Programação cliente com Javascript

Prof. Alexandre de Oliveira Paixão

Apresentação

Você estudará sobre o desenvolvimento web no lado cliente com a linguagem de programação Javascript e a biblioteca JQuery, manipulação da árvore DOM e implementação de requisições AJAX.

Propósito

Preparação

Faça aqui o download do arquivo com todos os códigos que serão tratados neste conteúdo. Para aplicação dos exemplos, será necessário um editor de texto com suporte à marcação HTML. No sistema operacional Windows, é indicado o Notepad++. No Linux, o Nano Editor. Uma alternativa aos editores instalados no computador é a utilização de interpretadores on-line, como Codepen e JSFiddle.

Objetivos

Módulo 1

Conceitos gerais e sintaxe básica da linguagem Javascript

Reconhecer os conceitos gerais e a sintaxe básica da linguagem Javascript.

Módulo 2

Manipulando a árvore DOM com jQuery

Descrever o modo de manipulação da árvore DOM com o framework jQuery.

Módulo 3

Manipulando e tratando eventos com jQuery

Analisar a forma de trabalho com eventos com o framework jQuery.

Módulo 4

AJAX com jQuery

Empregar requisições AJAX com o framework jQuery.

Introdução

A linguagem de programação Javascript compõe as tecnologias que rodam no lado cliente no ambiente web. Essa linguagem, interpretada diretamente pelo navegador, é responsável pelos aspectos relacionados ao comportamento e à interação nas páginas HTML.

A linguagem Javascript é simples, de fácil aprendizagem e, ao mesmo tempo, muito versátil, podendo ser utilizada em páginas web, na criação de jogos e, mais recentemente, na construção de aplicativos mobile e até mesmo no lado servidor (Node.js).

Ao longo deste estudo, alguns conceitos básicos da linguagem serão revisados. Além disso, será visto como utilizar a biblioteca Jquery, a fim de otimizar o processo de programação com uso de Javascript.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

1 – Conceitos gerais e sintaxe básica da linguagem Javascript

Ao final deste módulo, você será capaz de reconhecer os conceitos gerais e a sintaxe básica da linguagem Javascript.

Incorporando o Javascript (JS) em páginas HTML

O Javascript é uma linguagem que roda no lado cliente (considerando a interação cliente x servidor). Isso significa que, para serem executados, os códigos Javascript precisam estar inseridos em uma página web, que será renderizada pelos navegadores – processo no qual o código JS inserido na página é interpretado e executado. A inclusão de códigos JS em código HTML pode ser feita de diferentes formas.

Confira, neste vídeo, a incorporação do JavaScript em páginas HTML, explorando o escopo de variáveis, a utilização de var, let e const, o uso de arrow functions e a implementação de eventos. Aprenda a tornar suas páginas interativas e dinâmicas utilizando as principais técnicas do JavaScript!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

Em relação à utilização do Javascript em páginas web, o primeiro passo para usá-lo é incorporá-lo ao documento HTML. Conheça duas maneiras de fazer isso!

1

Inserindo o código Javascript na seção <head>, ao final da página, dentro da tag <script>.

2

Incorporando um arquivo externo, contendo apenas código JS, através da tag <script> e de seu atributo “src”, onde deve ser passado o endereço do arquivo

Considere as proposições a seguir!

Recomendação

Mantenha seu código Javascript em um arquivo externo, com a extensão “.js”. Além disso, sempre que possível, incorpore o script externo ao final da página HTML – imediatamente antes do fechamento da tag <body>. Além de tornar o carregamento da página mais rápido, uma vez que ela é renderizada de cima para baixo, ou seja, ao encontrar a tag <script> logo no <head>, o navegador só continuará a carregar o restante da página após interpretar e executar todo o código JS. Essa boa prática também permite que elementos da página sejam manipulados pelo JS, pois ele será carregado apenas após todos os elementos já estarem disponíveis. Por outro lado, caso algum conteúdo ou funcionalidade da página dependa do JS para ser corretamente exibido, será necessário mantê-lo no <head>. Em resumo: avalie caso a caso.

Os exemplos citados de incorporação podem ser vistos em sequência! O exemplo apresentado à esquerda é a prática recomendada, enquanto à direita exemplifica a forma menos indicada.

HTML

HTML

Variáveis: escopo e tipos

Em uma linguagem de programação, quando falamos de escopo, estamos tratando do local em que uma variável existe dentro de um programa. Nesse sentido, em Javascript, uma variável pode existir a nível global ou a nível de bloco.

Dica

Considere como bloco todo código escrito dentro de chaves “{}” (estruturas condicionais, estruturas de repetição e funções, por exemplo).

Veja esses conceitos na prática!

HTML

Agora, siga estas etapas!

- Copie o código anterior e salve em um documento com a extensão ".html". Em seguida, abra esse arquivo no navegador.
- Abra o inspecionador de elementos e selecione a aba **console**. Recarregue a página. Veja, no inspecionador, a saída dos comandos "console.log", utilizados no exemplo.
- Leia, com atenção, os comentários existentes no próprio código. Perceba a diferença entre a sobrescrita da variável **msg**, dentro do bloco if, e da variável **msg2**, dentro do bloco de função. Por fim, veja o que acontece com a variável **msg3**.

Em linhas gerais, uma variável em Javascript tem comportamento distinto de acordo com o bloco no qual foi declarada e teve valores atribuídos. O destaque, no código anterior, vai para a variável msg2 definida e inicializada dentro da função "imprimeVariavel". Nesse caso, embora precedida pela palavra reservada "var", essa variável não possui escopo global, mas local, por ter sido definida dentro da função.

Variáveis definidas dentro de funções possuem **escopo local**, ficando restritas ao escopo da função.

Utilizando var, let e const

Como visto no exemplo anterior, é preciso ter certos cuidados na declaração, inicialização e utilização das variáveis em JS. Nesse sentido, a utilização das palavras reservadas **var**, **let** e **const**, todas associadas à declaração de variáveis, ajuda-nos a ter controle total sobre o seu escopo. A seguir, veja como e quando utilizar cada uma delas!

var

A utilização dessa palavra reservada, na declaração de variáveis, concede escopo global a elas, ou seja, variáveis declaradas utilizando var podem ser acessadas em qualquer ponto do script,

dentro e fora de blocos – com exceção dos blocos de funções, nos quais, como já mencionado, as variáveis possuem escopo apenas local.

let

A partir do lançamento do Javascript 6 –ECMAScript 6 ou ES2015 –, tornou-se possível melhorar o controle sobre o escopo de variáveis em JS com a introdução da palavra-chave let. Embora não suportada por todos os navegadores (no Internet Explorer, ela só está disponível a partir da versão Edge), essa nova palavra reservada juntou-se às demais (var e const), tendo como função, sobretudo, restringir o acesso a variáveis a nível de bloco. Uma variável declarada com let, dentro de um bloco, não pode ser acessada em nenhum outro local do script, a não ser no bloco em questão. Isso implica ainda que, caso declarada fora de um bloco, essa variável passa a ter escopo global, assim como as variáveis declaradas com var.

const

Nas linguagens de programação, uma variável declarada como constante (const) é uma variável cujo valor é fixo, ou seja, o valor atribuído a essa variável não pode ser alterado.

Em Javascript, utilizamos const para declarar uma variável como constante. O seu uso é indicado para garantirmos que o valor atribuído a uma variável não será alterado ao longo da execução de nosso programa, ou seja, para assegurarmos que alterações não previstas ou indesejadas sejam realizadas no valor de determinada variável. Observe um exemplo simples de utilização de constante!

```
const url = "www.dominio.com.br";
console.log(url); //imprimirá www.dominio.com.br
//gerará um erro, pois uma constante não pode ser 'reatribuída'
url = "www.novo-dominio.com.br";
```

Atenção!

Utilizar let, var e const permite maior controle em relação à disponibilidade das variáveis e ao seu conteúdo, evitando erros de sobrescrita ou de acesso, entre outros.

Arrow functions

Uma importante novidade foi introduzida na especificação Javascript ES6, as arrow functions, que podem ser definidas como uma forma mais simples de se criar funções em JS. Confira um exemplo em que uma arrow function que faz a multiplicação de dois números é criada e utilizada!

Javascript

Quando uma função possuir apenas uma expressão – no exemplo anterior, a única instrução é retornar à multiplicação dos dois números recebidos como parâmetros –, a sua sintaxe pode ser ainda mais simples. Veja como ficaria, nesse caso, o exemplo anterior!

Javascript

Por último, atente-se em um comparativo no qual o método JS map pode ser definido e utilizado com e sem arrow function!

Método JS map

Utilizado para aplicar uma função, recebida como parâmetro, nos elementos de um array, devolvendo ao final um novo array.

Javascript

Há muito mais a ser visto a respeito das arrow functions. Na seção **Explore+**, você encontrará algumas referências a seu respeito.

Eventos

São responsáveis por fornecer interatividade a uma página HTML, fazendo uso da linguagem Javascript.

Exemplo

O clique do mouse em um link que abre uma janela modal ou revela um conteúdo até então escondido, como um submenu, faz uso de eventos.

Podemos criar inúmeras funcionalidades, além das duas mencionadas anteriormente, fazendo uso de eventos. Veja o exemplo a seguir. Embora simples, ele contém a sintaxe necessária para a criação de eventos utilizando apenas Javascript. Copie o código e salve-o em um arquivo com extensão ".html". A seguir, abra o arquivo em um navegador e veja o que acontece.

Javascript

No exemplo anterior, foram utilizados os eventos onload, atribuído à tag <body>, e onclick, atribuído por programação à tag <h1>, com id "titulo". No código em questão, foi utilizado apenas Javascript.

Atividade

Assinale a alternativa que corresponde à afirmativa que define o que é entendido por escopo de variáveis na linguagem Javascript.

A Ao falarmos de escopo de variáveis, estamos nos referindo ao local de nosso código em que determinada

variável pode ser acessada.

B

O escopo de variáveis está relacionado ao tipo de acesso permitido a ela, podendo o escopo então ser público ou privado.

C

A partir da especificação mais recente da Javascript, o controle de escopo foi simplificado e limitado à utilização da palavra reservada const para a declaração de variáveis.

D

Em Javascript, os tipos de dados possíveis de uma variável correspondem ao seu exemplo. Logo, o escopo de uma variável pode ser uma string, um número inteiro, um objeto, entre outros.

E

O escopo de uma variável, em JS, é definido pelas palavras reservadas var, let e const.

Parabéns! A alternativa A está correta.

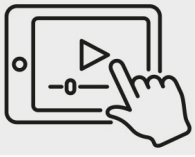
%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EO%20escopo%20de%20vari%C3%A1veis%2C%20em%20JS%2C%20diz%20respeito%20ao%20loca

Utilizando Javascript em uma página HTML

Com a prática a seguir, poderemos aplicar alguns dos conceitos básicos vistos sobre a linguagem Javascript, em especial, a inserção de código JS em páginas HTML e o escopo de variáveis. Sobre este último ponto, tal entendimento é fundamental, visto que as variáveis são um dos recursos mais usados quando programamos. Logo, é imprescindível sabermos onde declarar nossas variáveis, como atribuir e acessar os valores dessas variáveis.

Aprenda, neste vídeo, a como confeccionar uma página HTML com utilização de linguagem Javascript, aplicando a inserção de código JS e escopo de variáveis!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Roteiro de prática

Nossa prática consistirá na criação de uma página HTML simples, seguida da inserção de códigos Javascript, nos quais declaremos uma variável que deverá possuir escopo local. Para a resolução desse exercício, você precisará dos conhecimentos adquiridos ao longo do conteúdo e, em termos de ferramenta, de um editor de textos – pode ser o próprio Bloco de Notas do Windows ou o Nano Editor do Linux, ou então algo um pouco mais avançado, como o software (gratuito) Notepad++. Também precisará de um navegador – Google Chrome, Firefox, MS Edge etc. Agora, siga o roteiro a seguir:

1. Abra o editor de texto de sua preferência e insira as tags HTML básicas: Doctype, html, head, body etc.
2. Agora, insira o código JS (incorpore-o da forma como preferir) para:
    2.1. Declarar duas variáveis de escopo a nível de bloco.
    2.2. Atribuir valores – do tipo inteiro – a ambas as variáveis.
    2.3. Realizar uma operação matemática com as duas variáveis declaradas.
    2.4. Exibir o resultado da operação matemática no console (dentro do navegador, no inspecionador de elementos, aba console).

Algumas dicas adicionais sobre o roteiro: lembre-se do conceito de escopo local. Atente-se para o local e a forma como declarará suas variáveis e criará o código para realizar a operação matemática solicitada.

Como resolução, teremos o código a seguir!

Javascript

Faça você mesmo!

Considerando o fragmento de código a seguir, assinale a alternativa correspondente à sua saída – o que será exibido no comando “console.log”, na última linha.

Javascript

A Undefined.

B "Texto atribuído à variável".

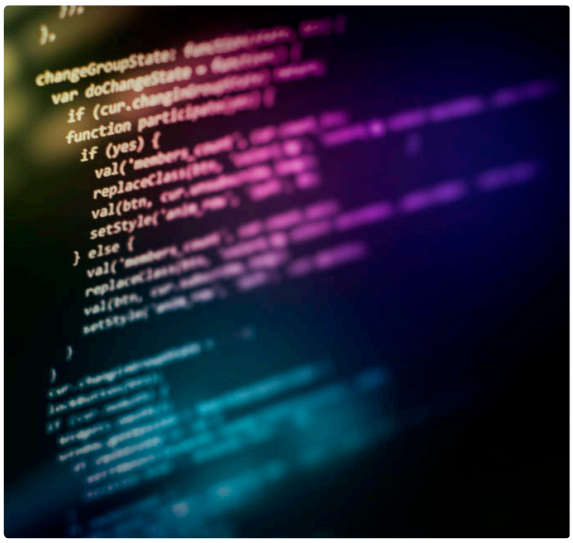
C "Novo texto atribuído à variável".

D "Texto atribuído à variável".
"Novo texto atribuído à variável".

E texto is not defined.
texto is not defined.

Parabéns! A alternativa B está correta.

%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EEm%20Javascript%2C%20as%20vari%C3%A1veis%20declaradas%20utilizando%20var%20podem



2 - Manipulando a árvore DOM com jQuery

Ao final deste módulo, você será capaz de descrever o modo de manipulação da árvore DOM com o framework jQuery.

Programação Javascript com uso de Frameworks

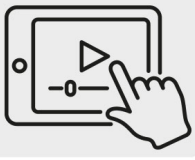
A utilização de frameworks facilita e agiliza o trabalho de programação. Por meio desses recursos, temos acesso a uma série de códigos e funcionalidades prontas, diminuindo a quantidade de códigos que precisamos escrever. Nesse contexto, veremos, a partir de agora, o framework jQuery, um dos mais conhecidos frameworks Javascript.

Frameworks

Os frameworks podem ser definidos como um conjunto de bibliotecas ou componentes reutilizáveis.

Aprenda, neste vídeo, a como manipular a árvore DOM utilizando o framework jQuery. Descubra os recursos disponíveis e como selecionar elementos de forma prática e eficiente!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



A biblioteca jQuery

É uma biblioteca Javascript rápida, pequena e rica em recursos. Essa biblioteca simplifica o processo de manipulação de documentos HTML, manipulação de eventos, animação e AJAX, com uma API fácil de usar, que funciona em vários navegadores (JQUERY, 2020).

O primeiro passo para utilizar a biblioteca jQuery é incluí-la em nosso código. Conheça duas maneiras de fazer isso!

1

Realizar o download com armazenamento local.

2

Criar um link a partir de um repositório remoto.

Além disso, é importante saber que há versões diferentes, com tamanhos diversos. Além de estar em contínua evolução, são disponibilizados diferentes pacotes: **completos**, **compactados** e **slim**. Esse último exclui alguns recursos, como os módulos AJAX e o módulo de efeitos.

> O jQuery, além de ser uma biblioteca rica em recursos, é gratuito.

O fragmento de código a seguir demonstra como incorporar a biblioteca a partir de um recurso remoto. Para utilizá-la localmente, basta fazer o download da versão desejada, modificando o endereço do atributo “src”.

Javascript

Não é necessário incorporar todas as versões demonstradas no fragmento. O exemplo serve apenas para mostrar as diferentes versões e os tamanhos disponíveis. Na prática, escolha uma versão e um tamanho e incorpore apenas o arquivo relacionado.

No fragmento anterior, foi utilizada a versão 3.6.4. Como mencionado, é possível utilizar versões anteriores. Nesse caso, tome os cuidados necessários para garantir que os recursos utilizados em sua página possuam suporte em versões mais recentes, se, em algum momento, decidir atualizar a versão.

A seguir, veremos alguns recursos jQuery relacionados à árvore DOM e a seletores.

Manipulando a árvore DOM

A árvore DOM disponibiliza uma representação estruturada do documento HTML em formato de árvore. Logo, a partir desse modelo, temos acesso a qualquer elemento de uma página. Observe!

DOM

Modelo de objeto de documento.

Árvore DOM.

A biblioteca jQuery fornece vários recursos para manipulação do DOM. Veremos alguns desses recursos a seguir.

Selecionando elementos

Com jQuery, podemos referenciar qualquer elemento da página HTML utilizando o objeto.

Javascript

Ao analisar a sintaxe, temos o método $(), que recebe como parâmetro um seletor. Tal seletor pode ser um elemento DOM, um array contendo um conjunto de elementos DOM ou um objeto. Veja o código a seguir:

Javascript

Agora, vamos analisar as instruções do código!

Primeira instrução

Na primeira instrução “console.log”, o atributo “id” da tag <p> foi passado como seletor. Perceba que, nesse caso, o nome do identificador foi precedido pelo símbolo “#”. Além disso, ambos foram englobados por aspas duplas.

Segunda instrução

Já na segunda instrução “console.log”, foi passado como seletor o nome da classe atribuída aos elementos <li>. Nesse caso, para referenciar classes, utilizamos o “.” antes do nome delas. Além disso, foram utilizadas aspas simples – só para demonstrar que podem ser utilizadas as aspas duplas e as aspas simples. Por fim, na segunda instrução, foi utilizado o método eq, que recebeu como parâmetro o número 0, ou seja, foi selecionado o primeiro elemento ao qual foi atribuída a classe “item_lista”. Cabe destacar ainda que, para acessar o conteúdo das tags, foi utilizado o método html().

Veja outra maneira de selecionar elementos. Nesse caso, utilizaremos o método :not. Vamos ao código!

Javascript

Veja que selecionamos o único elemento <li> que não possui a classe “item_lista”. Esses foram exemplos simples de como é possível selecionar elementos da árvore DOM utilizando jQuery.

Adicionando e removendo elementos

Existem alguns métodos jQuery que permitem a inserção e a remoção de elementos na árvore DOM. Vamos ver alguns exemplos!

Javascript

Para ver o resultado dos métodos utilizados, copie o código, salve-o como arquivo “html” e abra-o no navegador. Compare a estrutura inicial do código HTML e veja que, por meio dos métodos jQuery, novos elementos foram adicionados ao documento. Além disso, verifique também como ficou a árvore DOM após as modificações em questão no inspecionador de elementos.

Atenção!

Leia os comentários constantes no próprio código, no qual são passados mais detalhes sobre os métodos utilizados. Repare também que, junto com o método after, foi utilizado um novo tipo de seletor, o seletor múltiplo.

Vamos agora remover elementos de forma dinâmica, ou seja, removeremos, em tempo de execução, alguns elementos inicialmente presentes no documento HTML. Para isso, utilizaremos o código HTML resultante do código anterior!

Javascript

Manipulando conteúdo de elementos do DOM

Até aqui, vimos como manipular os elementos do DOM utilizando jQuery. Agora, veremos como manipular o conteúdo dos elementos, assim como os seus atributos. No exemplo a seguir, são demonstradas algumas formas de remover e adicionar conteúdo. Além disso, novos seletores são demonstrados.

Copie o código e salve como um arquivo “html”. Em seguida, comente o código Javascript e carregue a página no navegador. Em sequência, descomente linha a linha os códigos JS, salve e recarregue a página. Desse modo, você conseguirá observar melhor a diferença entre o conteúdo original e o conteúdo após as manipulações realizadas.

Vamos ao exemplo!

Javascript

Atividade

Por meio do framework jQuery, é possível manipular os elementos da árvore DOM e também o seu conteúdo. Nesse contexto, assinale a afirmativa abaixo que demonstra como remover o conteúdo de todas as tags <p> de uma página HTML.

A $(‘p’).find(text) = null

B $(body).text() = ‘’

C $(‘p’).html(‘’)

D $(‘p’).val(‘’)

E $(<‘p’>).html.value = ‘’

Parabéns! A alternativa C está correta.

%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EUtilizando%20jQuery%20e%20combinando%20seus%20seletores%20e%20m%C3%A9todos%2C%

Usando jQuery para manipular a árvore DOM

Como vimos, um dos valiosos recursos da lib jQuery é permitir a manipulação da árvore DOM. A seguir, colocaremos em prática o que vimos sobre esse assunto, criando, de forma dinâmica, algumas tags, incluindo seu conteúdo, em uma página HTML.

Confira, neste vídeo, o desenvolvimento de uma prática utilizando o framework jQuery para manipular a árvore DOM!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

Roteiro de prática

Para a realização desta prática, você precisará de algumas ferramentas: um editor de textos – pode ser o próprio Bloco de Notas do Windows ou o Nano Editor do Linux, ou então o software (gratuito) Notepad++. Precisará também de um navegador – Google Chrome, Firefox, MS Edge etc. Em termos de conceitos, reveja tudo o que aprendemos sobre a manipulação da árvore DOM utilizando jQuery. Você aplicará alguns dos recursos dessa biblioteca para realizar o exercício proposto. Agora, vamos ao roteiro:

1. No editor, crie a estrutura base de uma página HTML – doctype, html, head e body. Lembre-se de abrir e fechar corretamente cada tag. Além disso, não declare, inicialmente, nenhum elemento dentro da tag body.
2. Utilizando jQuery, insira, dentro da tag body, os seguintes elementos e respectivos conteúdos:
    2.1. Tag H1 (com id = “titulo-inicial”), com o conteúdo “Título Inicial”.
    2.2. Tag P, na qual deverá ser inserido um parágrafo de texto qualquer (dica: use o *lore ipsum*).
    2.3. Tag IMG, cujo atributo “src” deverá conter o endereço de uma imagem válida.
3. Por fim, salve a página no editor e a abra no navegador para ver o resultado – que deverá ser semelhante/próximo ao visto na imagem abaixo.

Resultado da prática no navegador.

A resolução da prática poderá ser feita de diferentes maneiras, como usando o código a seguir:

Javascript

Faça você mesmo!

Dentre as alternativas a seguir, assinale aquela que corresponde ao código que, fazendo uso do framework jQuery, permite adicionar uma tag <h1>, contendo o texto “Título inicial” como primeiro elemento dentro da tag <body>, considerando o fragmento de código HTML abaixo.

Javascript

A $('h1').first().html("Título inicial");

B $('body')[0].append(‘<h1>Título inicial</h1>’);

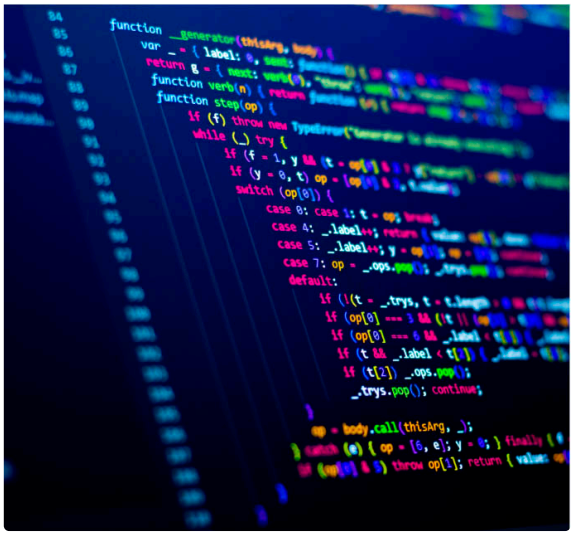
C $('body').find(‘h1’)[0].append(‘<h1>Título inicial</h1>’);

D $('body').before(‘<h1>Título inicial</h1>’);

E $(html').append(‘<h1>Título inicial</h1>’);

Parabéns! A alternativa D está correta.

%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EO%20framework%20jQuery%20possui%20v%C3%A1rios%20m%C3%A9todos%20para%20a%20inc



3 - Manipulando e tratando eventos com jQuery

Ao final deste módulo, você será capaz de analisar a forma de trabalho com eventos com o framework jQuery.

Manipulando eventos com jQuery

Na engenharia de software como um todo, quando falamos de eventos, podemos estar nos referindo às ações realizadas pelo usuário – em uma página web ou na utilização de uma tela de um software desktop ou de um aplicativo móvel por ex. – ou a eventos gerados por outros eventos/funções/programas – de forma agendada ou a partir de algum tipo de interação/estímulo, humano ou não.

Ao longo deste conteúdo, e considerando o ambiente web/lado cliente, veremos como manipular e tratar eventos utilizando jQuery.

Aprenda, neste vídeo, a manipular e tratar eventos utilizando jQuery. Descubra como trabalhar com eventos de clique, teclado, mouse e muito mais, além de explorar o método preventdefault() para personalizar o comportamento dos eventos em suas páginas HTML!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

Conceitos gerais sobre eventos em páginas HTML

Os eventos, no ambiente web, correspondem às ações realizadas pelos usuários em um elemento da página, como o clique ou o passar do mouse em um link, a seleção de opções ou a submissão de um formulário. Temos, então, eventos associados e acionados pela utilização do mouse e do teclado. Até mesmo o carregamento da página é um evento. Nesse contexto, conheça alguns termos muito comuns associados a evento!

---

fire/fired

Associado ao momento em que um evento é "disparado".

---

listener

Associado ao ato de "ouvir", monitorar, um elemento à espera do disparo de um evento.

handler

Associado à ação de manipular determinado evento.

Embora seja possível tratar todos os eventos utilizando apenas Javascript, esse processo se torna mais intuitivo e simples ao fazermos uso de jQuery. Sendo assim, veremos como manipular e tratar eventos em uma página HTML com essa biblioteca.

Eventos onload e window.onload

O evento onload ocorre quando um objeto, em uma página HTML, termina de ser carregado. Esse evento é normalmente utilizado na tag <body>, tornando possível que determinada ação seja realizada por meio de funções JS, quando a tag em questão terminar de carregar. Isso implica dizer que todas as imagens, scripts e arquivos CSS já foram carregados. Tal evento também pode ser utilizado com outras tags, como:

- <frame>
- <iframe>
- <img>
- <input type='image'>
- <link>

- <script>
- <style>

O fragmento a seguir demonstra um exemplo de sua utilização, no qual um alerta é exibido na tela assim que o <body> terminar de ser carregado.

Javascript

Assim como o onload, o evento window.onload ocorre quando determinado conteúdo é completamente carregado. Em JS, o objeto window está relacionado à janela do navegador – que contém um elemento DOM. Já o objeto document está relacionado ao DOM carregado na janela em questão. Portanto, o evento window.onload também é disparado quando todo o conteúdo da página é carregado.

Dica

Tendo em vista que a utilização do evento onload associado à tag <body> produz o mesmo resultado que o uso do evento window.onload, deve-se dar preferência ao segundo. Com isso, separamos a estrutura, no caso do <body>, da ação, já que o evento window.onload só pode ser manipulado por meio de código Javascript, sem precisar ser associado diretamente a uma tag, como acontece com o onload.

Veja como utilizar o evento window.onload!

Javascript

O evento jQuery $(document).ready()

Como vimos anteriormente, o evento window.load é disparado quando todo o conteúdo da página é carregado. Em muitas situações, desejamos alterar o comportamento desse conteúdo e, para isso, precisamos acessá-los antes do término do carregamento. Para essa função, temos o evento jQuery $(document).ready() – que pode ser abreviado da seguinte maneira.

Javascript

Com esse evento, podemos ter acesso à página assim que a árvore DOM (tags HTML) estiver disponível, o que ocorre antes de todo o conteúdo ser carregado. Observe um exemplo simples com o evento em questão!

Javascript

Para ver na prática a diferença entre os momentos em que os eventos window.load e $(document).ready() são disparados, copie o código anterior, salve-o em um arquivo HTML e abra-o no navegador. Veja que o

alerta do evento jQuery será disparado primeiro, antes de as imagens serem carregadas na página e que, somente após as imagens serem carregadas, será disparado o evento window.load.

Na biblioteca jQuery, também está disponível uma implementação do evento window.load, o $(window).on("load", function () { ... }) .

Os eventos click() e on()

Esses eventos estão relacionados ao click do mouse (quando pressionamos o botão do mouse e o liberamos na sequência) sobre um elemento na página. Embora tenham a mesma função, o "click" foi descontinuado, devendo ser substituído pelo "on". Em linhas gerais, esses eventos vinculam um manipulador (handler) ou disparador (trigger) ao evento Javascript "click" (descrito acima). Repare, a seguir, a sintaxe do evento jQuery "click" em um exemplo prático!

Javascript

O exemplo anterior, além de demonstrar o evento click em ação, adiciona um novo elemento, dinamicamente, à lista original, contida no próprio HTML. Teste o código no navegador. Repare que, ao clicar do primeiro ao quinto elemento, será exibido, no inspecionador de elementos, na aba console, o conteúdo da tag <li>.

Entretanto, isso não acontece no item adicionado dinamicamente, o "Item 6". Isso ocorre porque o evento click só reconhece os elementos já existentes na página. Para adicionar o evento de "clique" ao último elemento, é necessário utilizar outro evento, que veremos a seguir.

Diferença entre os eventos Click e On

No código apresentado a seguir, utilizamos, no lugar do evento click, o evento on. Esse evento é responsável por anexar uma função de manipulação de eventos para um ou mais eventos aos elementos

selecionados. No contexto apresentado em ambos os exemplos, confira a diferença entre o click e o on!

Click		On
É restrito ao DOM inicial.	×	É válido para novos elementos adicionados ao DOM.

Por fim, cabe destacar que o evento on não é limitado a lidar apenas com o evento click. Ele pode ser usado com outros eventos, como o submit – evento disparado quando um formulário HTML é submetido, por exemplo.

Javascript

Eventos de teclado e mouse

Em jQuery, estão disponíveis alguns eventos relacionados ao teclado e ao mouse. É possível manipular, por exemplo, o evento disparado quando determinada tecla é pressionada, ou com os eventos de mouseover, entre outros.

Veja, na prática, alguns desses exemplos. Para obter a lista completa, consulte a documentação da biblioteca!

Javascript

No exemplo anterior, foram aplicados os eventos mouseover e mouseleave sobre os elementos da lista. Veja, no console, no inspecionador de elementos, os logs sendo registrados conforme você passa o mouse sobre os elementos da lista e quando tira o mouse desses elementos. Por último, o evento keypress foi vinculado ao input do formulário. Nesse caso, pressione algumas teclas e a tecla enter, estando no campo input, e veja o resultado no console. Aqui, é exibido o código ASCII referente à tecla pressionada. Isso é útil quando precisamos, por exemplo, saber se determinada tecla foi pressionada.

Método preventdefault()

Embora não seja um evento, é importante falar sobre esse método. A sua função é, quando chamado, impedir que a ação padrão de um evento seja disparada. Vamos à prática: copie o código a seguir, salve-o e abra-o no navegador.

Código ASCII

American Standard Code for Information Interchange ou código padrão americano para intercâmbio de informação. É um código binário que codifica um conjunto de caracteres referentes às letras do alfabeto latino, sinais de pontuação, sinais matemáticos e sinais de controle.

Javascript

O comportamento normal do link, ao ser clicado, é abrir a página contida no atributo “href”. Já o comportamento esperado do botão de submissão do formulário é navegar para o local definido no seu atributo “action”.

> Você notou que nenhuma das ações padrão foi executada? Por que isso acontece?

O método event.preventDefault() serve para impedir que a ação default de determinado evento seja executada. Sua utilidade, na prática, é diversa, uma vez que nos permite não só impedir o comportamento normal, mas também definir outro comportamento para os eventos. Por exemplo, você pode usá-lo para submeter os dados de um formulário via AJAX, impedindo que a página seja recarregada e que, por padrão, a submissão do formulário o leve a outra página.

Atividade

Em algumas situações, pode ser necessário manipular os elementos ou o conteúdo da página antes que eles sejam apresentados ao usuário. Nesse sentido, assinale a alternativa que corresponda ao evento que deve ser utilizado para manipulação da página HTML antes que seu conteúdo (imagens, textos etc.) seja totalmente carregado.

A document.load

B $(document.load)()

C window.onload

D $(document).ready()

E document.load.ready()

Parabéns! A alternativa D está correta.

%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EO%20evento%20nativo%20JS%20window.onload%20ocorre%20quando%20todo%20o%20conte%C
se%20do%20%24(document).ready()%2C%20que%20pode%20ser%20usado%20em%20sua%20forma%20simplif

Trabalhando com eventos utilizando jQuery

Com jQuery, temos acesso a recursos que facilitam o trabalho de manipular e tratar eventos em uma página web. Na prática a seguir, faremos uso de alguns dos recursos existentes no framework para manipular o evento de click em um elemento HTML.

Confira, neste vídeo, a realização de uma prática empregando os eventos do jQuery!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

Roteiro de prática

Para executar a prática, que será descrita logo a seguir, você precisará de algumas ferramentas, a saber: um editor de textos – pode ser o próprio Bloco de Notas do Windows ou o Nano Editor do Linux, ou então algo um pouco mais avançado, como o software (gratuito) Notepad++. Precisará também de um navegador – Google Chrome, Firefox, MS Edge etc.

Em relação ao exercício, siga o seguinte roteiro!

1. No editor, crie a estrutura base de uma página HTML – doctype, html, head e body. Lembre-se de abrir e fechar corretamente cada tag.
2. Importe/inclua, dentro da tag <head>, o framework jQuery.
3. Insira, a seu critério, elementos dentro da tag <body>, como tags de título, parágrafo etc. Lembre-se de inserir também os conteúdos de cada uma dessas tags. Tal conteúdo será importante apenas para que a página não carregue totalmente em branco.
4. Crie, a seguir, uma tag e insira dentro dela uma tag <p> contendo um bloco de texto.

5. Após a tag acima ou dentro da tag <head>, crie uma função, utilizando jQuery, para:

    5.1. Adicionar, ao final da tag criada, uma tag <a> que deverá ter o seguinte conteúdo:
    Exibir alerta

    5.2. Utilizando jQuery, faça com que o link acima, ao ser clicado, no lugar de levar o usuário ir para outra página (definida em um atribudo “href”), exiba na tela um alerta com o texto “O novo link, inserido com jQuery, foi clicado”.

6. Ao final, salve a página no editor e a abra no navegador para ver o resultado.

A resolução da prática pode ser feita de diferentes maneiras, como usando o código em sequência!

Javascript

Faça você mesmo!

Analise o código a seguir:

Javascript

Ao ser carregada no navegador, o que essa página exibirá? Assinale a alternativa correta.

A Serão exibidos dois alertas, com os respectivos conteúdos: “Função "exibeAlerta” executada a partir do evento window.onload”; “Função "exibeAlerta" executada a partir da $(function(){}”.

B Serão exibidos três alertas, com os respectivos conteúdos: “Função "exibeAlerta" executada a partir da tag body”; “Função "exibeAlerta” executada a partir do evento window.onload”; “Função "exibeAlerta" executada a partir da $(function(){}”.

C Serão exibidos quatro alertas, com os respectivos conteúdos: “Função "exibeAlerta" executada a partir da tag body”; “Função "exibeAlerta" executada a partir da div”; “Função "exibeAlerta” executada a partir do evento window.onload”; “Função "exibeAlerta" executada a partir da $(function(){}”.

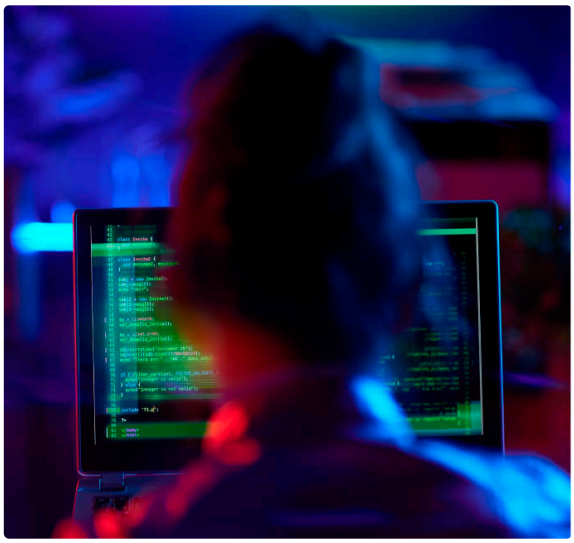
D Não será exibido nenhum alerta, apenas o conteúdo textual da tag <p>, que está dentro da tag .

E Serão exibidos dois alertas, com os respectivos conteúdos: “Função "exibeAlerta" executada a partir da

tag body” ; “Função "exibeAlerta" executada a partir da div”.

Parabéns! A alternativa A está correta.

%0A%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EConsiderando%20o%20c%C3%B3digo%20apresentado%2C%20onde%20o%20c%C3%B3digo%20Jlo%20a%20partir%20dessa%20tag.%3C%2Fp%3E%0A%20%20%20%20%20%20%20%20%20%20%20%20



4 – AJAX com jQuery

Ao final deste módulo, você será capaz de empregar requisições AJAX com o framework jQuery.

AJAX com jQuery

As requisições assíncronas são um recurso importantíssimo, até mesmo fundamental, no ambiente web, uma vez que otimizam a comunicação entre os lados cliente e servidor. A partir dessa premissa, veremos, ao longo deste conteúdo, como o framework jQuery nos possibilita, de forma bastante simples, realizar requisições AJAX.

Aprenda, neste vídeo, os fundamentos do AJAX e como utilizá-lo com o framework jQuery. Descubra também como realizar requisições assíncronas, transferir dados e otimizar a comunicação entre cliente e servidor de forma simples e eficiente!

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

O que é AJAX?

Esta sigla significa Asynchronous Javascript and XML ou Javascript e XML Assíncronos. Separando os termos que a compõem, temos:

Significado dos termos da sigla AJAX.

Isso implica a submissão de requisições, a partir de uma página web, sem interrupção em seu fluxo. Em outras palavras, a partir de requisições assíncronas, é possível requisitar recursos remotos, transferir dados e utilizá-los na página sem a necessidade de recarregá-la.

Embora constante em sua definição, o XML não é o único formato de transmissão de dados disponível em requisições AJAX. Além dele, podemos usar o JSON (que, inclusive, tem sido o formato mais utilizado atualmente), e outros tipos de dados.

A interface jQuery.ajax()

Para realizar requisições assíncronas por meio de jQuery, fazemos uso da interface jQuery.ajax(). A sintaxe desse método, como visto a seguir, é composta pela URL para a qual submeteremos a requisição e por um parâmetro, no formato de objeto, que contém informações adicionais relacionadas à requisição.

Javascript

Confira a seguir o primeiro exemplo, em que será demonstrado o código do lado cliente, o qual deverá ser salvo como arquivo html. Leia com atenção os comentários inseridos no próprio código, no qual há explicações adicionais. Observe ainda que a url de destino não existe. Logo, para tornar o exemplo real, substitua a url em questão por uma real.

Javascript

Como a url utilizada não existe, você poderá ver o erro retornado em caso de falhas (.fail) no console do inspecionador de elementos.

Esse primeiro exemplo foi bastante simples e serviu para apresentar como realizar uma requisição básica. No próximo exemplo, veremos como enviar dados na requisição e a subteremos para uma url real – um serviço on-line que responde a requisições AJAX. Vamos ao código!

Javascript

Atente-se a alguns comentários importantes sobre esse código!

- A exemplo do código anterior, é realizada uma requisição AJAX, utilizando o método HTTP POST.
- A URL para a qual a requisição foi submetida pertence a um serviço on-line que pode responder a diversos tipos de requisição. Em nosso caso, foi criado um recurso que, em seguida, foi devolvido como resposta à solicitação.
- O atributo “data” foi utilizado para enviar dados para a URL requisitada. Foram passados uma string, name, e um array, movies.
- Os dados recebidos em resposta à requisição vieram no formato JSON, sendo armazenados na variável msg. Esses resultados foram acessados por meio da notação de objetos – “objeto.atributo” e concatenados para serem inseridos dentro da onde foram exibidos na página.
- Entre os dados retornados, merece destaque o atributo “movies”, por se tratar de um array. Note no código que há dois métodos, tendo um permanecido comentado, para tratar esse tipo de estrutura: o $.map e o $.each.

A biblioteca jQuery possui ainda dois outros métodos para realização de requisições AJAX. São eles!

JSON

O Javascript Object Notation é um formato leve para armazenamento e troca de dados muito utilizado em aplicações web.

$.get()

Permite que sejam feitas requisições utilizando o método HTTP GET.

$.post()

Permite requisições HTTP POST.

Além disso, com o segundo método, é possível enviar dados na requisição. A sintaxe de ambos os métodos pode ser vista nos exemplos a seguir!

Javascript

Utilizando jQuery, AJAX e formulários

A combinação desses três componentes – jQuery, AJAX e formulários – é muito comum em páginas web: formulários para subscrição em newsletter ou cadastro e, sobretudo, filtro de listagens, como listas de produtos em sites de e-commerce, por exemplo. Veremos, de forma prática, como combinar esses três componentes. Para melhor entendimento, separamos os códigos em três seguintes blocos:

1. HTML
2. Javascript
3. JSON

Para executá-los, salve todos em uma mesma pasta.

Arquivo JSON

Esse arquivo contém alguns dados – id, fabricante, modelo, ano, cor e valor - relacionados a veículos. Salve-o como “json_data.json”. Veja!

Javascript

Arquivo Javascript

Neste arquivo, estão todas as funções e os códigos responsáveis por carregar o arquivo JSON, montar a tabela HTML e filtrar os dados. Tudo isso utilizando jQuery – olhe com atenção o código, pois algumas funções, não vistas anteriormente, foram usadas. A sintaxe e o papel dessas funções são autoexplicativos de acordo com o contexto em que foram aplicadas e em razão dos resultados que produziram. Salve esse arquivo como “scripts.js”.

Javascript

Arquivo HTML

Por último, o arquivo HTML contém o código HTML, alguns códigos CSS (que também poderiam ficar em um arquivo externo) e incorpora os scripts JS. Salve esse arquivo com o nome que desejar e com a extensão ".html".

Javascript

Seletores e métodos jQuery

Finalizando este estudo, cabem algumas considerações finais sobre os códigos utilizados ao longo dele – em especial aos relacionados ao tópico atual, no qual integramos jQuery, AJAX e HTML, também englobando os conceitos vistos ao longo dos demais conteúdos:

Seletores jQuery

Vários seletores foram usados para acessar os elementos e seus atributos. Por exemplo: no filtro de fabricante, foi verificado se um elemento radio estava marcado acessando sua propriedade "checked".

Eventos para manipulação do DOM

Usando o método append, por exemplo, o conteúdo JSON, consumido via AJAX, foi incorporado à página HTML.

Validação de dados

Alguns dados do formulário foram validados com a propriedade jQuery “val()”, que contém o valor dos elementos do formulário. Essa validação pode ser vista no filtro por ano.

Conheça outros eventos e métodos jQuery utilizados!

$.map

Para iteração em array – em nosso exemplo, foi usado para manipular os dados JSON e construir as linhas e colunas da tabela HTML.

filter

Para filtrar dados – em nosso exemplo, foi usado no filtro de anos.

:contains e :not(:contains)

Para filtrar o conteúdo de elementos – em nosso exemplo, foi utilizado nos filtros por modelo e fabricante.

Show e hide:

Para mostrar ou esconder elementos HTML – em nosso exemplo, foi utilizado para mostrar e esconder os dados de acordo com os filtros aplicados.

Embora tenha sido produzido um resultado bastante funcional com as técnicas empregadas, há uma série de outras possibilidades que não foram implementadas ou exploradas nos códigos desenvolvidos. A seguir, algumas dessas oportunidades de melhoria no código serão apresentadas.

Componentes ricos e filtros adicionais

Nos exemplos, foram utilizados alguns filtros que não cobriram todas as possibilidades, considerando os dados disponíveis. Por exemplo: não foram implementados filtros por cor e valor.

Além disso, uma funcionalidade importante ficou de fora: a ordenação dos dados. Todas essas funcionalidades podem ser implementadas e acrescentadas ao código existente, seguindo a mesma linha adotada, ou aperfeiçoadas com a utilização de componentes adicionais.

O framework jQuery possui uma biblioteca adicional, a jQuery UI, repleta de componentes visuais ricos, como slider (que poderia ser aplicado para filtrar os valores), autocomplete (que poderia ser aplicado na busca por modelo e fabricante), entre outros.

Arquitetura

Na arquitetura utilizada, os filtros foram aplicados sobre os dados a partir dos elementos HTML. Essa abordagem economiza o tráfego de dados, uma vez que não realiza novas requisições AJAX de acordo com o filtro escolhido. Por outro lado, dependendo da quantidade de dados retornados ou de outros fatores, nem sempre trazemos todos os dados de uma única vez.

É muito comum utilizarmos paginação, carregando uma quantidade inicial de dados, e, mediante demanda e interação do usuário, carregarmos mais dados. Nesse caso, os filtros em elementos HTML não nos atenderiam. Por outro lado, as técnicas e funções jQuery empregadas também podem ser empregadas para filtrar os dados diretamente na requisição AJAX, combinando-as àquelas relativas à paginação e ao carregamento gradual de informações.

Atividade

Em relação ao método jQuery $.ajax, utilizado para a realização de requisições AJAX, assinale a alternativa correta relativa à sua sintaxe e às suas propriedades.

A

A principal limitação desse método é não permitir que requisições AJAX sejam realizadas no carregamento da página, dentro, por exemplo, do evento $(document).ready().

B

Na função “done”, podemos tratar o retorno da requisição, tendo havido sucesso ou ocorrido algum erro durante o processamento, como, por exemplo, a utilização de uma URL inexistente.

C

O método $.ajax permite que sejam acessadas apenas URL remotas, ou seja, fazendo uso desse método, não é possível, por exemplo, carregarmos arquivos JSON locais.

D

Além de acessar recursos externos de forma assíncrona, esse método permite a manipulação dos dados obtidos, possibilitando que eles sejam, por exemplo, acrescentados aos demais conteúdos já existentes na página.

E

O método $.ajax facilita a realização de requisições assíncronas. Entretanto, ele possui a limitação de não permitir que sejam realizadas requisições do tipo GET, mas apenas POST.

Parabéns! A alternativa D está correta.

%0A%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EA%20sintaxe%20do%20m%C3%A9todo%20%24.ajax%20%C3%A9%20composta%20pela%20URL%

Realizando requisições AJAX com jQuery

Como vimos, o framework jQuery fornece uma série de recursos para a realização de requisições AJAX e para o tratamento do retorno de tais requisições, incluindo eventuais erros. Por mais que seja importante e estimulante lidarmos com desafios mais complexos, é fundamental sabermos de forma sólida a base, o fundamento, tanto conceitual quanto prático, de tudo o que codificamos. Nesse sentido, esta prática consistirá na criação de uma página HTML e do código Javascript, utilizando jQuery, para consumir os dados de uma API externa e exibir seus resultados.

Confira, neste vídeo, a implementação de requisições utilizando jQuery.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

Roteiro de prática

Antes de codificar, tome nota das ferramentas que você precisará para a realização dessa atividade: um editor de textos – pode ser o próprio Bloco de Notas do Windows ou o Nano Editor do Linux, ou então algo um pouquinho mais avançado, como o software (gratuito) Notepad++. Você precisará também de um navegador – Google Chrome, Firefox, MS Edge etc. Agora, vamos ao roteiro da prática:

1. No editor, crie a estrutura base de uma página HTML – doctype, html, head e body. Lembre-se de abrir e fechar corretamente cada tag.
2. Importe/inclua, dentro da tag <head>, o framework jQuery.
3. Utilizando alguns dos recursos jQuery que vimos – em especial a interface $.ajax, crie um link na página que, ao ser clicado, dispare uma função que:
    3.1. Acesse a API https://reqres.in/api/users?page=1.
    3.2. Colete os dados retornados pela API e os exiba em uma tabela HTML.

    4. Lembre-se de tratar os eventuais erros que podem acontecer ao longo do código implementado.
    5. Ao final, salve a página no editor e a abra no navegador para ver o resultado.

O resultado deverá ser parecido com o exibido na seguinte imagem!

Resultado da prática no navegador.

A resolução da prática poderá ser feita de diferentes maneiras, como usando o código a seguir!

Javascript

Faça você mesmo!

Considere o fragmento de código correspondente, onde temos uma string JSON como retorno de uma requisição AJAX, e também o método .done(), que recebe como parâmetro tal string, representada pela variável result.

A partir desse código, assinale a alternativa correta, que corresponda às lacunas do código quanto à forma de acessar o nome do aluno, a sua nota e a disciplina “Matematica”.

Javascript

A result[0].nome / result[0].nota / result[0].disciplinas[0]

B result.nome / result.nota / result.disciplinas[0]

C nome / note / disciplinas[0]

D nomeAluno.nome / notaAluno.nota / disciplina.disciplinas[0]

E id / nome / nota / disciplinas.Matematica

Parabéns! A alternativa B está correta.

%0A%20%20%20%20%20%20%20%20%20%20%20%20%3Cp%20class%3D'c-paragraph'%3EO%20resultado%20de%20uma%20requisi%C3%A7%C3%A3o%20AJAX%2C%20quando%20no%20f

O que você aprendeu neste conteúdo?

- Como incorporar Javascript em páginas HTML.
- Reconhecer os conceitos básicos (variáveis etc.) e avançados (arrow functions, eventos etc.) de Javascript.
- Utilizar o framework jQuery no desenvolvimento com Javascript.
- Utilizar o framerwork jQuery, como tratamentos de eventos e requisições AJAX.

Explore +

A especificação Javascript ES6 apresentou alguns novos recursos importantes. Embora não possua suporte em todos os navegadores, como nas versões do Internet Explorer anteriores à Edge, tal especificação deve ser estudada a fundo. Entre as novidades, destacam-se as arrow functions. Pesquise na internet como o guia de referência Javascript da Fundação Mozilla aborda esse assunto, introduzindo uma série de conceitos e diversos exemplos.

Referências

JQUERY. **O que é jQuery**. Consultado na internet em: 4 ago. 2020.

MDN WEB DOCS. **Referência Javascript**. Consultado na internet em: 4 ago. 2020.