação de seletor `$()`

```
//cria o parágrafo e atribua referência a variável $novo
var $novo = $( "<p>New element</p>" );
//insere em #content
$novo.appendTo( "#content" );
//remove de #content inserindo após a última "ul"
$novo.insertAfter( "ul:last" );
// "Clona" o parágrafo, criando um novo e inserindo-o
depois do último parágrafo
$( "ul" ).last().after( $novo.clone() );
```