

Este curso é de propriedade do aplicativo [SoloLearn](#) e foi traduzido por [Ana Beatriz Augusto](#) usando os recursos [Reverso Context](#) e [Google Tradutor](#).

Efeitos `hide()` e `show()`

O jQuery tem alguns efeitos de fácil execução para criar animações.

Os métodos `hide()` e `show()` são usados para esconder e mostrar elementos selecionados.

O método `toggle()` é usado para alternar entre esconder e mostrar elementos.

Por exemplo:

```
$(function(){  
    $("p").click(function(){  
        $("div").toggle();  
    });  
});
```

Os métodos `hide()`, `show()` e `toggle()` podem ter um parâmetro opcional, `velocidade`, que especifica a velocidade da animação em milissegundos. Por exemplo, vamos colocar 1000 milissegundos como velocidade no método `toggle()`:

```
$(function(){  
    $("p").click(function(){  
        $("div").toggle(1000);  
    });  
});
```

Estes métodos também podem ter um segundo parâmetro opcional que é uma função a ser executada depois que a animação se completa.

Efeitos `fadeIn()` e `fadeOut()`

Similares aos efeitos `hide()` e `show()`, o jQuery fornece os efeitos `fadeIn()` e `fadeOut()` que fazem o aparecimento e o desaparecimento gradual.

Igual ao método `toggle()`, que alterna entre esconder e mostrar, o `fadeToggle()` alterna entre fazer o aparecimento e o desaparecimento gradual.

Vamos analisar o `fadeToggle()`:

```
$(function(){  
    $("p").click(function(){  
        $("div").fadeToggle(1000);  
    });  
});
```

Outro método usado para o mesmo efeito é o `fadeTo()`, que permite especificar uma opacidade entre 0 e 1. Por exemplo:

```
$("div").fadeTo(1500, 0.7);
```

Efeitos `slideUp()` e `slideDown()`

Os métodos `slideUp()` e `slideDown()` são usados para criar um efeito de deslizar nos elementos.

De novo, de modo similar aos métodos `toggle` anteriores, o método `slideToggle()` alterna entre os efeitos `slideUp()` e `slideDown()` podendo também receber dois parâmetros opcionais: velocidade e função.

Por exemplo:

```
$(function(){  
    $("p").click(function(){  
        $("div").slideToggle(500);  
    });  
});
```

O método `animate()`

Te permite animar usando um valor fixo ou um valor relativo ao valor atual. Você precisa definir as propriedades CSS a serem animadas como parâmetro no formato JSON (pares de "chave": "valor").

O segundo parâmetro define a velocidade da animação.

Por exemplo, o seguinte código anima a propriedade `width` da `div` em 1 segundo para o valor `250px`:

```
$("div").click(function(){  
    $("div").animate({width: '250px'}, 1000);  
});
```

Deve se prestar atenção ao formato JSON ao se fornecer os parâmetros CSS. A sintaxe JSON também foi usada nos assuntos anteriores quando manipulamos propriedades CSS.

Você pode animar qualquer propriedade CSS usando a sintaxe mencionada na página anterior, mas há algo importante a se lembrar: todos os nomes de propriedade devem ser camelizadas quando usadas com o método `animate()`.

Você precisará escrever `paddingLeft` invés de `padding-left`, `marginRight` invés de `margin-right` e assim em diante.

Múltiplas propriedades podem ser animadas ao mesmo tempo, sendo separadas por vírgulas.

Por exemplo:

```
$("div").animate({  
  width: '250px',  
  height: '250px'  
}, 1000);
```

Também é possível definir valores relativos (o valor é relativo ao atual valor do elemento). Isso é feito colocando += ou -= na frente do valor.

```
$("div").animate({  
  width: '+=250px',  
  height: '+=250px'  
}, 1000);
```


Fila de animação

Por padrão, o jQuery vem com uma funcionalidade de fila para animações. Isso significa que se você escrever vários `animate()` seguidos, jQuery cria uma lista "interna" para estes métodos. Então ele executa as animações uma por uma.

Por exemplo:

```
var div = $("div");
div.animate({opacity: 1});
div.animate({height: '+=100px',
width: '+=100px', top:
'+=100px'}, 500);
div.animate({height: '-=100px',
width: '-=100px', left:
'+=100px'}, 500);
div.animate({height: '+=100px',
width: '+=100px', top:
'-=100px'}, 500);
div.animate({height: '-=100px',
width: '-=100px', left:
'-=100px'}, 500);
div.animate({opacity: 0.5});
```

Cada método `animate()` irá ser executado um após o outro.
Lembre-se, para manipular a posição dos elementos, você precisa definir a propriedade CSS do elemento como `relative`, `fixed` ou `absolute`.

O método `animate()`, assim como os métodos `hide/show/fade/slide`, podem ter uma função como parâmetro, que é executada depois que o atual efeito termina.

Menu

Vamos criar um simples menu que irá abrir quando for clicado no item menu.

HTML:

```
<div class="menu">
  <div id="item">
    Drop-Down
  </div>
  <div id="submenu">
    <a href="#">Link 1</a>
    <a href="#">Link 2</a>
    <a href="#">Link 3</a>
  </div>
</div>
```

JavaScript:

```
$("#item").click(function(){
  $("#submenu").slideToggle(500);
});
```

Este curso é de propriedade do aplicativo SoloLearn e foi traduzido por Ana Beatriz Augusto usando os recursos Reverso Context e Google Tradutor.