Desenvolvimento Web com jQuery

Web1: Introdução, Seletores e Atributos

Prof: Jacson Luiz Matte

jacson.matte@unoesc.edu.br

2019/1

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

Por que Utilizá-lo?

- Aumento de produtividade
- 86,1kb (versão 3.4.1) com inúmeras possibilidades:
  - Requisições AJAX
  - Iteração e criação de elementos DOM
  - Tratamento de eventos

- Cross-browser:
  - Funciona no Internet Explorer 6.0+, Firefox 2.0+, Safari 2.0+ e Opera 9.0+.
- Open Source

jQuery

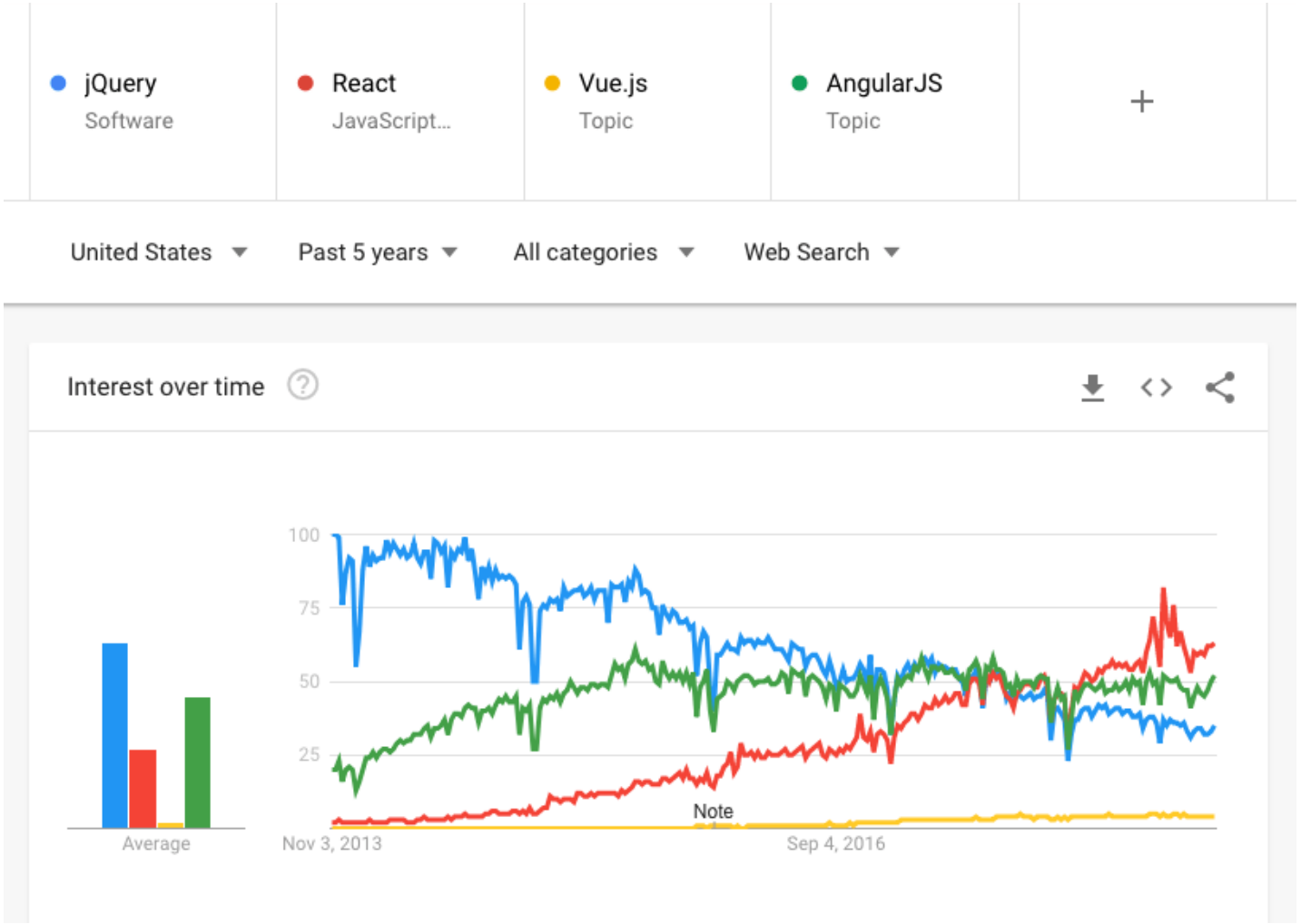
Por que Utilizá-lo?

- Diversos plugins para tarefas comuns (calendário, sliders, menus, etc.)

- Quem usa?
  - Wikipedia, Google, Netflix, Wordpress, dentre outros.

jQuery

Por que Utilizá-lo?



Deve usar o jQuery em 2019?

- A tendência é que o *jQuery* não deve ser mais usado em novos projetos que só são direcionados a navegadores modernos.
  - se o projeto depende dele por algum motivo específico, como pelo uso de plugins ou outros códigos que precisam do jQuery, definitivamente continue usando.

Deve usar o jQuery em 2019?

- Algumas bibliotecas também dependem do jQuery, como o Bootstrap. Também pode comprar modelos prontos que apenas usam *plugins jQuery*.
- Usado em projetos que devem dar suporte a navegadores antigos
- Hoje, o cenário do JavaScript mudou drasticamente. Dito isso, ainda é útil saber o que a jQuery pode ser útil.

Instalando

- Duas maneiras:
  1) Baixar a biblioteca e salvar o arquivo no seu diret de desenvolvimento
     http://jquery.com/download/
  2) Usando um CDN que contenha uma versão do jQuery.
     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.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*

Manipulação de Elementos DOM

- *Padrão da W3C que define um conjunto de objetos para o HTML e forma de acessá-los e manipulá-los*
  - *Todos os elementos HTML, assim como seus textos e atributos podem ser acessados via DOM*
  - *Podem ser modificados, removidos, e novos elementos podem ser criados*
- *O HTML DOM é uma plataforma independente de linguagem*
  - *A linguagem mais usada para acessá-lo é o JavaScript (onde o jQuery entra para facilitar as coisas)*
- *Window, Document, Table, IFrame, Link, Input Button, InputRadio e Select são alguns dos objetos DOM.*

$ (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!" );
});
```

jQuery

$(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!" );
});
```

jQuery

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 );
```

jQuery

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	$("#myId");
Nome da classe	$(".myClass");
Atributo	$("input[name='identidade']");
CSS composto	$("#conteudo ul.pessoas li");

SELETORES

- Seleciona o elemento com determinado id
  - Exemplo: `$("#primDiv").css("background-color","black");`
- Seleção pelo elemento HTML
  - Exemplo: `$("div").css("border","9px double red");`
- Elementos de determinada classe
  - Exemplo: `$(".destaque").css("fontWeight","bolder");`
- Elementos que pertençam a ambas classes
  - Exemplo: `$(".destaque. especial").css("color","red");`
- Todos elementos
  - Exemplo: `$("*").css("color","black");`
- Combinação dos anteriores com virgula
  - Exemplo: `$("div,span,p.olho").css("margin","3px 0 0 0");`

SELETORES

- Seleção de elementos-filho de um determinado elemento
  - Exemplo: `$("#primDiv").css("background-color","black");`
- Seleção pelo elemento HTML
  - Exemplo: `$("div").css("border","9px double red");`
- Elementos de determinada classe
  - Exemplo: `$(".destaque").css("fontWeight","bolder");`
- Elementos que pertençam a ambas classes
  - Exemplo: `$(".destaque. especial").css("color","red");`
- Todos elementos
  - Exemplo: `$("*").css("color","black");`
- Combinação dos anteriores com virgula
  - Exemplo: `$("div,span,p.olho").css("margin","3px 0 0 0");`

jQuery

PSEUDO - SELETORES

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

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

jQuery

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 ) {
        ...
}
```

jQuery

OBTER E MODIFICAR INFORMAÇÕES DE UM ELEMENTO

Método	Significado
.html()	Obter ou inserir conteúdo HTML. O *mesmo que .innerHTML em JS.*
.text()	Obter ou inserir conteúdo de texto
.val()	Obter ou inserir o valor de um elemento de formulário. *O mesmo que .value em JS.*
.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

jQuery

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");
```

jQuery

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() );
```

jQuery

MUITOS EXEMPLOS

- https://github.com/jquery