Aula 03 – JQuery

O que é

Surgiu em 2006, jQuery é uma biblioteca de funções JavaScript que interage com o HTML, desenvolvida para simplificar os *scripts* interpretados no navegador do cliente. Criada por John Resig, é disponibilizada como software libre e aberto.
A biblioteca possui as seguintes características:

- utiliza seletores CSS para localizar elementos componentes da estrutura de marcação HTML da página
- possui arquitetura compatível com instalação de plugins e extensões em geral
- é indiferente às inconsistências de renderização entre navegadores
- é capaz de interação implícita, isto é, não há necessidade de construção de loops para localização de elementos no documento.
- Admite programação encadeada, ou seja, cada método retorna um objeto.
- É extensível, pos admite a criação e inserção de novas funcionalidades na biblioteca existente

Como o Jquery Funciona

A partir de um arquivo local

Acesse o site jquery.com e realize o download da última versão.
Observa as versão compreensed (para ser utilizado em produção) e umpressed (para entender como a biblioteca funciona). Para inserir o jquery no cabeçalho (tags HEAD), siga o modelo a seguir:

```
<html>
	<head>
		<meta charset="utf-8">
		<title></title>
		<script type="text/javascript" src="jquery-3.2.1.min.js"></script>
	</head>
	<body>

	</body>
</html>
```

O *JQuery* também pode ser carregado no corpo (tag BODY), conforme modelo a seguir:

```
<html>
	<head>
		<meta charset="utf-8">
		<title></title>
	</head>
	<body>
		<script type="text/javascript" src="jquery-3.2.1.min.js"></script>
	</body>
</html>
```

Além de carregar a biblioteca, vamos criar um arquivo (script.js) afim de separar a linguagem de marcação (HTML) do código javascript

```
<html>
	<head>
		<meta charset="utf-8">
		<title></title>
	</head>
	<body>
		<script type="text/javascript" src="jquery-3.2.1.min.js"></script>
		<script type="text/javascript" src="script.js"></script>
	</body>
</html>
```

A partir de um servidor remoto (CDN)

Google e Microsoft disponibilizam para o uso público a biblioteca via CDN.
CDN é abreviação de Content Delivery Network (ou Rede de Distribuição de Conteúdo). É uma rede de servidores(pontos de presença) que armazenam réplicas do conteúdo de outros sites na memória (cache) e depois os entrega aos visitantes, baseando-se na localização geográfica para conectá-los ao servidor mais próximo e mais rápido, reduzindo o tempo de transferência dos dados (latência).

Google

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>
```

Microsoft

```
<head>
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js"></script>
</head>
```

Primeiros passos

Na proposta dos primeiros passos, queremos uma página webna qual exista um botão que, ao ser clicado, mude a cor de um cabeçalho de verde para vermelho.

Observe o código **arquivo-1.1.6.1.a** . Esta primeira versão não funciona. Porque ?

Observe o código **arquivo-1.1.6.1.b** . Porque funciona ?

Observe o código **arquivo-1.1.6d.** Qual a diferença em termos de sintaxe ?

Construtor jQuery $()

A estrutura básica do jquery :

```
$(elemento).ação
```

Observe o código script.js.

```
function tudoPronto() {
   alert("Oi Mundo");
}

$ (document).ready(tudoPronto);
```

A função tudoPronto é executada somente após o carregamento total da página. Observe como o elemento é informado dentro dos parênteses.
Existe outra forma de se obter o mesmo resultado:

```
function tudoPronto() {
   alert("Oi Mundo");
}

$ (tudoPronto);
```

ou

```
$ (function tudoPronto() {
   alert("Oi Mundo");
});
```

Para evitar conflitos entre a biblioteca jquery e outras bibliotecas que utilizem $() como padrão, podemos criar um alias, e fazer as chamadas ao jquery utilizando este alias.

```
var $j = jQuery.noConflict();

$j()
```

O exemplo anterior ficaria:

```
function tudoPronto() {
   alert("E continua funcionando.... ");
}

var $j = jQuery.noConflict();

$j(document).ready(tudoPronto);
```

Outra forma de evitar conflito entre bibliotecas javascript em seu projeto é chamando-se diretamente a biblitoteca jquery, conforme exemplo:

```
function tudoPronto() {
   alert("E continua funcionando ainda.... ");
}

jQuery(document).ready(tudoPronto);
```

Funções padrão e seletores jQuery

$(id) → Seletor de id acessa o elemento cujo valor do atributo id tenha sido especificado no argumento. (arquivo 2.2.1.a.html) Faz uso do “jogo-da-velha” ou *hashtag*.

$(classe) → Seletor de classe acessa os elementos cujo valor do atributo class tenha sido especificado no argumento. (arquivo 2.2.1.b.html) Faz uso do ponto final.

$(elemento) → Seletor de elementos: acessa todos os elementos especificados no argumento. (arquivo 2.2.1.c.html) Nome do elemento entre aspas.

Grupaento de seletores: acesa um agrupamento de seletores. O argumento é uma lista dos seletores a acessar. (arquivo 2.2.1.d.html)

Seletores compostos

Seletor composto é aquele constituído pela combinação de dois ou mais seletores simples. A combinação entre seletores simples para criar um seletor composto. (arquivos da série 2.2.2)

Seletores de formulário

Arquivos da série 2.2.8

Exercício

Observe o código **selectorpractice.html** e o arquivo **app.js**.

Realize as atividades propostas:

1. Selecione o elemento <h2> pela classe.
2. Selecione o primeiro parágrafo entre os parágrafos relevantes
3. Selecione o terceiro parágrafo entre os parágrafos relevantes
4. Selecione todos os parágrafos da página
5. Selecione todos os parágrafos relevantes
6. Selecione o segundo, o quarto e o sexto parágrafos relevantes
7. Selecione o sétimo parágrafo relevantes
8. Selecione o quinto, o sexto e o sétimo parágrafos relevantes.
9. Selecione os parágrafos relevantes que não sejam da classe a.