



Animações simples e eventos

O que queremos agora é adicionar uma animação para quando passarem o mouse em cima de um produto, a linha dele mudar de cor.

Precisamos saber o momento em que o mouse passa por cima de um elemento. É o chamado **hover**.

No nosso código, selecionamos os **trs** e da mesma maneira que utilizamos o **.click()**, podemos usar o **.hover()**, executando uma função. Podemos chamá-la de **daDestaque**.

Queremos executar isso sempre, então adicionamos o código dentro da função **aposInicializado**:

```
$( "tr" ).hover( daDestaque );
```

Na função **daDestaque** queremos mudar apenas um atributo do **css**. Podemos usar o método **css**, passando o atributo e o seu novo valor:

```
var daDestaque = function(){  
  
    $(this).css( "background", "#ccc" );  
  
}
```

O método **.css()** pode receber mais de um atributo, com os seus respectivos valores, por exemplo:

```
$(this).css( { "background" : "#ccc",  
              "opacity" : 0.5,  
              } );
```

No nosso caso, queremos alterar só o **background** no **hover**, mas você pode passar mais valores se precisar.

Se testarmos na nossa página, podemos ver que quando passamos o mouse em cima de uma linha ela muda de cor, mas o problema é que quando o mouse não está mais em cima da linha, ela não volta para a sua cor anterior.

Para fazermos isso, o método **hover** recebe um segundo parâmetro, que será a função que ele executará quando o mouse sair de cima do **tr**:

```
$( "tr" ).hover( daDestaque, tiraDestaque );
```

Dentro da função **tiraDestaque**, precisamos lembrar qual é a cor anterior para que colocarmos dentro do **.css**. Mas fica difícil de lembrar. E se eu deixar o valor da cor vazio?

```
var tiraDestaque = function(){  
  
    $(this).css("background", "");  
  
}
```

Se testarmos, isso irá funcionar. Mas não parece muito certo deixamos um valor vazio, não é mesmo? Nesse caso seria melhor usarmos uma **classe** para as características de destaque, assim esses detalhes do css não precisam ser alterados dentro do nosso código javascript.

Vamos criar uma classe com essas características, no nosso arquivo de css, podemos adicionar a classe **hovering**:

```
.hovering{  
    background: #ccc;  
}
```

Agora podemos usar o **.addClass()** na função **daDestaque** e o **.removeClass()** na função **tiraDestaque**:

```
var daDestaque = function(){  
  
    $(this).addClass("hovering");  
  
}  
  
var tiraDestaque = function(){  
  
    $(this).removeClass("hovering");  
  
}
```

Atualizamos a página, e está tudo funcionando!

Poderíamos implementar essa funcionalidade do **.hover()**, que recebe um ou dois parâmetros, um quanto o mouse entra em cima do elemento e outro quando o mouse sai de cima do elemento, de uma outra maneira.

Na verdade os dois eventos que executamos são o **mouseenter** e o **mouseleave**, queremos que todas vez que esses eventos acontecerem, chame as funções **daDestaque** e **tiraDestaque** respectivamente:

```
$( "tr" ).on( "mouseenter", daDestaque );  
$( "tr" ).on( "mouseleave", tiraDestaque );
```

Utilizando o `.on()`, podemos chamar uma função, quando um determinado evento acontecer.

Repare que estamos mudando de cor no **hover** de todos os **trs** da nossa página, o nosso cabeçalho também é um **tr**, então consequentemente ele também mudará de cor. Não é isso que queremos, precisamos selecionar somente os **trs** que estão dentro da tag **tbody**.

No seletor do Jquery, podemos dizer que queremos todos os **trs**, mas somente se forem filhos de um **tbody**:

```
$( "tbody tr" ).hover( daDestaque, tiraDestaque );
```

Se tivéssemos mais tabelas no nosso HTML correríamos algum risco, nesse caso seria melhor selecionarmos o **tbody** dentro do elemento com a classe **carrinho**:

```
$( ".carrinho tbody tr" ).hover( daDestaque, tiraDestaque );
```

Repare que quanto mais restrições adicionamos no seletor, mais atrelados ficamos a estrutura da nossa página. Talvez fosse melhor adicionar uma classe ou no **tbody** ou mesmo nos **trs** que desejamos selecionar. São coisas que precisamos pesar para tomar o melhor rumo no nosso projeto.

