

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

Desenvolvimento Web

jQuery – aula II –

*CSS e métodos para percorrer a
árvore DOM*

Professor: Bruno E. G. Gomes

2013

INTRODUÇÃO

- Nesta aula, continuando o estudo de *jQuery*
 - Manipular CSS com *jQuery*
 - Percorrer a árvore DOM a partir do relacionamento de “parentesco” das marcações



OBTENDO PROPRIEDADES CSS

- **.css** (*nome-da-propriedade*) é usado para obter o seu valor

```
$( "h1" ).css( "fontSize" ); // Retorna o tamanho da fonte  
$( "h1" ).css( "font-size" ); // outra forma, com hífen.
```



MODIFICANDO UMA PROPRIEDADE CSS

- `.css` (nome, valor) ou
- `.css` (nome, conjunto-valores-como-objeto)
- *OBS. 1: Se usar a notação com hífen coloque o nome da propriedade entre “ ” (aspas)*
- *OBS 2.: No código “final” é melhor manter o CSS separado do Javascript. Você pode chamar, no Javascript, uma classe definida no CSS*

```
// Propriedade individual
$( "h1" ).css( "fontSize", "100px" );

//Modificando múltiplas propriedades
$( "h1" ).css({
  fontSize: "100px",
  color: "red"
});
```

MELHOR PRÁTICA AO USAR CSS COM JAVASCRIPT

- Manter o CSS definido em um arquivo separado
 - Incluir, remover ou alterar classes CSS no Javascript quando necessário

```
// Trabalhando com CSS
var $h1 = $( "h1" );
$h1.addClass( "big" );
$h1.removeClass( "big" );

if ( $h1.hasClass( "big" ) ) {
...
}
```

PERCORRENDO A ÁRVORE DOM

- Além do uso de seletores, é possível atravessar a árvore DOM através de diversos métodos
 - Obter nós pais, filhos, irmãos
 - Pegar o primeiro nó, próximo, o último, dentre outros.

```
<div class="avo">  
  <div class="pai">  
    <div class="filho">  
      <span class="subfilho"></span>  
    </div>  
  </div>  
  <div class="irmao"></div>  
  <div class="irmao2"></div>  
</div>
```



PAIS (PARENTS)

```
<div class="avo">
  <div class="pai">
    <div class="filho">
      <span class="subfilho"></span>
    </div>
  </div>
</div>
```

// Selecionando o pai direto

```
$( "span.subfilho" ).parent(); //retorna [ div.filho ]
```

// Selecionando todos os pais de acordo com um seletor

```
$( "span.subfiho" ).parents( "div.pai" ); //retorna [div.pai]
```

```
$( "span.subfilho" ).parents(); // retorna [ div.filho, div.pai, div.avo ]
```

// Seleciona todos os pais de um elemento até o seletor (sem incluí-lo)

// retorna [div.filho, div.pai]

```
$( "span.subfilho" ).parentsUntil( "div.avo" );
```



FILHOS (CHILDREN)

```
<div class="avo">  
  <div class="pai">  
    <div class="filho">  
      <span class="subfilho"></span>  
    </div>  
  </div>  
  <div class="irmao"></div>  
  <div class="irmao2"></div>  
</div>
```

// *children* – seleciona os filhos ‘diretos’

```
$("#div.avo").children ("div"); //retorna [ div.pai, div.irmao1, div.irmao2 ]
```

// *find* – Seleciona todos os elementos da seleção de acordo com o seletor

```
$("#div.avo").find ("div"); //retorna [ div.pai, div.filho, div.irmao1, div.irmao2 ]
```


IRMÃOS (SIBLINGS)

```
<div class="avo">
  <div class="pai">
    <div class="filho">
      <span class="subfilho"></span>
    </div>
  </div>
  <div class="irmao"></div>
  <div class="irmao2"></div>
</div>
```

// Seleciona o próximo irmão

```
$("#div.pai").next(); //retorna [ div.irmao1]
```

// Seleciona o irmão antecessor

```
$("#div.pai").prev( "div.pai" );
```

// Seleciona todos os próximos irmãos

```
$("#div.pai").nextAll();
```

// Seleciona o primeiro elemento dentre todos os próximos

```
$("#div.pai").nextAll().first();
```

// Seleciona o último elemento dentre todos os próximos

```
$("#div.pai").nextAll().last();
```

// Todos os antecessores (prevAll):

```
prevAll(), prevAll().first(), prevAll().last()
```

