



Andre Noel

# Resumo

- Introdução
- Sintaxe básica
- Manipulação de elementos
- CSS e inspeção de elementos
- Eventos
- Efeitos
- Animações

# Introdução



# O que é?

- Biblioteca JS
- Software Livre e Aberto
- Foco em simplicidade
- Leve
- Crossbrowser

# Simplicidade

- *“O foco principal da biblioteca jQuery é a simplicidade. Por que submeter os desenvolvedores ao martírio de escrever longos e complexos códigos para criar simples efeitos?”*
  - John Resig, no prefácio do livro *jQuery in Action*

# jQuery é JavaScript!

- “Posso misturar jQuery e JavaScript?”
- Não precisa conhecer muito de JS
- Usa os seletores de CSS

# Pra que serve?

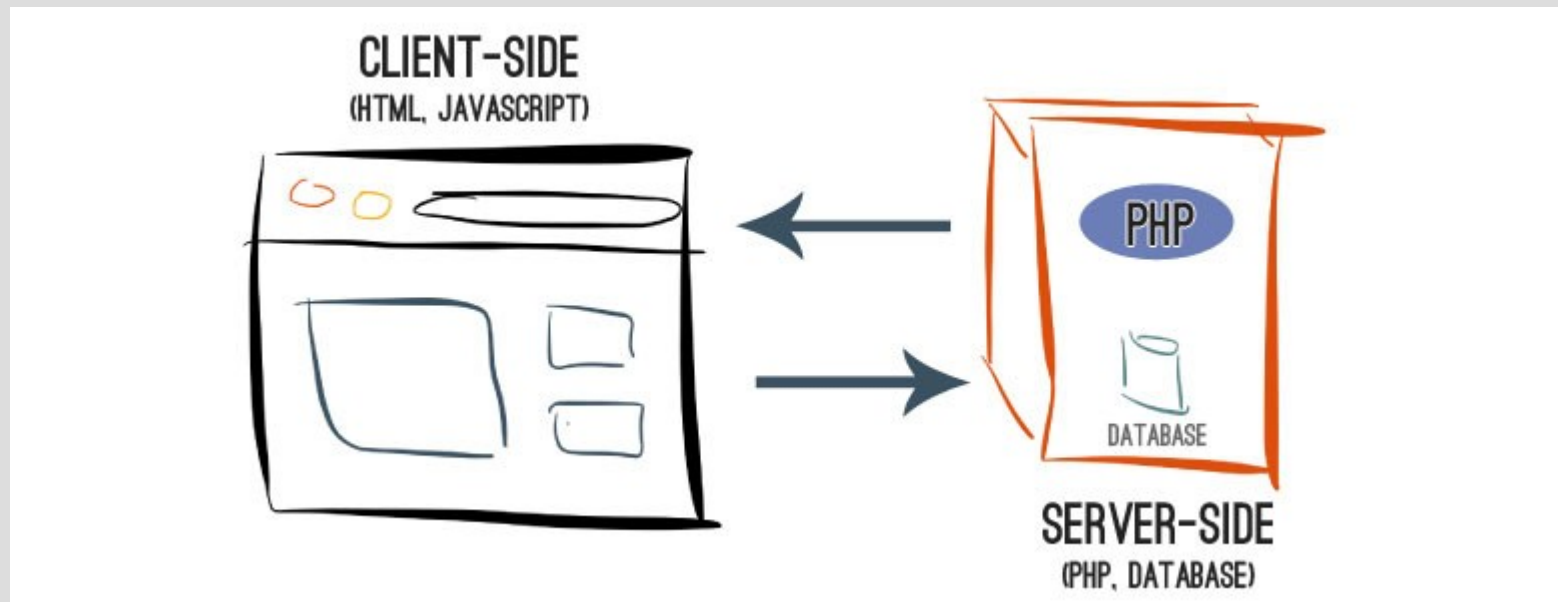
- Interatividade e dinamismo
- Adicionar efeitos e animações
- Acessar e manipular o DOM
- Buscar informações no servidor sem recarregar a página (ajax)
- Prover interatividade

# Pra que serve?

- Alterar conteúdos
- Modificar a apresentação e estilização
- Simplificar tarefas específicas de JS
- Realizar outras tarefas relacionadas



# Cliente x Servidor



# Em conformidade com os padrões web

- W3C
- Qualquer SO e navegador
- Total suporte a seletores CSS3

# Características

- Utiliza seletores CSS
- Arquitetura compatível com plugins e extensões
- Indiferente a inconsistências de renderização
- Capaz de interação implícita
- Admite programação encadeada
- Extensível

# Como instalar

- Download de 1 arquivo
- `<script type="text/javascript" src="jquery-2.2.0.min.js"></script>`

# Um pouco de JS

- ex1.html

# Sintaxe básica

- `jQuery(<seletor>)`
- `$(<seletor>)`
  - `$('h1')`
  - `$('p')`
  - `$('#rodape')` → pelo id do elemento
  - `$('.post')` → pela classe dos elementos
  - `$('body p:nth-child(3)')`

# Encadeamento

- `$('#div').children('p').fadeOut().addClass('xpto')`
  - “Construtor, encontre todos os elementos parágrafos que sejam filhos dos elementos div, neles aplique um efeito de esmaecimento (fadeOut) e adicione a classe 'xpto'”
  - Cada método retorna um objeto pronto para receber outro método

# .each()

- Executa uma função sempre que encontrar um elemento do conjunto



# .length

- O número de ocorrências do elemento alvo
- \$('div').length

# .eq(posição)

- `$('li').eq(2).css('color', 'red')`

# Seletores

- `$(document)`
- `$('p')`
- `$('#id')`
- `$('.classe')`
- `$('p, .cor, li')`
- `$('div p')`
- `$('div > p')`
- `$('div + p')`
- `$('div ~ p')`
- `$('p:first')`
- `$('p:last')`
- `$('li:not(li:first)')`
- `$('tr:even')`
- `$('tr:odd')`

# Seletores

- `$('tr:eq(3)')`
- `$('tr:gt(3)')`
- `$('tr:lt(3)')`
- `$(':header')`
- `$('p:contains(texto)')`
- `$('p:has(strong)')`
- `$('p:parent')`
- `$(':hidden')`
- `$(':visible')`
- `$('p[title="bla"]')`
- `$('p[title!="bla"]')`
- `$('p[title^="bla"]')`
- `$('p[title$="bla"]')`
- `$('p[title*="bla"]')`

# Manipulação de elementos

- Manipulação de atributos gerais
  - `.attr('title')` ← recuperação
  - `.attr({atributo:valor})`
    - `$('img').css('display', 'block').attr({  
    src:"arquivo.jpg",  
    alt:"Foto"  
});`
  - `.attr(atributo, valor)`
  - `.removeAttr(atributo)`

# Manipulação de elementos

- Classes
  - `.addClass(nome-da-classe)`
  - `.hasClass(nome-da-classe)`
  - `.removeClass(nome-da-classe)`
  - `.toggleClass(nome-da-classe)`

# Manipulação de elementos

- HTML
  - .html() ← recuperação
  - .html(valor) ← alteração

# Manipulação de elementos

- Textos
  - `.text()` ← recuperação
  - `.text(valor)` ← alteração



# Manipulação de elementos

- Valores
  - `.val()` ← recuperação
  - `.val(valor)` ← alteração

# Manipulação de elementos

- Conteúdos
  - .append(conteúdo)
  - .appendTo(alvo)
  - .prepend(conteúdo)
  - .prependTo(alvo)
  - .after(conteúdo)
  - .insertAfter(alvo)
  - .before(conteúdo)
  - .insertBefore(alvo)
  - .wrap(html)
  - .wrap(elemento)
  - .wrapAll(html)
  - .wrapAll(elemento)

# CSS e inspeção de elementos

- `.css(propriedade)`
- `.css(propriedade, valor)`
- `.css({propriedade: valor, propriedade: valor})`

# Eventos

- `window.onload` → `$(document).ready()`
  - Funciona tão logo a árvore esteja carregada
  - Não precisa carregar todos os componentes

# Eventos

- `.blur()`
- `.change()`
- `.click()`
- `.dblclick()`
- `.error()`
- `.focus()`
- `.keydown()`
- `.keyup()`
- `.keypress()`
- `.mousedown()`
- `.mouseup()`
- `.mouseover()`
- `.mouseout()`
- `.mousemove()`

# Eventos

- .resize()
- .scroll()
- .select()
- .submit()
- .unload()

# Efeitos

- `.show()`
- `.hide()`



# Efeitos

- `.show()`
- `.hide()`
- `.toggle()`
- `.slideDown()`
- `.slideUp()`
- `.slideToggle()`
- `.fadeIn()`
- `.fadeOut()`
- `.fadeTo()`



# Animações

- `.animate()`
- `.animate({height: "200px"}, 500)`