Trilha Fundamentar

* JavaScript
* Guia Estelar de JavaScript  
    
   --JavaScript Core - > Linguagem de programação que roda no navegador do usuário (Front-End).  
  - Se você clicar em algum botão da página e aparece uma janela, isso é JavaScript.   
  - Alteração do site ou aplicativo, conforme a interação do usuário.   
  - Roda também no computador (Back End)  
  - O JavaScript é a Inteligência da Tríade. O HTML é a estrutura, O CSS é a beleza.  
  - JavaScript é diferente de JAVA. Apesar do nome semelhante, são linguagens diferentes.   
  - Ele é usado em aplicativos, para WEB, Desktop (Electron) e mobile (React Native)  
  -   
    
   --O que podemos fazer com JS?   
    
  - Podemos criar aplicações Web, mobile (React Native), desktop (Electron)  
  - 99.99% dos sites na web usam JavaScript. É uma linguagem obrigatória para quem programa o front-end web.
* A importância da Sintaxe -> Toda linguagem de programação tem.  
  -Uma boa comunicação necessita de uma boa sintaxe.  
  - 82% dos erros de iniciantes na programação está na sintaxe.
* Maneiras de executar o JavaScript -> No navegador, através do Console.  
  - Através do Codepen Io.  
  - Através do Vscode.
* Comentários -> É possível escrever comentários dentro do meu código que vão ser ignorados no processamento dele.   
  Se eu começo uma linha com. // , eu estou dizendo que aquela linha que eu estou escrevendo é um comentário e deve ser desconsiderada. Porém, esse é uma forma de fazer o comentário em uma linha. Caso eu queria fazer um bloco de comentário, eu devo começar a linha com /\* e fechar com \*/
* Tipos de dados -  
  - Gramática -> A gramática são os elementos da linguagem e suas combinações. É a arte de falar e escrever corretamente.   
  - Vocabulário -> É o conjunto de termos e expressões de uma certa linguagem. Ou um agrupamento de palavras.   
  - Precisamos saber como escrever no JavaScript.   
  - Precisamos saber o significado das palavras no JavaScript.  
  - Precisamos continuar aprendendo, para crescer o nosso vocabulário.
* String – String é uma cadeia de caractéres. E isso formam textos dentro do JavaScript. E se eu quero escrever textos no JavaScritp eu preciso usar um dos seguintes elementos:   
    
  “” – Aspas duplas   
  ‘ ‘ – Aspas simples   
  `` - Crases duplas (Template literals ou template Strings)
* Number – São outro tipos de dados. O tipo number podem ser inteiros, que podem ser positivos ou negativos.  
  Podem ser números reais ou float, que são os números quebrados, com casas decimais.   
  Podem ser considerados a not number, que quer dizer que não é um número.   
  E podem ser infinity, que corresponde a infinito (Essa opção quase não é utilizável porém existe).
* Boolean – É um tipo de dados que só possui dois valores. Ou ele é verdadeiro ou ele é falso. A maior parte das lógicas de programação vão estar dentro desse valor “True” ou “ false”.
* Undefined vs null - São dois tipos de dados muito confundidos que porém em JavaScript são diferentes.   
  O Undefined é um item que é indefinido. Que não existe.   
  Já o Null, é um item que é Nulo, um objeto que não possui nada dentro, diferente de indefinido.
* Object – Objeto, é um tipo de dado estrutural, que cria uma estrutura. Tudo é objeto.   
  Todo objeto vai possuir propriedades ou atributos e Funcionalidades ou métodos.  
  Os atributos são as propriedades de um objeto.   
  Já os metodos são as funcionalidades de um objeto.
* Array (Vetores) – também é um tipo de dado estrutural. É uma lista, um agrupamento de dados.
  + Variáveis -> Variáveis são nomes simbólicos para receber algum tipo de valor. São como caixinhas onde guardamos um tipo de dado para usar mais tarde.   
     - São atalhos de código.   
     - São identificadores.   
    -Podemos usar 3 palavras reservadas para criar uma variável:   
     -var = Var é uma variável Global e também Local.   
     -let =Let é uma variável local. E só funciona no escopo onde foram criadas  
     -Const = A contante não pode mudar o seu valor durante a aplicação. Ela é constante, ela não muda. E só funciona dentro do escopo onde foi criada.   
      
    EX:   
      
    Var clima = “quente”   
     (Estou declarando uma variavel, depois do igual, estou atribuindo um valor a essa variavel)
  + Scope e var -> Escopo determina a visibilidade de alguma variável no JavaScript. Pois nem sempre as variáveis vão estár visívies para o código de JavaScript.   
      
    -Block Statement - É uma declaração de bloco. Ele começa e termina com {}.  
    {  
    // Aqui dentro é um bloco e posso colocar qualquer código   
    } //Aqui fecha o código.   
      
    A criação desse bloco de declaração, também criará um novo escopo. Chamamos de ‘Block -Scoped’  
      
    -Hoisting -> É uma ação invisível do JavaScript que eleva a variável criada para o início do código. E depois vai até a linha onde está definida a variável e define o seu valor.
  + Criando nomes para as variáveis -   
    O JavaScript é case-sensitive, ou seja, ele percebe a diferença entre letras maíusculas e letras minúsculas.   
    Além aceitar a cadeia de caractéres Unicode. Ou seja, ele aceita aspas, acentos no nome das minhas variáveis.   
      
    -Posso:   
     🡪 Iniciar com caracteres especiais ($ &)  
     🡪 Iniciar com letras   
     🡪 Colocar Acentos   
     🡪 Letras maíusculas e minúsuclas fazem diferença.   
      
    - Não Posso:   
      
     🡪 Iniciar com números   
     🡪 Colocar espaços vazios no nome (ex: Let oi aqui é)  
      
    -Ideal:   
      
     🡪Criar nomes que fazem sentido e que explique o que a variável é ou faz.   
     🡪CamelCase – Que é ao invés de colocar um espaço entre as palavras, eu uso a palavra com letra maíuscula.   
     🡪Escreva o nome da variável e o seu código em Ingles. Para que o seu código possa ser universal.
  + Funções - Funções são:   
      
    🡪Servem para agrupamento de código  
      
    🡪Reuso de código   
      
    🡪Ela é como se fosse um mini programa dentro do programa maior (código todo)  
      
    🡪Toda linguagem oferece muitas opções de funções.   
    Os dados dentro de uma função se chamam argumento.