



SENAI AVAK BEDOUIAN

EDUARDA MASSON COSTA

ATIVIDADE JAVASCRIPT - LISTA 2
EXPLICAÇÃO DA RESOLUÇÃO

BIRIGUI - SP
2024

RESUMO

Este documento tem como finalidade expor explicações do processo de resolução de exercícios solicitados no curso de Desenvolvimento de Sistemas da instituição supracitada. A partir dele, é possível realizar uma análise mais profunda e eficiente dos conceitos referentes a JavaScript e suas aplicações gerais. As resoluções, além de auxiliarem nas pesquisas, permitem um aprofundamento nos estudos sobre a Linguagem da Programação e os mais diversos âmbitos que nisso agrega.

Palavras-chave: JavaScript. Resolução. Exercícios. Linguagem da Programação.

ABSTRACT

This document aims to explain the process of solving exercises requested in the Systems Development course at the aforementioned institution. From there, it is possible to carry out a deeper and more efficient analysis of the concepts relating to JavaScript and its general applications. The resolutions, in addition to assisting in research, allow for in-depth studies on Programming Language and the most diverse areas that this brings.

Keywords: JavaScript. Resolution. Exercises. Programming Language.

SUMÁRIO

INTRODUÇÃO	5
DESENVOLVIMENTO	6
CONCLUSÃO	14
REFERÊNCIAS BIBLIOGRÁFICAS	15

1. INTRODUÇÃO

O documento tem como finalidade apresentar a resolução de exercícios utilizando a linguagem de programação JavaScript, abordando conceitos fundamentais e a aplicação prática de suas funcionalidades. Através da resolução de questões propostas, busca-se facilitar o entendimento dos principais recursos da linguagem, contribuindo para o aprendizado e aprimoramento dos conhecimentos técnicos. A prática de codificação é essencial para a consolidação do conteúdo teórico e para a familiarização com os padrões de desenvolvimento em JavaScript, permitindo uma melhor compreensão das atividades solicitadas no contexto acadêmico e profissional.

2. DESENVOLVIMENTO

2.1 Exercício: Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.

Resolução e explicação: Primeiramente, foi criada uma variável para a lista, que foi preenchida com números de 1 a 5. Logo, utilizou-se o comando “for” a fim de percorrer os elementos, que se referem a posição da variável correspondente (0,1,2,3,4,5). Por fim, a saída formatada, feita através do “alert” e (lista[contar]) (para exibir os números um por um).

2.2 Exercício: Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.

Resolução e explicação: Primeiramente, foi criada uma variável para a lista, que foi preenchida com números de 1 a 10. Logo, foi feita outra variável para a inversão desta, que foi completada pelo código “lista.reverse()”. Tal código tem como objetivo inverter um array diretamente, tornando o primeiro elemento o último, e o último, o primeiro. Ademais, este comando foi encontrado em um site, o qual será mencionado nas referências bibliográficas. Por fim, a saída formatada, feita através do “alert” e a segunda variável criada, exibindo a lista invertida.

2.3 Exercício: Faça um Programa que leia 4 notas, mostre as notas e a média na tela.

Resolução e explicação: Primeiramente, foi criada uma variável para a lista, que foi preenchida com números de 7 a 10 (notas solicitadas). Logo, foi feita outra variável para a soma, a qual possui o valor igual a 0. Após isso, utilizando o método do for como explicado anteriormente, os elementos foram percorridos e

por meio do código `soma = (nota[contar]) + soma`, a variável `soma` passou a valer o primeiro elemento, além de somar com o próximo. Em seguida, fora do `for`, criou-se um novo código, sendo este: `media = soma / nota.length`. Ele foi feito com a finalidade de calcular a média dos valores, contendo o valor da soma dos valores (feito anteriormente) e dividindo ele (/) pela quantidade de elementos da lista, realizado através da variável `.length`, a qual tem como responsabilidade retornar o tamanho de um array e contar os elementos ali encontrados. Ademais, este comando foi encontrado em um site, o qual será mencionado nas referências bibliográficas. Por fim, realizou-se a saída formatada, feita através do `alert` e as variáveis das notas e médias respectivamente.

2.4 Exercício: Faça um Programa que leia um vetor de 10 caracteres, e diga quantas consoantes foram lidas. Imprima as consoantes.

Resolução e explicação: De forma geral, o código remove vogais da lista inicial e exibe as letras que restaram, além de mostrar a lista final com as consoantes restantes. Primeiramente, foi criada uma variável para a lista, que foi preenchida com uma lista de letras de `"a"` a `"j"`. Da mesma forma, foi feita para as vogais. Além disso, uma variável nomeada `"contar"` foi feita valendo 0, que será utilizada posteriormente. Em seguida, o comando `"while"` executa enquanto o `"contar"` for menor que 5 (números de vogais correspondentes). Foi criada, também, outra variável com a finalidade de retornar, armazenando o índice da vogal atual (`vogais[contar]`) na lista. Para isso, utilizou-se o comando `".indexOf()"` (comando foi encontrado em um site, o qual será mencionado nas referências bibliográficas), que retorna o índice da primeira ocorrência do valor fornecido, retornando em -1 se o valor não for encontrado. Se a vogal não estiver na lista (diferente de -1 `"!="`), `indexOf` retorna -1. Se a vogal for

encontrada (retornar $\neq -1$), ela é removida da lista usando splice, que remove o elemento na posição retornar. Utiliza-se o operador “increment” (++) (comando foi encontrado em um site, o qual será mencionado nas referências bibliográficas), que incrementa seu operando e retorna o valor antes ou depois do incremento. Dessa forma, “contar” é incrementado em 1 para passar para a próxima vogal no próximo ciclo. Por fim, é exibido as consoantes (lista.length, para contar os elementos da lista), e a lista, que está somente com as consoantes restantes.

2.5 Exercício: Faça um Programa que leia 20 números inteiros e armazene-os num vetor. Armazene os números pares no vetor PAR e os números ÍMPARES no vetor ímpar. Imprima os três vetores.

Resolução e explicação: Primeiramente, foram criadas três variáveis do tipo lista, sendo a primeira com números de 1 a 20 e as duas últimas vazias, sendo uma para os números pares e a outra para os números ímpares. Logo, através do “for”, a lista foi percorrida, e por meio do “if”, que indica condição, um comando para os números pares foi criado. Caso o número dentro da lista principal for par, adicionar este a lista vazia (par). O mesmo processo foi feito com o dos números ímpares, com a lista “ímpar”, utilizando o “else” (caso não aconteça o anterior, adicionar na lista de números ímpares). Por fim, a saída formatada, feita através do “alert” e as variáveis criadas, exibindo a lista original, os números pares e os ímpares, respectivamente.

2.6 Exercício: Faça um Programa que peça as quatro notas de 10 alunos, calcule e armazene num vetor a média de cada aluno, imprima o número de alunos com média maior ou igual a 7.0.

Resolução e explicação: Primeiramente, foram declaradas as variáveis, sendo uma das notas, outra da soma delas e a outra

para contar o número de alunos com média maior que 7. Logo, o “for” faz com que o pedido das notas vindas posteriormente sejam repetidas 10 vezes ao usuário, um para cada aluno. Além disso, quando multiplica-se a soma por 0, faz com que seja redundante e zera a variável para o próximo loop. Em seguida, são solicitadas as quatro notas ao usuário, através do prompt e convertidas para números (que podem ser quebrados) com `parseFloat`, como ensinado em sala de aula. Logo após, os comandos para o cálculo da soma das notas e para a média são expostos, dividindo eles por 4, o valor solicitado. Se a média for maior ou igual a 7 ($media \geq 7$), incrementa 1 a variável “maior” vazia. Quando finalizado todos os loops, o valor do número de alunos estará na variável “maior”, que por conseguinte, é exibida ao final do código pelo comando “alert”.

2.7 Exercício: Faça um Programa que leia um vetor de 5 números inteiros, mostre a soma, a multiplicação e os números.

Resolução e explicação: Primeiramente, foram criadas três variáveis do tipo lista, sendo a primeira com números de 1 a 5, outra vazia e a última valendo 1, sendo uma para a soma e a outra para a multiplicação, respectivamente. Logo, através do “for”, a lista foi percorrida, e por meio do “ $soma = (lista[contar]) + soma$ ”, os valores supracitados foram somados e a variável passou a valer tal valor. A mesma proposta aconteceu com a multiplicação, mas utilizando o símbolo “*” que refere-se a multiplicação. Vale ressaltar que a variável começa valendo 1 pois se valesse 0, por exemplo, a conta seria zerada. Por fim, a saída formatada, feita através do “alert” e as variáveis criadas, exibindo a lista original de números, a soma e a multiplicação.

2.8 Exercício: Faça um Programa que peça a idade e a altura de 5 pessoas, armazene cada informação no seu respectivo vetor. Imprima a idade e a altura na ordem inversa a ordem lida.

Resolução e explicação: Primeiramente, foram criadas as variáveis vazias, correspondentes a idade e a altura. Logo, através do “for(il = 0; il < 5; il++)”, a repetição para solicitar as perguntas é feita 5 vezes (o comando “++” foi encontrado através de pesquisas que podem ser encontradas nas referências bibliográficas). Após isso, foi solicitado, por meio do parseInt e do parseFloat (para números quebrados), a altura e a idade. Logo, foram adicionados estes as variáveis vazias, por meio do “.push”, o qual adiciona o item na lista. Fora do “for” duas novas variáveis foram criadas, a fim de inverter os valores, utilizando o “.reverse()” que, assim como supracitado, inverte os valores. Por fim, foi exibido a saída formatada.

2.9 Exercício: Faça um Programa que leia um vetor A com 10 números inteiros, calcule e mostre a soma dos quadrados dos elementos do vetor.

Resolução e explicação: Primeiramente, foram criadas as variáveis da lista com números de 1 a 10 e outra para a soma valendo 0. Logo, utilizou-se o mesmo esquema do “for” (como supracitado), para percorrer a lista e realizar o cálculo posterior: soma = soma + (lista[contar]) ** 2 (dentro da lista percorrida, realiza-se o quadrado do número (**2, como ensinado ao decorrer do curso) e soma todos eles ao final (+ soma). Por fim, foi exibido a saída formatada.

2.10 Exercício: Faça um Programa que leia dois vetores com 10 elementos cada. Gere um terceiro vetor de 20 elementos, cujos valores deverão ser compostos pelos elementos intercalados dos dois outros vetores.

Resolução e explicação: Primeiramente, criaram-se 3 variáveis, duas para listas já feitas e outra vazia para intercalar (essa lista foi adicionando um elemento de uma lista e depois outra). Em

seguida, com o “for”, ele percorreu uma das listas e adicionou na lista vazia (.push(), como aprendido nas aulas), o elemento da primeira lista. Após isso, percorreu a outra lista e adicionou o elemento dela, até acabar todos, intercalando-os. Por fim, a saída formatada foi exibida com a lista “intercalar”.

2.11 Explicação: Altere o programa anterior, intercalando 3 vetores de 10 elementos cada.

Resolução e explicação: Para a resolução desse exercício, foi somente preciso adicionar uma variável no começo referente a lista nova com mais 10 valores. Logo em seguida, percorrer ela no mesmo “for” para intercalar com as outras. Por fim, exibir a saída formatada.

2.12 Explicação: Foram anotadas as idades e alturas de 30 alunos. Faça um Programa que determine quantos alunos com mais de 13 anos possuem altura inferior à média de altura desses Alunos.

Resolução e explicação: Primeiramente, foram criadas duas variáveis, uma para a soma e outra para a quantidade de alunos com altura inferior à média. Logo, através do “for(i = 0; i < 30; i++)”, a repetição para solicitar as perguntas é feita 30 vezes (o comando “++” foi encontrado através de pesquisas que pode ser visto nas referências bibliográficas). Além disso, “soma = soma + altura” tem a função de adicionar a altura à soma total. Após isso, foi criada outra variável (var media = soma / 30) para calcular a média da altura dos alunos. Em seguida, foi realizado outro “for” a fim de reavaliar a lista para contar alunos com mais de 13 anos e altura abaixo da média. Depois disso, uma condição foi informada para saber a quantidade de alunos com altura inferior à média, para especificar a idade do estudante juntamente com a análise sugerida. O comando “&&”, pode substituir o “and”,

trazendo a ideia de duas condições. Este comando também foi estudado através de pesquisas, cujo link se encontra nas referências. Desse modo, ele adiciona um número para a variável maior (maior = maior + 1). Por fim, exibir a saída formatada.

2.13 Exercício: Faça um programa que receba a temperatura média de cada mês do ano e armazene-as em uma lista. Após isto, calcule a média anual das temperaturas e mostre todas as temperaturas acima da média anual, e em que mês elas ocorreram (mostrar o mês por extenso: 1 – Janeiro, 2 – Fevereiro, . . .).

Resolução e explicação: Primeiramente, foram criadas duas variáveis, uma vazia (para a temperatura) e outra contendo as strings com os meses do ano. Logo em seguida, através do “for(i = 0; i < 12; i++)”, a repetição para solicitar as perguntas é feita 12 vezes (o comando “++” foi encontrado através de pesquisas que pode ser visto nas referências bibliográficas). Em seguida é feita a pergunta ao usuário, e este é adicionado à lista “temperatura”, com o comando “push()”, ensinado em sala de aula. Logo em seguida, foi declarada a variável “soma” vazia. Após isso, o “for” para percorrer a quantidade de elementos da lista da temperatura (.length) e realizando a soma com o comando “soma = soma + temperatura[i]”. Vale ressaltar que o comando [i] após a variável especifica e imprime o primeiro elemento da lista, como visto durante o curso. Em seguida, criou-se a variável da média, utilizando novamente o length para contar a quantidade de itens da lista e realizando a divisão delas (/). Logo, foi criada a variável “maior” que contém uma frase que posteriormente é somada com os meses e as temperaturas correspondentes, tornando o conjunto todo, incluindo a string e os valores, uma só variável (maior). Utilizou-se o if dentro do for para indicar a condição se caso a temperatura for maior que a média. Por fim, foi exibida a saída formatada, neste caso, por meio do “maior”.

2.14 Exercício: Utilizando listas faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são: "Telefonou para a vítima?" "Esteve no local do crime?" "Mora perto da vítima?" "Devia para a vítima?" "Já trabalhou com a vítima?" O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele será classificado como "Inocente".

Resolução e explicação: Primeiramente, criaram-se uma lista vazias para as respostas e outra com as respectivas perguntas. Logo após, utilizou-se o for para percorrer a lista das perguntas, e exibir a primeira dentre elas, especificando-a ([i]). Logo, adicione as respostas do usuário na lista vazia. Em seguida, uma variável valendo 0 foi feita para a soma das respostas 'sim'. Dessa forma, com outro 'for', a lista de respostas foi percorrida e aquelas que foram respondidas com 'sim', foram somadas a variável 'sim'. Por último, as condições foram expostas, classificando como assassino para questões respondidas igual a 5 (==5), cúmplice para maior ou igual a 3 (>=3), suspeito para igual a 2 (==2) e inocente para nenhum dos casos (utilizando o else).

3. CONCLUSÃO

Neste documento, foram abordados e resolvidos inúmeros exercícios práticos em JavaScript, tendo como fito aprofundar o conhecimento e a compreensão da linguagem. Através da análise e resolução dos problemas apresentados, foi possível explorar conceitos fundamentais como manipulação de listas, estruturas de controle, e operadores lógicos, além de praticar e aprimorar o código.

Os exercícios foram estruturados para demonstrar a aplicação prática de técnicas essenciais. Os primeiros exercícios, por exemplo, focaram na remoção de elementos específicos de um array, utilizando os métodos `indexOf` e `splice`. Esse exercício ilustrou a importância do controle de fluxo e das operações de manipulação de tais, além de destacar a necessidade de compreender o funcionamento interno dos métodos para evitar erros comuns.

A resolução desses exercícios não apenas reforçou a sintaxe e os conceitos do JavaScript, mas também ajudou a desenvolver habilidades de resolução de problemas. A prática com exemplos reais permite uma compreensão mais profunda da aplicação das funções e métodos da linguagem, e contribui para a construção de um conhecimento sólido em programação.

Portanto, a realização dos exercícios contribui significativamente para o aprendizado do JavaScript ao proporcionar uma experiência prática que complementa o estudo teórico. A habilidade de identificar e corrigir erros, bem como a capacidade de aplicar conceitos em contextos diversos, são competências fundamentais que foram aprimoradas ao longo deste trabalho.

4. REFERÊNCIAS BIBLIOGRÁFICAS

<https://www.freecodecamp.org/portuguese/news/tres-maneras-de-invertir-una-string-en-javascript/>

<https://www.devmedia.com.br/javascript-length-como-e-quando-utilizar-essa-propriedade/39233>

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/String/indexOf

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Increment>

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical AND assignment](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_AND_assignment)