**Curso de JavaScript**

**1. Executando JavaScript no navegador**

Primeiro, dentro da pasta Explorador de Arquivos, criamos uma pasta que daremos o nome de “projetos” e a abrimos usando o visual studio code. Do mesmo modo como foi apresentado no curso de html e css, dentro da pasta PROJETOS aberta no *vscode*, criamos um arquivo sob o nome de *index.html* e nele geramos a estrutura básica de um documento html, digitando ! e teclando Enter. Vamos começar digitando na área corpo do html as tags **<script>** e **</script>**. Em seguida, dentro da tag script, digitamos a função *console.log*, que imprime um texto no navegador.

Texto

Descrição gerada automaticamente

Figura 1.

Vamos então abrir o arquivo *index.html* no Google Chrome. Abre-se então uma tela em nosso navegador sem nenhum documento escrito. Porém, clicando com o botão direito do mouse sobre qualquer área da tela, abre-se uma pequena janela de opções e ali selecionamos Inspecionar. Acessamos então o DevTools e em um de seus menus, ao lado da aba Elements, existe a aba Console. Ao clicar sobre esta aba, é aberto um espaço na tela onde podemos observar o parâmetro da função *console.log* sendo executado, o que demonstra a conexão do JavaScript com o navegador.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 2.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura 3.

**2. Números em JavaScript**

Vamos abrir a nossa pasta projetos com o Code e logo abaixo do nome PROJETOS no VS Code vamos clicar com o botão direito do mouse e selecionar a opção New Folder e criar uma nova pasta chamada **1. Numbers**. Em seguida, clicamos com o botão direito do mouse sobre o nome da pasta criada e na janela de opções que aparecerá, selecionamos New File. No campo que aparecerá para nomear o arquivo escrevemos index.html, clicamos em Enter e no documento gerado, criamos a estrutura de documento html. Na área head, logo abaixo de title, inserimos a tag script. A figura abaixo exemplifica este processo.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Em seguida, criamos um novo arquivo sob o nome script.js e no documento gerado digitamos a função console.log e inserimos como parâmetro alguns números.

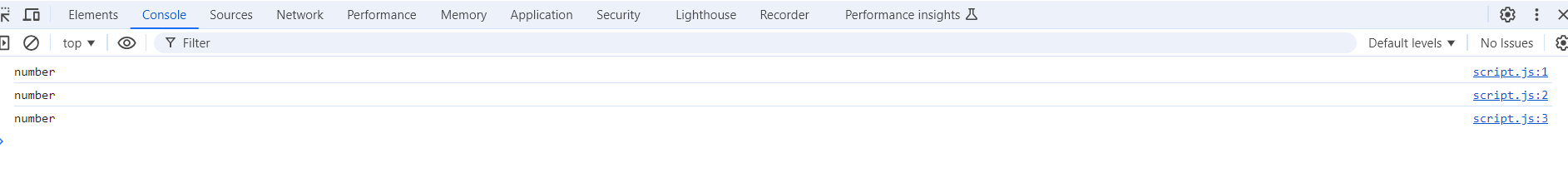
Interface gráfica do usuário

Descrição gerada automaticamente

Abrimos então o arquivo index.html no Google Chrome e clicamos com o botão direito do mouse sobre qualquer área da tela aberta. Na janela de opções que aparecerá em seguida, selecionamos Inspecionar para acessarmos o DevTools. Nele, selecionamos o menu console, que nos apresentará os resultados impressos. A função typeof da maneira como segue na figura abaixo revela no navegador a real natureza desses parâmetros em JS: todos são *numbers*.

Texto

Descrição gerada automaticamente



**3. Operações aritméticas**

Em JavaScript, as operações básicas da aritmética são realizadas com uso dos seguintes operadores.

|  |  |
| --- | --- |
| + | Adição |
| - | Subtração |
| \* | Multiplicação |
| / | Divisão |
| % | Resto |

Em nossa pasta projetos criamos uma nova pasta com o título 2. operações\_aritméticas e nela criamos um arquivo index.html e um arquivo script.js. As figuras abaixo trazem uma representação desta situação.

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**4. Divisibilidade**

Se e são inteiros, dizemos que divide , denotamos , quando existe um inteiro tal que . Se não divide , escrevemos .

Podemos montar um programa em JavaScript que nos mostra se ou não e, em caso afirmativo, imprime na tela o resultado dessa divisão.

Primeiro, criamos o nosso código html.

Texto

Descrição gerada automaticamente

Em nosso html acima estamos utilizando um mecanismo de exibição de texto matemático, o MathJax. Para tanto, inserimos no código html o código seguinte

**<script id="MathJax-script" async**

**src="https://cdn.jsdelivr.net/npm/mathjax@3/es5/tex-mml-chtml.js">**

O código acima é encontrado através do endereço abaixo.

<https://www.mathjax.org/#gettingstarted>

Com este mecanismo, para reproduzir um texto matemático em nossa página html, podemos utilizar a notação em LaTeX e delimitar o texto com o delimitador \(...\), tal como se vê no código acima.

Após criar o nosso código html, criamos então o nosso arquivo script.js para salvar o nosso código em JavaScript. Em JS, declaramos uma variável cujo valor não é constante utilizando a palavra-chave **let**.

<let nome\_da\_variável> = <valor\_atribuído\_a\_variável>

Texto

Descrição gerada automaticamente

Inspecionando o nosso código, tendo como entradas e , obtemos

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

No console do DevTools

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Proposição 1.** *Dados inteiros, se , então .*

É fácil ver que o enunciado da proposição acima é verdadeiro. Já a demonstração da próxima proposição pode não ser tão trivial.

**Proposição 2.** *Dados , se e então .*

**Demonstração:** Como e , tais que e . Multiplicando um inteiro qualquer por ambos os membros da igualdade , obtemos . De modo análogo, obtemos . Somando e , obtemos , donde vem que .

Pelo mesmo motivo da proposição 1, a próxima proposição também não será demonstrada.

**Proposição 3.** *A divisão tem as seguintes propriedades:*

1. *.*

Até o momento estamos considerando o processo de divisão exata, que é este quando ocorre que um inteiro pode ser tomado como o produto de dois fatores . Vamos ampliar o conceito de divisão para incluir os casos em que a divisão não é exata.

**Teorema 1. (Teorema da Divisão)** *Dados dois inteiros e , , existe um único par de inteiros e tais que*

**Demonstração:** Se , temos então que tal que . Logo podemos tomar e . Suponhamos , vamos considerar o conjunto . Temos , pois . No entanto, não podemos ter , pois se assim fosse, teríamos um inteiro maior que todo natural , o que é um absurdo. Logo tal que . Segue-se que, dado , temos , o que implica . De fato, se fosse , , o que contradiz a afirmação de que é o maior elemento de . Concluímos, portanto, que tal que . Segue-se da desigualdade que tal que .

Suponhamos que existem dois pares e tais que , com e . Assim,

Logo temos e , o que é um absurdo. Portanto, o par tal que é único.

O teorema da divisão serve de base para o importante algoritmo da divisão. No livro *Números Inteiros e Criptografia RSA*, de S. C. Coutinho, obtemos a seguinte descrição deste algoritmo.

**Algoritmo da divisão**

**Entrada:** *inteiros positivos e .*

**Saída:** *inteiros não-negativos e tais que e .*

**Etapa 1:** *Comece fazendo e .*

**Etapa 2:** *Se escreva o quociente é e o resto é e pare; senão vá para a Etapa 3.*

**Etapa 3:** *Se subtraia de e incremente de 1 e volte à Etapa 2.*

Podemos montar o código em JS para o algoritmo da divisão. Primeiro criamos a pasta ALGORITMO\_DE\_DIVISAO e nela criamos os arquivos index.html e script.js com os respectivos códigos descritos abaixo.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.