UNINOVE

Introdução a Lógica Computacional com JavaScript

APRESENTAR AO ALUNO OS CONCEITOS BÁSICOS DA LINGUAGEM JAVASCRIPT E DA LÓGICA DE PROGRAMAÇÃO COMPUTACIONAL.

AUTOR(A): PROF. EDSON MELO DE SOUZA

1. Introdução

No passado recente as páginas de internet eram construídas com conteúdo quase que totalmente estático, ou seja, eram disponibilizadas para navegação apenas com recursos de *links* para outras páginas e envio de dados através de formulários. Em 1995 a empresa Netscape lançou o navegador Netscape 2.0 que trouxe a linguagem JavaScript (JS) incorporada, permitindo que houvesse o processamento do lado do cliente, ou seja, *client-side*, não havendo a necessidade de passar por um servidor. Desde então, o JavaScript não parou de crescer, sendo amplamente utilizado em páginas web para realizar controles, validações e aplicação de efeitos em documentos HTML.

O sucesso foi tão grande que a linguagem evoluiu e ganhou bibliotecas e *frameworks* e passou a ser utilizada em outros ambientes como servidores, desenvolvimento de jogos e aplicações móveis, entre outras utilidades, tornando-se a linguagem mais popular da web.

Você deve estar curioso (a) para aplicar, ou seja, escrever códigos que possam ser executados no seu navegador. Mas, antes de começarmos a escrever códigos, é necessário estudar alguns conceitos fundamentais. Após estudarmos esses conceitos, vamos então ver como criar códigos em JavaScript e executa-los no navegador, utilizando os algoritmos apresentados neste tópico.

SAIBA MAIS!

JavaScript - https://www.javascript.com/ (https://www.javascript.com/)

W3Schools - https://www.w3schools.com/js/ (https://www.w3schools.com/js/)

2. Lógica Computacional (Programação)

A Lógica de Programação pode ser entendida como uma sequência de passos lógicos para atingir um determinado objetivo, sendo esses passos conhecidos como algoritmos.

A construção de um algoritmo não depende do conhecimento de uma linguagem de programação específica, pois as sequências lógicas, acompanhadas de regras (instruções), podem ser adaptadas para qualquer linguagem. Um exemplo bastante comum são os idiomas, pois, um brasileiro que fala e escreve em Língua Portuguesa pode aprender outro idioma, como o Inglês ou Alemão e, neste caso, vai utilizar o mesmo raciocínio (algoritmo) para desenvolver uma conversa ou escrever um texto, mas em outro idioma (linguagem de programação), ou seja, vai mudar apenas a sintaxe, pois o conteúdo (ideia/assunto) será o mesmo.

2.1. Desenvolvimento de Algoritmos

Para desenvolver um algoritmo são utilizados os recursos da linguagem natural e do pseudocódigo, ou seja, uma linguagem natural tem a intenção de mostrar qual é a ideia e os caminhos a serem seguidos para criação de um programa.
Existem algumas regras que devem ser seguidas, sempre com simplicidade e objetividade para descrever os passos ou sequências de instruções para um programa. Vejamos algumas regras:

- Considerar, SEMPRE, que o leitor NÃO é da área de informática;
- Descrever de maneira simples e utilizar apenas um verbo por frase;
- Usar frases curtas e objetivas;
- Não usar palavras com duplo sentido.

Veja os exemplos a seguir:

- CALCULAR o saldo da conta bancária
- EFETUAR a contagem do estoque

O pseudocódigo é uma representação entre a linguagem natural e um código de programa real. Ele tem a função de apresentar, de forma mais próxima do código real, o que deverá ser codificado para descrever um algoritmo. A seguir um exemplo de pseudocódigo para realizar uma soma entre dois números:

```
1. variáveis
2. numero1, numero2, soma
3.
4. início
5.         leia numero1
6.         leia numero3
7.
8.         soma = (numero1 + numero2)
9.
10.        escreva soma
11.
12. fim
```

2.2. Separando Tarefas

Quando uma tarefa necessita ser descrita, é importante considerar que um processo ou ação pode conter mais de uma fase e, portanto, devem ser consideradas as três fases fundamentais para divisão de um problema, a saber:

Legenda: FASES DE UM ALGORITMO

De forma geral as três fases são: *Entrada – Processamento – Saída*

ENTRADA: são os dados necessários informados inicialmente

PROCESSAMENTO: são os procedimentos utilizados para que o objetivo final seja alcançado. *Obs.: Nesta fase podem ocorrer outras divisões de tarefas*

SAÍDA: o resultado esperado

Vamos analisar o seguinte problema: *No final de cada semestre é necessário apresentar uma média por aluno com base nas provas realizadas durante o período letivo. Considerando que são realizadas quatro (04) provas, calcule o valor da nota final.*

Com base na interpretação do texto, vamos desenvolver nosso algoritmo. Mas, antes de tudo, é muito importante identificar quais são os elementos envolvidos.

O cálculo de uma média simples é descrito como:

> A soma de todos os elementos *dividido* pelo número de elementos somados.

Então, para montarmos o algoritmo em linguagem natural para este problema, vamos, inicialmente, fazer três perguntas:

1. Quais são os dados de entrada?

R.: Nota 1, Nota 2, Nota 3 e Nota 4

2. Qual será o mecanismo de cálculo a ser utilizado?

R.: Nota 1 + Nota 2 + Nota 3 + Nota 4 / 4

3. Qual será o valor de saída?

R.: A média calculada, ou seja, apenas um número.

Agora vamos montar o algoritmo efetivamente:

Receber a nota da prova 1 Receber a nota da prova 2 Receber a nota da prova 3 Receber a nota da prova 4	Entrada
Somar todas as notas Dividir o resultado da soma por 4	Processamento

Mostrar o valor obtido	Saída

O pseudocódigo para este algoritmo é escrito da seguinte forma:

variáveis nota1, nota2, nota3, nota4, soma, resultado	Declara variáveis para armazenar os valores que serão usados no cálculo da média
início leia nota1 leia nota2 leia nota3 leia nota4 soma = (nota1 + nota2 + nota3 + nota4) resultado = soma / 4 escreva resultado fim	Etapas realizadas para a realização do cálculo e exibição do resultado

Observação: Variáveis são espaços reservados na memória RAM do computador ue tem a função de guardar/armazenar dados que serão utilizados durante a execução programa. Elas podem ter valores de diversos tamanhos e tipos, tais como números inteiros, números reais, caracteres, frases, entre outros.

2.3 O JavaScript: Sintaxe e Aplicação

Como dito anteriormente, o JavaScript é uma linguagem de script que é executada do lado do cliente (navegador) e, portanto, deve ser "embutida" no código HTML de uma página.

Para criarmos nossas páginas HTML é necessário utilizar um editor de textos. Existem diversos editores que são gratuitos (*Freeware* ou *OpenSource*) e também os pagos. No vídeo a seguir são mostrados alguns editores explicando como realizar o *download* e também como criar e salvar um arquivo no formato HTML.

Os editores facilitam muito o desenvolvimento, pois possuem recursos que melhoram a produtividade como: preenchimento automático de comandos, diferenciação por cores entre os comandos e tags, alinhamento, etc.

Não existe um editor melhor ou pior, existe aquele em que você melhor se adapta, até mesmo o Bloco de Notas que já vem instalado no Windows, ou então o vi ou nano para Linux, já são suficientes para criar suas páginas de internet.
Então vamos ao vídeo!

Agora que você já tem um editor instalado, vamos colocar a "mão na massa". O código a seguir mostra como incluir um código JavaScript em um arquivo HTML.
Para facilitar a compreensão, os termos estão incluídos diretamente nos comentários do código fonte.

ATENçãO!!!

É importante destacar que os códigos escritos em JavaScript possuem uma sintaxe (forma de escrever) que precisam ser seguidas, caso contrário o código não será executado.
Portanto, fique atento!

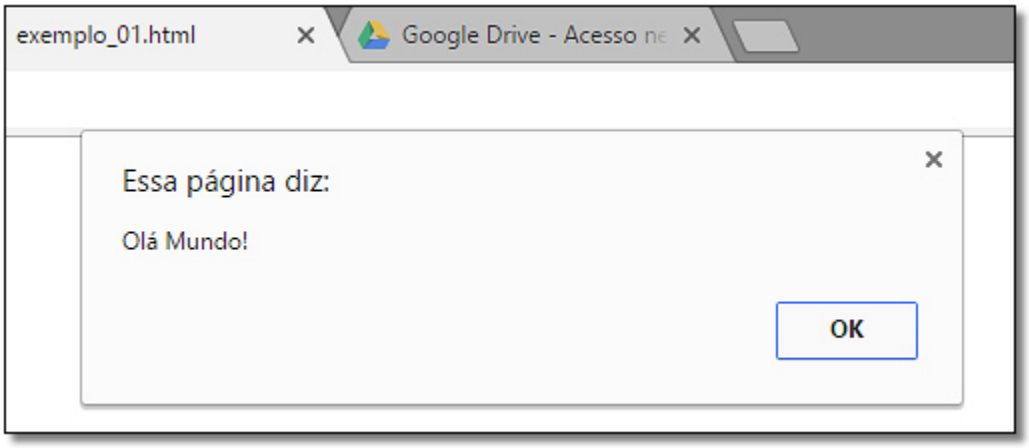
No exemplo a seguir, o código JavaScript tem a função de exibir uma caixa de mensagem, na tela do navegador, no momento em que a página HTML é requisitada, ou seja, executada pelo navegador. Vejamos o código:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.         <title></title>
5. </head>
6. <body>
7.         <!--
8.                 Aqui começa a inclusão do código JavaScript.
9.                 Todo código escrito em JavaScript de estar cercado pela
10.        -->
11.        <script type="text/javascript">
12.                // Este comando "alert" tem a função de apresentar uma c
13.                alert("Olá Mundo!")
14.        </script> <!-- Fim do bloco do JavaScript -->
15. </body>
16. </html>
```

O código em JavaScript é incluído em um arquivo HTML utilizando a tag “script” (linha 11), a qual deve iniciar o bloco de código e fecha-lo, incluindo uma barra antes da palavra “script” (linha 14). Ou seja, todo código JavaScript deve estar contido em um bloco de códigos e este bloco pode estar em qualquer parte do documento HTML.

No exemplo anterior, linhas de 11 a 14, o código foi inserido diretamente após a tag “body”, o que significa que será executado assim que a página carregar, ou seja, o interpretador da linguagem JavaScript “percebe” ou “identifica” que há um código que necessita ser executado, pois está dentro um bloco “script”. Na linha 13 a palavra “alert” tem a função de criar uma caixa de mensagem e apresentar o texto que está indicado dentro dos parênteses e entre aspas.

Na figura a seguir é possível verificar o resultado da execução do código após a renderização (processamento) pelo navegador.



IMPORTANTE!!!

Toda linha de comando em JavaScript deve ser finalizada utilizando o “;” ponto-e-vírgula.

No próximo exemplo vamos implementar, ou seja, criar o código para solucionar o problema do cálculo da média das notas dos alunos apresentado anteriormente.

```
<!DOCTYPE html>
<html>
<head>
        <title></title>
</head>
<body>

        <!-- Aqui começa a inclusão do código JavaScript. -->
        <script type="text/javascript">

                // Declaração das variáveis para armazenar o valor das n
                // Obs.: uma varíavel é um espaço na memória do computad
                var nota1 = 6;
                var nota2 = 8;
                var nota3 = 7;
                var nota4 = 9;

                // Declaração de uma variável para realizar a soma e arm
                var soma = (nota1 + nota2 + nota3 + nota4);

                // Decalaração de uma variável para armazenar o resultad
                var resultado = (soma / 4);

                // A instrução abaixo imprime o valor obtido no cálculo
                // Perceba que foi inserdio um texto, entre aspas, e uti
                // para "concatenar", ou seja, juntar um texto com um re
                document.write("Média: " + resultado);

        </script>
        <!-- Fim do bloco do JavaScript -->

</body>
</html>
```

Você pode perceber que, após executar o arquivo, o valor da média foi exibido no navegador. Perceba que para mostrar o resultado foi utilizado a instrução *document.write*. Este comando tem a função de escrever em uma página HTML. O *document* é um objeto JavaScript que permite acesso à diversos outros elementos do HTML, inclusive à própria página.

Para exibir o resultado foi “concatenado” um texto, ou seja, juntamos um texto com o resultado obtido pelo cálculo utilizando o símbolo de concatenação “+” (adição), pois esse símbolo permite “juntar” termos.

A figura a seguir mostra o resultado após o processamento da página.

SAIBA MAIS!

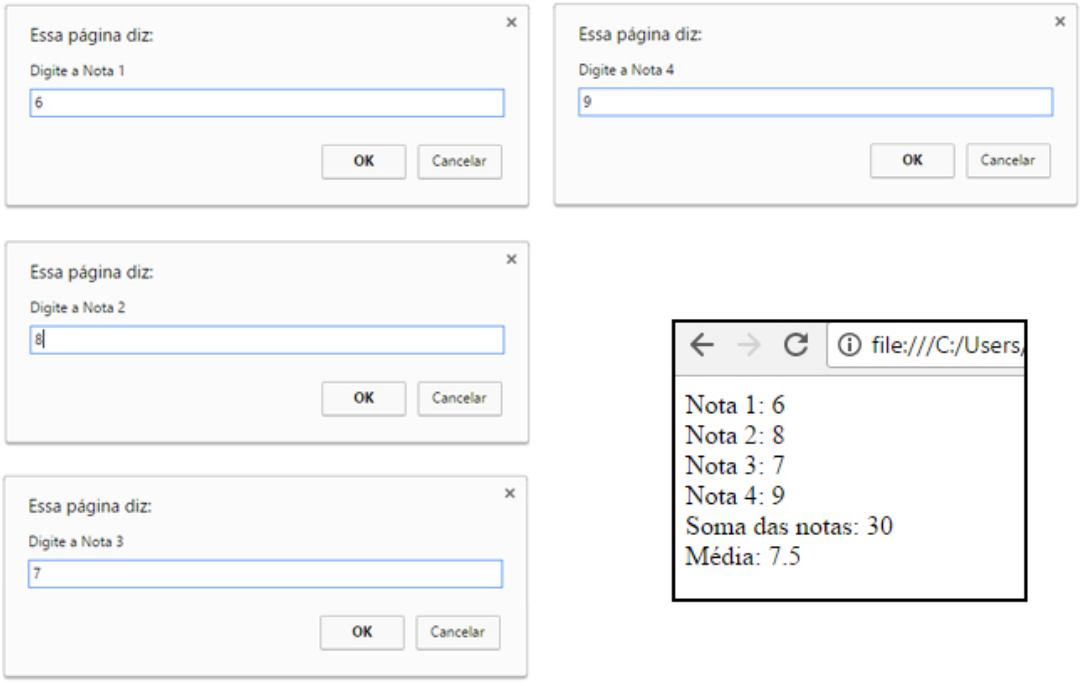
Para saber mais sobre o objeto *document*, acesse esse link: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model (https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model)

Mas, e se você desejasse fazer outro cálculo, teria que digitar tudo novamente? A resposta é SIM! Mas, como o JavaScript é uma linguagem dinâmica, vamos incrementar nosso programa e permitir que os valores sejam informados (digitados) pelo usuário, assim, ao invés de colocarmos os valores fixos, vamos permitir que qualquer valor seja informado. Desta forma, nosso programa, seguindo o algoritmo do cálculo da média, para o problema apresentado, ficará dinâmico.

Vejamos então como ficou a alteração do programa que permite a entrada de dados dinamicamente:

```
<!DOCTYPE html>
<html>
<head>
        <title></title>
</head>
<body>

        <!-- Aqui começa a inclusão do código JavaScript. -->
        <script type="text/javascript">

                // Declaração das variáveis para armazenar o valor das n
                // Aqui foi utilizado um conversor de tipo de dados pars
                // A instrução prompt() recupera um valor digitado, mas,
                // sempre um texto e, por isso, é necessário converter o
                // O Float significa que o número pode ter casas decimai
                var nota1 = parseFloat(prompt("Digite a Nota 1"));
                var nota2 = parseFloat(prompt("Digite a Nota 2"));
                var nota3 = parseFloat(prompt("Digite a Nota 3"));
                var nota4 = parseFloat(prompt("Digite a Nota 4"));

                // Declaração de uma variável para realizar a soma e arm
                var soma = (nota1 + nota2 + nota3 + nota4);

                // Decalaração de uma variável para armazenar o resultad
                var resultado = (soma / 4);

                // A instrução abaixo imprime o valor obtido no cálculo
                // Perceba que foi inserdio um texto, entre aspas, e uti
                // para "concatenar", ou seja, juntar um texto com um re
                // Para melhorar a visualização, as notas digitadas (ent
                // Foi utilizada a tag "br" para gerar uma quebra de lin
                document.write("Nota 1: " + nota1 + "<br>");
                document.write("Nota 2: " + nota2 + "<br>");
                document.write("Nota 3: " + nota3 + "<br>");
                document.write("Nota 4: " + nota4 + "<br>");
                document.write("Soma das notas: " + soma + "<br>");
                document.write("Média: " + resultado);

        </script>
```

```
40.         <!-- Fim do bloco do JavaScript -->
41.
42. </body>
43. </html>
```



Você percebeu como agora ficou muito mais interessante, pois dá liberdade para o usuário digitar valores sem a necessidade de alteração no código, bastando apenas recarregar a página.

Estas são apenas algumas demonstrações do que pode ser criado utilizando o JavaScript, basta aprofundar os conhecimentos e criar seus próprios algoritmos.

2.4. Teste de Mesa

A técnica do teste de mesa, no primeiro momento, aparenta ser insignificante ou até mesmo perda de tempo, mas é uma etapa fundamental para verificar se o algoritmo atende às necessidades e se não há falhas na sua descrição. Desta forma, antes de transformar o algoritmo na linguagem de uma máquina, é possível verificar se tudo está ou não correto.

O exemplo a seguir mostra como realizar o Teste de Mesa para o problema anterior.

Teste de Mesa: cálculo da média

Nota 1	Nota 2	Nota 3	Nota 4	Soma	Operação	Média (Resultado)

Nota 1	Nota 2	Nota 3	Nota 4	Soma	Operação	Média (Resultado)
6	8	7	9	30	30 divido por 4	7,5
9	8	8	9	34	34 divido por 4	8,5
5	2	3	6	16	16 divido por 4	4,0

Para realizar a codificação em uma linguagem que o computador “entenda”, é fundamental adotar os procedimentos para realização dos Testes de Mesa, dessa forma é possível garantir uma codificação mais tranquila e eficiente.

SAIBA MAIS!

Algoritmos (KHAN) - https://pt.khanacademy.org/computing/computer-science/algorithms (https://pt.khanacademy.org/computing/computer-science/algorithms)

Lógica de Programação Computacional - https://becode.com.br/melhor-forma-de-aprender-logica-de-programacao/ (https://becode.com.br/melhor-forma-de-aprender-logica-de-programacao/)

Resumo

Neste tópico foi abordada uma rápida introdução ao Javascript e aos Algoritmos, destacando a importância dos passos a serem seguidos na construção de programas utilizando a lógica computacional. Apresentou também exemplos da utilização do JavaScript, mostrando algumas funções e também a sintaxe para driação dos códigos.

Conclusão

Programar um computador é uma tarefa que exige, além do conhecimento técnico de uma ou mais linguagens de programação, um bom entendimento sobre a construção de algoritmos, pois, programar é o “ato de transferir uma ideia” para o computador, de forma lógica que ele possa realizar as tarefas determinadas.

ATIVIDADE FINAL

O Javascript foi, inicialmente, criado com o objetivo de:

A. Processar dados do lado do cliente.

B. Processar os dados do lado do servidor.

C. Melhorar a eficiência das páginas web.

D. Deixar os navegadores mais rápidos.

As regras de um algoritmo devem ser escritas utilizando:

A. Um verbo por frase.

B. Um pronome por frase.

C. Uma frase por verbo.

D. Um pronome por verbo.

O Teste de Mesa deve ser realizado para:

A. Verificar se um algoritmo foi escrito corretamente.

B. Verificar o tempo que será gasto para codificação de um programa.

C. Testar se os valores de "Entrada" estão corretos.

D. Não tem muita importância, podendo deixar de ser realizado.

REFERÊNCIA

MANZANO, JOSÉ AUGUSTO N. G. Lógica para Desenvolvimento de Programação de Computadores. São Paulo: Érica, 2016.

MICHAEL, Morrison. Use a cabeça! Javascript. 2008.

SILVA, Maurício Samy. JavaScript: guia do programador. São Paulo: Novatec, 2010.

SOUZA, João. Lógica para ciência da computação. Elsevier Brasil, 2015.