

Boas-vindas!

Agora você faz parte da maior comunidade de aprendizagem online e ao vivo da América Latina!

Como você **chega?**

1



2



3



Esta aula será

- **gravada**

Resumo

da aula anterior

- ✓ Vimos o que é o JavaScript.
- ✓ Conhecemos sua história.
- ✓ Diferenciamos ele do Java.
- ✓ Compreendemos sua relação com HTML e CSS.
- ✓ Introduzimos as ferramentas do curso.

Dúvidas?

Aula 2. JAVASCRIPT

Conceitos Iniciais

Objetivos da aula

- Conhecer os fundamentos da linguagem JavaScript.
- Vincular o código JavaScript ao HTML.
- Compreender o que é uma variável, como fazer sua declaração e atribuir um valor.
- Fazer interações com o usuário por meio do prompt, alert e console.
- Revisão de Git e GitHub

Lógica de programação

Lógica

A lógica está presente na vida de todas as pessoas, todos os dias, praticamente todo o tempo.

Ao usarmos o raciocínio para buscar a melhor solução para um problema ou uma situação, exercitamos a lógica.

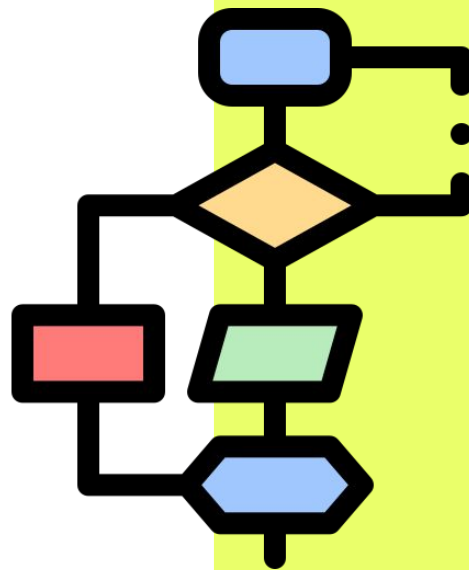
A lógica define o encadeamento de ações mais coerentes para chegar a um objetivo.

Lógica de programação

A lógica de programação é uma forma de organizar pensamentos que permite a tradução do raciocínio lógico humano para a linguagem das máquinas, permitindo que elas realizem determinadas tarefas.

Algoritmo

O algoritmo é a própria sequência de instruções para a execução de uma tarefa. De uma forma mais simples, dentro da lógica de programação ele é uma receita ou roteiro que indica todos os procedimentos necessários para realizar algo ou resolver algum problema.



Exemplo de Algoritmo

Trabalhar pela manhã

1. Acordar
2. Tomar banho
3. Vestir-se
4. Tomar café
5. Tirar o carro da garagem
6. Ir para o trabalho

Chupar uma bala

1. Pegar a bala
2. Retirar o papel
3. Chupar a bala
4. Jogar o papel no lixo

Sintaxe e Código

Código Javascript

O JavaScript tem suas próprias regras de sintaxe, por mais que ainda respeite os padrões de muitas outras linguagens de programação lógicas.

O código JavaScript se associa ao arquivo HTML que é carregado no navegador. **Temos duas maneiras de escrever um código JavaScript em nossos aplicativos web.**



Como escrever o código em JS

Dentro de um arquivo html, entre as tags <script>

Exemplo:

```
<script>  
    // aqui se escreve o código JS  
</script>
```

Como escrever o código em JS

Em um arquivo individual com **extensão .js**

Exemplo: **meu-script.js**

Vincula-se com o rótulo **<script>** e o atributo **"src"**

Lembre-se de não utilizar espaços nos nomes dos arquivos.

Exemplo:

```
<script src="script.js"></script>
```

Regras Básicas

Os espaços em branco e as novas linhas não são considerados (da mesma forma que no HTML).

Case sensitive: se distingue entre letras maiúsculas e minúsculas.

Podem ser incluídos blocos de comentários:

```
<script>  
    // Comentário simples: uma linha  
    /* Comentário de mais de uma linha I  
       Comentário de mais de uma linha II  
    */  
</script>
```


Regras Básicas

São aquelas utilizadas para construir sentenças em JavaScript e que, portanto, não podem ser utilizadas livremente.

Alguns exemplos são:

```
break, case, catch, continue, default, let  
delete, do, else, finally, for, function, if,  
in, instanceof, new, return, switch, this,  
throw, try, typeof, var, void, while, with e  
várias outras.
```



Para praticar

Duração: 05 minutos



ATIVIDADE EM SALA

Para praticar

Descrição da atividade.

- ✓ Crie um arquivo com a extensão .html
- ✓ Insira a tag `<script>` para colocarmos nosso código JavaScript

Tipos, variáveis e valores

Tipos de Dados

Texto: Representam uma sequência de um ou mais caracteres. Normalmente são colocados entre aspas. Ex: nome

Inteiro: Representados por valores numéricos, tanto negativos, quando positivos (sem casas decimais). Ex: idade

Real: Valores numéricos, tanto positivos, quanto negativos que utilizam casas decimais. Ex: altura

Lógico: Representado por alternativas: SIM ou NÃO, VERDADEIRO ou FALSO. Também podem ser chamados de Booleano.

Orientação para professor(a):

A partir desse momento você poderá escolher por onde mostrar os conteúdos abaixo. Você pode abrir o código e mostrar no ambiente de trabalho ou através dos slides. Indicamos que sempre abra o código se houver espaço para isso.

Variáveis

Uma **variável** é um espaço reservado na memória que, como seu nome indica, **pode mudar de conteúdo ao longo da execução de um programa.**

Nas variáveis, armazenamos diversos tipos de dados que utilizamos no aplicativo.

```
<script>  
  var cliente = 'Maria';  
  var quantidade = 30;  
  var preco = 50.5;  
  var emEstoque = true;  
</script>
```

Declaração

Declarar uma variável significa **criá-la**. Para isso, usamos a palavra reservada "var". Escrevemos essa palavra-chave e, em seguida, o nome de nossa variável. Não devemos utilizar espaços nem caracteres especiais.

```
<script>
  // correto
  var nome;
  var telefone;
  var anoNascimento;

  // incorreto
  var o ano
  var preço
  var número
</script>
```


Atribuição

Em uma variável, podemos atribuir diferentes tipos de valores através do operador de atribuição, representado pelo símbolo **=**

```
<script>
  // declaração
  var idade
  var nome
  var sobrenome

  // atribuição
  idade = 5
  nome = "Maria"
  sobrenome = 'Silva'
</script>
```

Inicializar variáveis

Podemos declarar uma variável e atribuir um valor inicial a ela no mesmo processo

```
<script>  
    // variáveis declaradas e  
    inicializadas  
    var idade = 5  
    var nome = "Maria"  
    var sobrenome = 'Silva'  
</script>
```

Operadores Aritméticos

Utilizados para obter resultados numéricos. Os símbolos para os operadores aritméticos são:

- ✓ Soma = +
- ✓ Subtração = -
- ✓ Multiplicação = *
- ✓ Divisão = /
- ✓ Exponenciação = **
- ✓ Módulo (Resto de uma divisão) = %

Operações básicas

Com as variáveis de valores numéricos, é possível realizar operações matemáticas.

```
> resultadoSoma
< 3
> resultadoSubtracao
< 1
> resultadoProduto
< 6
> resultadoDivisao
< 1.5
```

```
<script>
  var numeroA = 1;
  var numeroB = 2;
  var numeroC = 3;

  //Soma de dois números
  var resultadoSoma = numeroA + numeroB;

  //Subtração de dois números
  var resultadoSubtracao = numeroB - numeroA;

  //Produto de dois números
  var resultadoProduto = numeroB * numeroC;

  //Divisão de dois números
  var resultadoDivisao = numeroC / numeroB;
</script>
```

Operações básicas

Com as variáveis do tipo string (texto), podemos concatenar os valores. Ou seja, combiná-las.

```
> resultadoA
< 'CODERHOUSE'
> resultadoB
< 'HOUSE1'
> resultadoC
< 'CODER HOUSE'
```

```
<script>
  var textoA = "CODER";
  var textoB = "HOUSE";
  var branco = " ";

  //Concatenar textoA e texto B
  var resultadoA = textoA + textoB;

  //Concatenar textoB e 1
  var resultadoB = textoB + 1;

  //Concatenar textoA, branco e textoB
  var resultadoC = textoA + branco +
textoB;
</script>
```



Para praticar

Duração: 10 minutos



ATIVIDADE EM SALA

Para praticar

Usando o arquivo que criamos anteriormente, declare e atribua as seguintes variáveis:

- ✓ Seu nome
- ✓ Seu sobrenome
- ✓ Ano atual
- ✓ Idade que você completa até o final deste ano

Usando as operações básicas, crie variáveis com as seguintes informações:

- ✓ Seu nome e sobrenome concatenados
- ✓ O ano que você nasceu (subtraia o ano atual pela sua idade no final deste ano)



ATIVIDADE EM SALA

Para praticar!



Orientação para a equipe de aula: Se sentir necessidade, faça primeiro o seu como exemplo e peça para os alunos replicarem depois com o deles.



Break

5 minutos e voltamos!





Break

10 minutos e voltamos!



Prompt, console e alert

Prompt

O **prompt()** mostrará ao usuário uma caixa de diálogo para que insira um dado.

Podemos definir uma mensagem que aparecerá para o usuário.

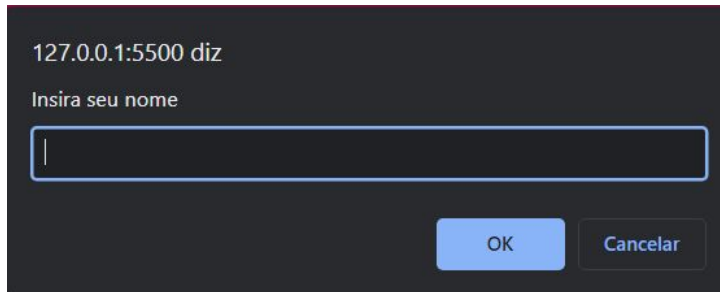
O valor que retorna é uma string que representa o que o usuário inseriu.

```
<script>  
    var nomeInserido = prompt("Insira seu  
nome");  
</script>
```

Prompt

Na tela do navegador, o usuário verá sobre a página uma janela que solicitará um dado.

O valor que o usuário insere é conhecido como a **entrada**.



A screenshot of a web browser's prompt dialog box. The dialog has a dark gray background. At the top, it displays the text "127.0.0.1:5500 diz" in a light gray font. Below this, the prompt text "Insira seu nome" is shown in a slightly larger, light gray font. Underneath the text is a white rectangular input field with a thin blue border and a vertical cursor on the left. At the bottom of the dialog, there are two buttons: a blue button with the text "OK" and a gray button with the text "Cancelar".

Console

O `console.log()` mostra a mensagem que passamos como parâmetro à chamada no console JavaScript do Navegador.

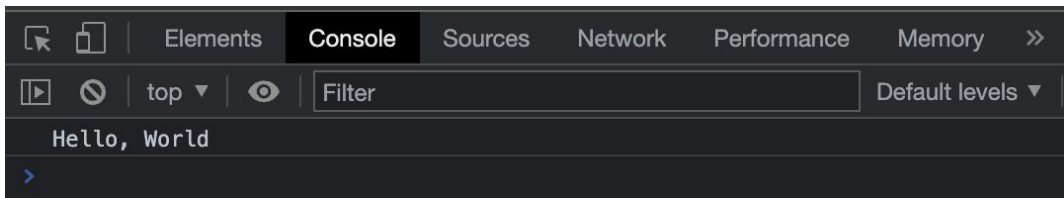
```
<script>
  console.log("Mensagem de teste");

  var saudacao = "Hello, World"
  console.log(saudacao)
</script>
```

Console

No Chrome, o console do navegador pode ser acessado da seguinte forma:

Clique com o botão direito do mouse sobre alguma parte da página (ou no botão F12 do teclado) > Inspecionar > Console.



Alert

O **alert()** mostrará uma janela sobre a página que estivermos acessando, exibindo a mensagem que for passada como parâmetro à chamada.

```
<script>  
  alert("Olá, Coder!");  
</script>
```


Alert

Na tela do navegador, o usuário verá uma janela sobre a página que mostrará uma mensagem.

O valor que mostramos ao usuário como um resultado é conhecido como a saída.



Exemplo de script completo

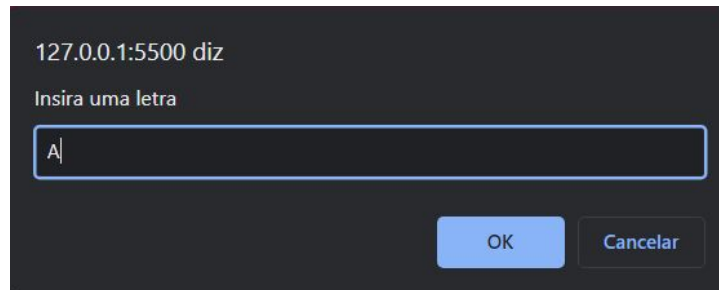
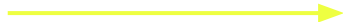
Este é um exemplo de um Script JS rodando em um arquivo HTML.

```
<!DOCTYPE html>
<html>
<head>
  <title>Meu primeiro App</title>
  <script>
    var entrada = prompt("Insira uma
letra");
    var saida = entrada + " " +
"inserida";
    alert(saida);
  </script>
</head>

<body>
  <h2>Esta página contém um app</h2>
</body>
</html>
```

Entrada/saída

Se insiro "A"...

A dark-themed terminal dialog box. At the top, it says '127.0.0.1:5500 diz'. Below that, it says 'Insira uma letra'. There is a text input field containing the letter 'A' with a cursor at the end. At the bottom right, there are two buttons: 'OK' and 'Cancelar'.

Obtenho...

A dark-themed terminal dialog box. At the top, it says '127.0.0.1:5500 diz'. Below that, it says 'A inserida'. At the bottom right, there is a single button labeled 'OK'.

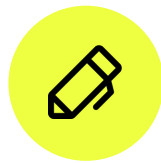
Parse

Usamos o `prompt()` para solicitar dados ao usuário e `console.log()` ou `alert()` para mostrar o resultado das operações realizadas com esses dados.

Os dados que recebemos são sempre considerados string.

Portanto, para trabalhar com números é preciso fazer uma conversão usando `parseInt()` ou `parseFloat()`.

```
<script>
  var valorString = prompt("Insira um número");
  var valorNumerico = parseInt(valorString);
</script>
```



Para Praticar

Duração: 10 minutos



ATIVIDADE EM SALA

Para praticar

No mesmo arquivo que criamos, siga as seguintes instruções:

- ✓ Utilizando o prompt, receba 2 números do usuário
- ✓ Realize a soma desses 2 números e exiba o resultado no console
- ✓ Realize o produto desses 2 números e exiba o resultado no alert

Revisão de Git e Github

O que é git?

Git é um sistema de **controle de versões** gratuito e de código aberto, desenvolvido para manipular de pequenos a grandes projetos de maneira rápida e eficaz.

Para utilizá-lo é preciso fazer a instalação no nosso computador.

Entende-se como controle de versões todas as ferramentas que nos permitem **fazer modificações** no nosso projeto e **registrar as alterações realizadas** em um arquivo ou conjunto de arquivos ao longo do tempo.

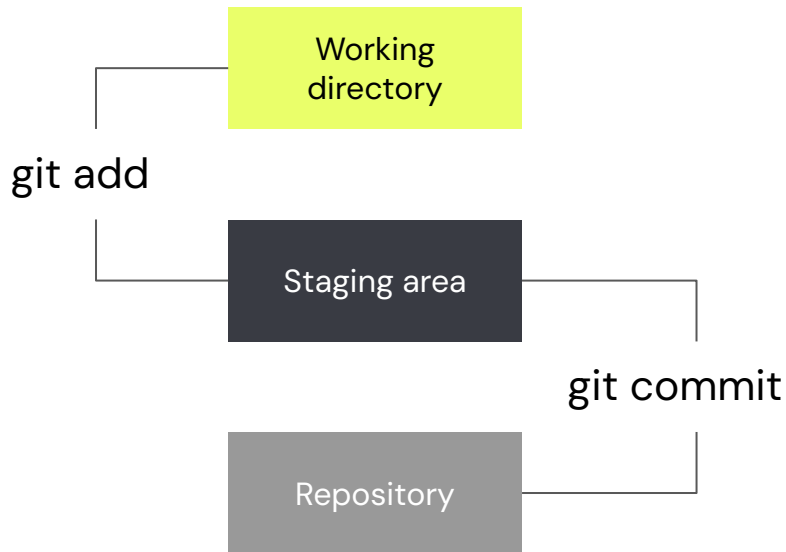
Fonte: <https://git-scm.com/>



CODERHOUSE

Git: 3 estágios

- 1º **estágio** (no qual trabalhamos)
"preparamos as caixas".
- 2º **estágio** (arquivos prontos)
"acrescentamos as caixas prontas".
- 3º **estágio** (registro de todos os arquivos)
"lote pronto".



O que é um repositório?

Um repositório é um **espaço centralizado** onde se armazena, organiza, mantém e divulga informação.

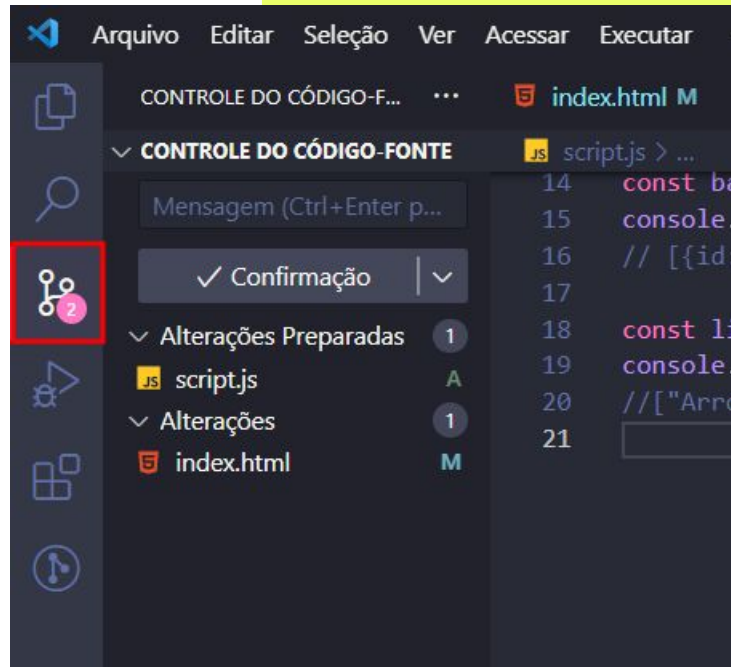
Será “a pasta” ou espaço na qual você **guardará o seu projeto**, para depois compartilhá-lo com a equipe por meio de um repositório na nuvem, por exemplo, no Github.

Comandos

- ✓ **git init:** indicar que nesse diretório, onde executamos o comando, será criado um repositório para ser usado com o GIT.
- ✓ **git status:** para verificar o status do nosso repositório.
- ✓ **git add .:** adicionar todos os arquivos criados, modificados ou eliminados ao estágio 2 (Staging area).
- ✓ **git commit -m "Mensagem":** envio das alterações para o repositório, no estágio 3.
- ✓ **git log --oneline:** para visualizar os commits realizados.

GIT no VS Code

- ✓ Abra seu repositório no VS Code
- ✓ No canto superior esquerdo, temos um ícone que nos permite trabalhar com o Git de uma forma visual dentro do VS Code
- ✓ Ao clicar sobre ele vemos todas as alterações que foram feitas no projeto e estão no 1º estágio (Alterações)
- ✓ Clicando no ícone + em cada arquivo, enviamos eles para o 2º estágio (Alterações Preparadas)
- ✓ Para fazer o commit, basta escrever a mensagem no campo superior e apertar o botão "Confirmação". Assim, todos os arquivos que estiverem em "Alterações Preparadas" serão enviados para o Repositório, no 3º estágio



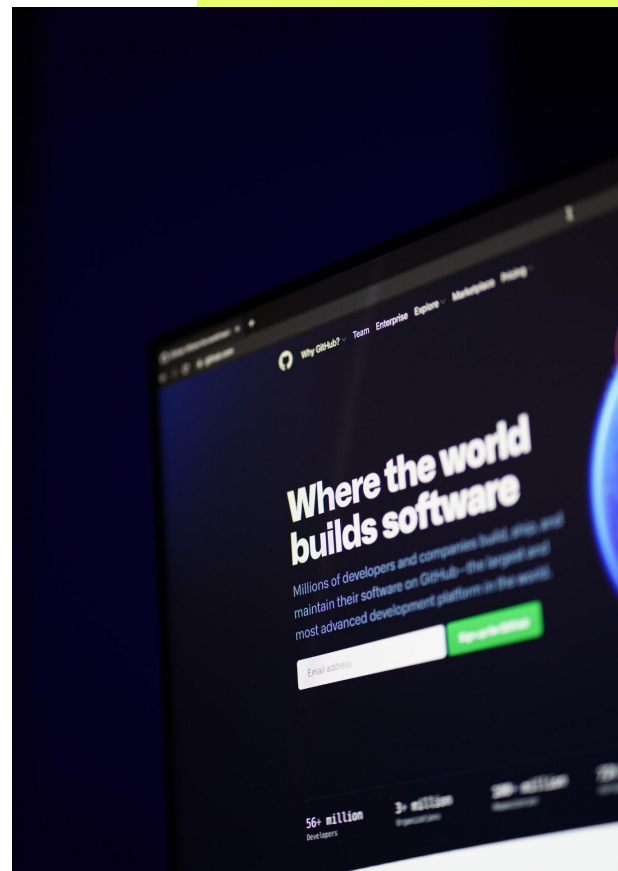
O que é Github?

Por enquanto, tudo o que vinha acontecendo no Git era de maneira local, e não precisávamos da internet para salvar os nossos *commits* e o nosso repositório.

Agora queremos compartilhar o nosso trabalho com outras pessoas (parceiros de projeto, clientes, etc.). Para isso, utilizamos o Github!

O Github é uma espécie de **“rede social” de programadores**. Com esta plataforma, podemos subir os nossos projetos e deixar que outras pessoas colaborem.

<https://github.com>

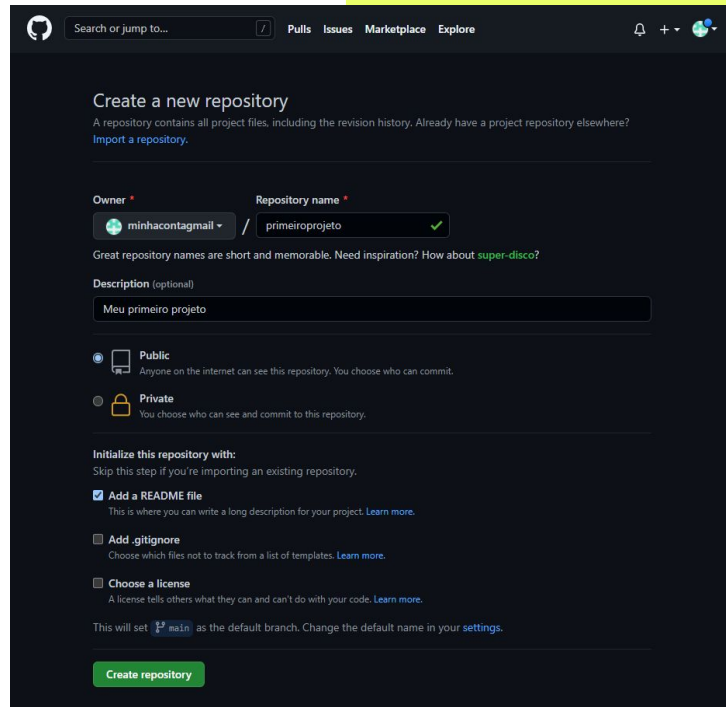


Criando um repositório

Por exemplo, ele poderia se chamar **"primeiroprojeto"**, para que você teste seu repositório com os arquivos que criamos na aula.

Escolhemos entre **"público"** ou **"privado"** e depois, clicamos em **"Create repository"**.

OBS: Use público para que possamos compartilhar e publicar.



The screenshot shows the GitHub interface for creating a new repository. At the top, there's a search bar and navigation links for Pulls, Issues, Marketplace, and Explore. The main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'

The form includes two input fields: 'Owner' with a dropdown menu showing 'minhacontagmail' and 'Repository name' with a text input containing 'primeiroprojeto' and a green checkmark. Below these, a message says: 'Great repository names are short and memorable. Need inspiration? How about [super-disco](#)?'

The 'Description (optional)' section has a text input field containing 'Meu primeiro projeto'.

There are two radio button options for visibility: 'Public' (selected) with the description 'Anyone on the internet can see this repository. You choose who can commit.' and 'Private' with the description 'You choose who can see and commit to this repository.'

The 'Initialize this repository with:' section has a note: 'Skip this step if you're importing an existing repository.' It includes three checkboxes: 'Add a README file' (checked), 'Add .gitignore', and 'Choose a license'. Each checkbox has a link to 'Learn more'.

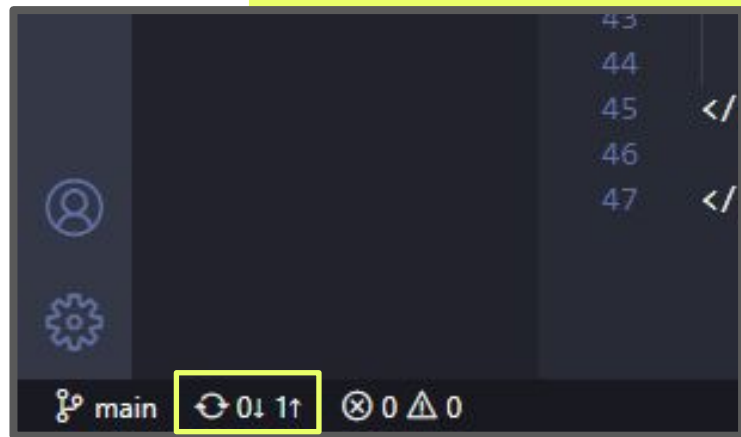
At the bottom, it says 'This will set [main](#) as the default branch. Change the default name in your [settings](#).' and a green 'Create repository' button.

Comandos para o GitHub

- ✓ **git push:** para enviar os commits para o GitHub
- ✓ **git pull:** para buscar os commits que já foram feitos
- ✓ **git clone:** para baixar um repositório do GitHub para o seu computador
- ✓ **git remote add origin:** para vincular seu projeto local com um repositório do GitHub

GitHub no VS Code

- ✓ Abra seu repositório no VS Code
- ✓ No canto inferior esquerdo, vemos a indicação de commits para serem enviados (push) ou baixados (pull) do GitHub
- ✓ Ao clicar sobre eles o pull e o push são feitos automaticamente, caso não haja nenhum conflito



GitHub Pages

O GitHub lhe permite **publicar seus projetos na internet**. Para gerar uma GitHub page, você deve:

1

Acesse seu repositório no GitHub e vá em "Settings".

2

Clique em "Pages".

3

Selecione a branch que será publicada (master ou main).

4

Salve as alterações. O GitHub irá publicar sua página e disponibilizar o link de acesso.

Perguntas?

Como foi a aula?

1

Que bom

O que foi super legal na aula e podemos sempre trazer para as próximas?

2

Que pena

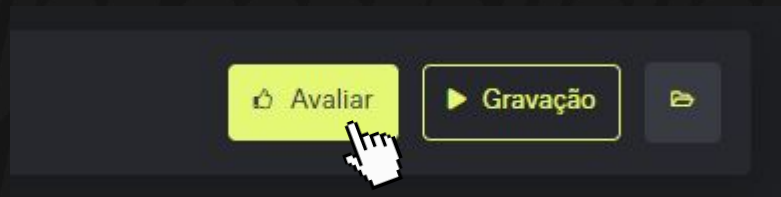
O que você acha que não funcionou bem e precisamos melhorar?

3

Que tal

Qual sugestão deveríamos tentar em próximas aulas?

O que você achou da aula?



Seu feedback vale pontos para o Top 10!! 😎



Deixe sua opinião!

1. Acesse a plataforma
2. Vá na aula do dia
3. Clique em **Avaliar**



Criar um algoritmo JS simples

DESAFIO COMPLEMENTAR



DESAFIO COMPLEMENTAR

Criar um algoritmo JS simples

Descrição

- ✓ Crie um script em JS que solicite ao usuário que insira no mínimo um dado do tipo texto e outro do tipo numérico.
- ✓ Em seguida, com JavaScript, realize operações matemáticas ou de concatenação sobre as entradas, considerando o tipo de dado.
- ✓ Ao finalizar, mostre os resultados por meio de `alert()` ou `console.log()`.

Formato

- ✓ Link do código no GitHub e da publicação no GitHubPages.

Exemplos

- ✓ Solicitar o nome pelo prompt e mostrá-lo em console junto a algum texto de saudação. Exemplo: Olá, João!
- ✓ Solicitar um número pelo prompt, parseá-lo, somá-lo a outro que esteja armazenado em uma variável e, por fim, mostrar o resultado por console.
- ✓ Solicitar um texto pelo prompt, depois outro, concatená-los e mostrar o resultado através de alert.

Resumo

da aula de hoje

- ✓ Fundamentos do desenvolvimento com JS.
- ✓ Como vincular o código JavaScript ao HTML.
- ✓ Declaração e atribuição de variáveis.
- ✓ Funções de prompt, alert e console.
- ✓ Revisão de Git e GitHub.



**Obrigado por estudar
conosco!**