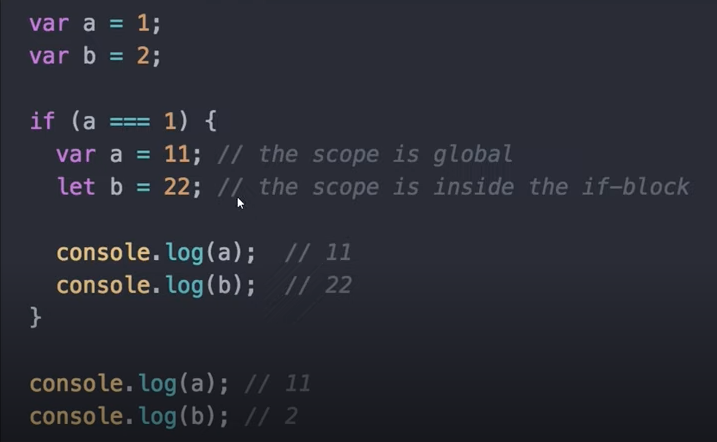
**Variáveis**

**Boas práticas:**

* **Original Variable as String** – some awesome var
* **Camel Case** – someAwesomeVar
* **Snake Case** – some\_awesome\_var
* **Kebab Case** – some-awesome-var
* **Pascal Case** - SomeAwesomeVar
* **Upper Case Snake Case** – SOME\_AWESOME\_VAR //muito utilizado para constantes

**Var e let**

**let** – ela introduz o conceito de escopo de bloco. **SÓ** **vai funcionar dentro do bloco em que ele está inserido**

****

* possuem escopos diferentes
* **hoisting** – atribuição de um valor à variável antes de declarar ela. **let não faz hoisting**
* **redeclaração** –
* **reatribuição** - (só acontece com o var)
* **camelCase** – someAwesomeVar
* let numberOne; //quando for utilizar o let, É NECESSÁRIO DECLARAR ELE ANTES
* numberOne = 1; //é uma atribuição e não uma declaração
* console.log(numberOne + 2);

var primeiroNome = "João"; //escopo global

let ultimoNome = "Silva"; //escopo global

if (primeiroNome === "João") { //=== é para ver se é igual e do mesmo tipo

    var primeiroNome = "Pedro";

    let ultimoNome = "Souza";//escopo de bloco

}

console.log (primeiroNome, ultimoNome);

**Resultado:** Pedro Silva, pois o console.log está fora do bloco em que está o let

Como o if é igual a João, a variável é redeclarada

var primeiroNome = "João"; //escopo global

let ultimoNome = "Silva"; //escopo global

if (primeiroNome === "João") { //=== é para ver se é igual e do mesmo tipo

    var primeiroNome = "Pedro";

    let ultimoNome = "Souza";//escopo de bloco

    console.log (primeiroNome, ultimoNome);

}

Resultado: Pedro Souza, pois o console.log está dentro do bloco do let (esse let está dentro de um escopo de bloco)

**Constantes**

* são declaradas em SNAKE\_UPPER\_CASE
* escopo de bloco
* não faz hoisting

const PRIMEIRO\_NOME = "Renan";

PRIMEIRO\_NOME = "Toporoff"; //NÃO é possível alterar ou redeclarar o valor de uma constante

