2.2.1 Exercícios - Lógica de Programação com JavaScript (240 minutos)

Instruções

Siga as instruções abaixo para concluir os exercícios:

- **Grupo**: Individualmente
- **Tempo**: 240 minutos no total
- **Objetivo(s)**:
    - Praticar as habilidades técnicas aprendidas durante a sessão de aula sobre Lógica de Programação com JavaScript.
    - Compartilhar conhecimento sobre funções através de discussão com seus colegas.
- **Recursos**:
    - Seu **computador**
    - **PPT** da sessão
    - Suas **notas** da sessão técnica

Objetivos de Aprendizagem do Participante

Até o final da aula, eu serei capaz de:

- Descrever os fundamentos da lógica de programação usando a linguagem JavaScript:
    - Sintaxe e tipos de valor (números, strings e booleanos)
    - Digitação dinâmica
    - Operadores
    - Variáveis
    - Condicionais
- Criar um programa em JavaScript

Pauta

Parte 1: Aprendizagem Com Base em Projeto em JavaScript (90 minutos)

1. **Baixe** o projeto a seguir, **abra** o console do desenvolvedor e **clique** no botão enviar para ver a mensagem de conexão: https://drive.google.com/file/d/1upaZ4Fs51yY8m_TTSb76IUU7i-6yRXQi/view?usp=sharing

2. **Use** a função a seguir para recuperar a entrada de valor para o e-mail e a senha. **Conecte** o console, a entrada do e-mail e senha no documento formulário:

   ```
   var email = document.getElementById("email").value;
   ```

3. **Confirme** que os campos de e-mail e senha não estão vazios (você pode usar o comprimento da função: https://www.w3schools.com/jsref/jsref_length_string.asp). Se o comprimento da entra da senha ou e-mail for menor que 1, **mostre** a seguinte mensagem de erro usando essas linhas de código:

```
 document.getElementById("alert-email").innerHTML = "Invalid input field. Email should have at least 1 character!";
     document.getElementById("alert-email").style.visibility = "visible";
```

   Se a entrada estiver correta, **oculte** a mensagem de erro usando este código:

```
document.getElementById("alert").style.visibility = "hidden";
```

4. **Confirme** se o e-mail está correto verificando se contém o caractere @. (Você pode usar a função https://www.w3schools.com/jsref/jsref_includes.asp)

5. **Confirme** se o e-mail termina com qualquer um dos seguintes (use a função https://www.w3schools.com/jsref/jsref_endswith.asp):
   - .com
   - .co
   - .org
   - .net
   - .br

Parte 2: Exercícios de Funções JavaScript para mais prática (120 minutos)

Faça os exercícios a seguir. Eles estão em inglês para você praticar a codificação nesse idioma, o que é importante para qualquer desenvolvedor. Se precisar, use um tradutor para fazer os exercícios.

1. **Write** a JavaScript function that reverse a number. Go to the editor

   Example x = 32243;

   Expected Output : 34223

   Click me to see the solution

2. **Write** a JavaScript function that checks whether a passed string is a palindrome or not? Go to the editor
    - A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.

Click me to see the solution

3. **Write** a JavaScript function that generates all combinations of a string. Go to the editor

Example string : 'dog'

Expected Output : d,do,dog,o,og,g

Click me to see the solution

4. **Write** a JavaScript function that returns a passed string with letters in alphabetical order. Go to the editor

Example string : 'webmaster'

Expected Output : 'abeemrstw'

Assume punctuation and numbers symbols are not included in the passed string.

Click me to see the solution

5. **Write** a JavaScript function that accepts a string as a parameter and converts the first letter of each word of the string in upper case. Go to the editor

Example string : 'the quick brown fox'

Expected Output : 'The Quick Brown Fox'

Click me to see the solution

6. **Write** a JavaScript function that accepts a string as a parameter and find the longest word within the string. Go to the editor

Example string : 'Web Development Tutorial'

Expected Output : 'Development'

Click me to see the solution

7. **Write** a JavaScript function that accepts a string as a parameter and counts the number of vowels within the string. Go to the editor
    - Note : As the letter 'y' can be regarded as both a vowel and a consonant, we do not count 'y' as vowel here.

Example string : 'The quick brown fox'

Expected Output : 5

Click me to see the solution

8. **Write** a JavaScript function that accepts a number as a parameter and check the number is prime or not. Go to the editor
    - Note : A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

Click me to see the solution

9. **Write** a JavaScript function which accepts an argument and returns the type. Go to the editor
    - Note : There are six possible values that typeof returns: object, boolean, function, number, string, and undefined.

Click me to see the solution

10. **Write** a JavaScript function which returns the n rows by n columns identity matrix. Go to the editor

Click me to see the solution

11. **Write** a JavaScript function which will take an array of numbers stored and find the second lowest and second greatest numbers, respectively. Go to the editor

Sample array : [1,2,3,4,5]

Expected Output : 2,4

Click me to see the solution.

12. **Write** a JavaScript function which says whether a number is perfect. Go to the editor
    - According to Wikipedia : In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).

Example : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.

Click me to see the solution.

13. **Write** a JavaScript function to compute the factors of a positive integer. Go to the editor

Click me to see the solution.

14. **Write** a JavaScript function to convert an amount to coins. Go to the editor

    Sample function : amountTocoins(46, [25, 10, 5, 2, 1])

    - Here 46 is the amount. and 25, 10, 5, 2, 1 are coins.

    Output : 25, 10, 10, 1

    Click me to see the solution.

15. **Write** a JavaScript function to compute the value of b^n where n is the exponent and b is the bases. Accept b and n from the user and display the result. Go to the editor

    Click me to see the solution.

16. **Write** a JavaScript function to extract unique characters from a string. Go to the editor

    Example string : "thequickbrownfoxjumpsoverthelazydog"

    Expected Output : "thequickbrownfxjmpsvlazydg"

    Click me to see the solution.

17. **Write** a JavaScript function to get the number of occurrences of each letter in specified string. Go to the editor

    Click me to see the solution.

18. **Write** a function for searching JavaScript arrays with a binary search. Go to the editor
    - Note : A binary search searches by splitting an array into smaller and smaller chunks until it finds the desired value.

    Click me to see the solution.

19. **Write** a JavaScript function that returns array elements larger than a number. Go to the editor

    Click me to see the solution.

20. **Write** a JavaScript function that generates a string id (specified length) of random characters. Go to the editor

    Sample character list :
    "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"

    Click me to see the solution.

21. **Write** a JavaScript function to get all possible subset with a fixed length (for example 2) combinations in an array. Go to the editor

Sample array : [1, 2, 3] and subset length is 2

Expected output : [[2, 1], [3, 1], [3, 2], [3, 2, 1]]

Click me to see the solution.

22. **Write** a JavaScript function that accepts two arguments, a string and a letter and the function will count the number of occurrences of the specified letter within the string. Go to the editor

Sample arguments : 'w3resource.com', 'o'

Expected output : 2

Click me to see the solution

23. **Write** a JavaScript function to find the first not repeated character. Go to the editor

Sample arguments : 'abacddbec'

Expected output : 'e'

Click me to see the solution

24. **Write** a JavaScript function to apply Bubble Sort algorithm. Go to the editor
    - Note : According to wikipedia "Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order".

Sample array : [12, 345, 4, 546, 122, 84, 98, 64, 9, 1, 3223, 455, 23, 234, 213]

Expected output : [3223, 546, 455, 345, 234, 213, 122, 98, 84, 64, 23, 12, 9, 4, 1]

Click me to see the solution

25. **Write** a JavaScript function that accepts a list of country names as input and returns the longest country name as output. Go to the editor

Sample function : Longest_Country_Name(["Australia", "Germany", "United States of America"])

Expected output : "United States of America"

Click me to see the solution

26. **Write** a JavaScript function to find the longest substring in a given a string without repeating characters. Go to the editor

Click me to see the solution

27. **Write** a JavaScript function that returns the longest palindrome in a given string. Go to the editor
    - Note: According to Wikipedia "In computer science, the longest palindromic substring or longest symmetric factor problem is the problem of finding a maximum-length contiguous substring of a given string that is also a palindrome. For example, the longest palindromic substring of "bananas" is "anana". The longest palindromic substring is not guaranteed to be unique; for example, in the string "abracadabra", there is no palindromic substring with length greater than three, but there are two palindromic substrings with length three, namely, "aca" and "ada".

    - In some applications it may be necessary to return all maximal palindromic substrings (that is, all substrings that are themselves palindromes and cannot be extended to larger palindromic substrings) rather than returning only one substring or returning the maximum length of a palindromic substring.

    Click me to see the solution

28. **Write** a JavaScript program to pass a 'JavaScript function' as parameter. Go to the editor

    Click me to see the solution

29. **Write** a JavaScript function to get the function name. Go to the editor

    Click me to see the solution

☞⟲ **Reflexão Não Técnica (30 minutos)**

1. **Discuta** com seus colegas o valor de ter funções.

2. **Apresente** seu código implementado para validar o formulário a seus colegas e compare seus resultados.