**Aula 01**

**Variáveis com Var.**

a declaração VAR tem escopo de função e sofre hoisting, significando que a variável pode ser cassada antes da sua declaração no código, no entanto, isso pode levar a erros de programação devido ao comportamento imprevisível do Hoting.

**LET E CONST.**

LET permite declarar variaveis com escopo de bloco, reduzindo o risco de erros, reduzindo o risco de erros por uso indevido em diferentes partes do código.

CONST é similar, mas cria variáveis imutáveis, ideal para valores que não devem ser alterados após a atribição.

**HOISTING DE VARIÁVEIS**

Ocorre quando declarações de variáveis são movidas para o topo do seu escopo antes da execução do código.

VAR é afetada pela HOISTIN enquanto a LET e CONST não, tornando-as mais previsíveis e seguras.

**EX:**

//uso de var

var carro = 2;

//uso do let

let = carro = 3;

//uso da const

const = 4;

**aula 02**

Node js é uma plataforma de execução para javascript no servidor baseado no motor v8 do javascript, ele expande o uso do javascript, tradicionalmente uma linguagem de front-end para desenvolvedores back-end, oferecendo alta performance e escalabilidade.

**Ambiente de execução**

No node js, o javascript é executado no lado do servidor ,permitindo a construção de aplicações web, apis e processamento de dados em BACK-END.

diferente do navegador, tem acesso a apis de sistema operacional, permitindo interações a nível de sistema de arquivos e rede.

**Event loop e non-blocking 1/0**

O event loop é o coração do node js permitindo operações assíncronas e não bloqueantes. isso significa que o nodejs pode lidar com múltiplas operações, como 1/0 de rede ou acesso ao banco de dados, sem a necessidade de múltiplas threads, otimizando o uso de recursos

**Back-end frameworks e ferramentas.**

frameworks como EXPRESS JS simplificam a criação de aplicações APIS RESFUL em NODEJS, essas ferramentas fornecem abstração e funcionalidade para loteamento, manipulação de requisição/ resposta e muito mais, acelerando o desenvolvimento.

**Operadores**

Operadores são simbolos especiais ou palavras-cave que realizam operações em um, dois ou até três valores.

| Operadores | Significado |
| --- | --- |
| == | igual |
| != | diferente |
| < | menor |
| > | maior |
| <= | menor que ou igual a |
| >= | maior que ou igual a |

Eles são fundamentais para a construção de lógica e algoritmos em programação, permitindo que desenvolvedores executam tarefas como cálculos matemáticos, comparações entre valores ou condições.

ok