

```
//LET:  
//No Javascript podemos declarar variáveis de 2 formas, usando as palavras reservadas "var" e "let". Cada uma delas têm características especiais quanto a sua aplicação. O let por exemplo, são variáveis que sempre obedecerão ao escopo mais próximo do qual ela foi declarada, isso significa que, se temos 2 variáveis let com o mesmo nome e fazemos uma chamada delas, essa chamada vai referenciar a declaração mais próxima dela. O que não acontece com as variáveis var, que podem ser referenciadas de qualquer lugar do código.  
//Além disso, variáveis let só podem ser sobrescritas nunca poderão ser redeclaradas, o que não acontece com as variáveis "var" que podem ser tanto sobrescritas quanto redeclaradas.
```

```
//LET NÃO PODE SER REDECLARADO:
```

```
var a = 2; //Perceba que redeclaramos a variável a logo abaixo...  
let b = 2; //em seguida tentamos fazer a mesma coisa com a variável b...  
var a = 3;  
let b = 3; //A segunda tentativa de redeclará-la dará erro  
console.log(a, b);  
//Veja no console que o erro é apresentado só na variável b que usa let, pois as variáveis que usam var podem ser redeclaradas.
```

```
//LET VAI OBEDECER SEMPRE AO ESCOPO MAIS PRÓXIMO:
```

```
//Perceba que no uso de let, ela vai referenciar sempre ao contexto mais próximo quando for chamada, caso uma variável não esteja no seu contexto, ela vai abrir uma busca para fora do contexto, mas nunca vai invadir um contexto que está para dentro, só os contextos de fora...  
let numero = 1; //Global  
{  
  let numero2 = 2; //Dentro do 1º Bloco  
  {  
    let numero = 3; //Dentro do escopo do 2º Bloco...  
    console.log('dentro = ' + numero); //Essa chamada encontra primeiro a que está dentro do escopo do 2º bloco...  
    console.log('dentro = ' + numero2); //Essa chamada procura no seu escopo e não acha, então ela sai a procura no escopo de fora até encontrar uma variável com o nome "numero2"...  
  }  
}  
console.log('fora = ' + numero); //Essa variável procura somente no contexto global, ela não entra em blocos para procurar por variáveis com o mesmo nome dentro deles...
```

```
console.log('fora = ' + numero2); //Essa tenta encontrar o nome da variável no contexto global, por não encontrar, ela finaliza com erro, apesar de ter uma variável com o nome "número2" dentro dos blocos...
```

```
//COMO UMA VARIÁVEL COM LET NÃO É INFLUENCIADA PELO ESCOPO DE GLOBAL:
```

```
let i = 'Eu não estou dentro.';
for (let i = 0; i < 10; i++) { //Perceba, que a variável let que está dentro do laço for não pode ser referenciada pela chamada de fora do bloco do laço ...
    console.log(i);
}
console.log(i); //Veja que a variável que é chamada é a de fora do laço...
```

```
//ESCOPO DE FUNÇÃO DO LET:
```

//Outra coisa a ser levada em consideração quando se trata de escopo do let, é que o let também tem escopo de função, por isso ele vai agir segundo as regras do closure, ou seja, se uma variável let for declarada dentro do tempo de execução de uma função, os valores declarados sobre ela serão respeitados, o que não aconteceria com um var:

```
const juncs = []
for(let j = 0; j < 10; j++){
    juncs.push(function () { //usando a função push vamos incrementar valores dentro do array juncs dentro do escopo de função...
        console.log(j);
    })
}
```

```
juncs[2](); //Perceba que, quando chamamos a função juncs, os valores serão respeitados...
juncs[8]();
```

```
//HOSTING NÃO ACONTECE EM LET:
```

```
console.log('5) ', c);
let c = 'Eu não foi possível'; //Como let tem um comportamento de escopo diferente de var, ele não sofre hosting
console.log('6) ', c);
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js"
3 2
dentro = 3
dentro = 2
fora = 1
0
1
2
3
4
5
6
7
8
9
Eu não estou dentro.
2
8
6) Eu não foi possível

[Done] exited with code=0 in 0.191 seconds
```

...

RESULTADO DE LET NÃO SENDO REDECLARADO...

```
[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js"
c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js:11
let b = 3; //A segunda tentativa de redeclará-la dará erro
  ^

SyntaxError: Identifier 'b' has already been declared
    at wrapSafe (internal/modules/cjs/loader.js:988:16)
    at Module._compile (internal/modules/cjs/loader.js:1036:27)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1101:10)
    at Module.load (internal/modules/cjs/loader.js:937:32)
    at Function.Module._load (internal/modules/cjs/loader.js:778:12)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:76:12)
    at internal/main/run_main_module.js:17:47

[Done] exited with code=1 in 0.137 seconds
```

....

RESULTADO DE LET NÃO ENTRANDO NO ESCOPO INTERNO DE UM BLOCO...

```
[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js"
3 2
dentro = 3
dentro = 2
fora = 1
c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js:29
console.log('fora = ' + numero2); //Essa tenta encontrar o nome da variável no contexto global, por não encontrar, ela finaliza com
erro, apesar de ter uma variável com o nome "número2" dentro dos blocos...
    |   |   |   |   |   |   ^
    |   |   |   |   |   |   |
ReferenceError: numero2 is not defined
    at Object.<anonymous> (c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js:29:25)
    at Module._compile (internal/modules/cjs/loader.js:1072:14)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1101:10)
    at Module.load (internal/modules/cjs/loader.js:937:32)
    at Function.Module._load (internal/modules/cjs/loader.js:778:12)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:76:12)
    at internal/main/run_main_module.js:17:47

[Done] exited with code=1 in 0.167 seconds
```

...

RESULTADO DE LET DANDO ERRO DE HOSTING...

```
3 2
dentro = 3
dentro = 2
fora = 1
0
1
2
3
4
5
6
7
8
9
Eu não estou dentro.
2
8
c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js:57
console.log('5) ', c);
                ^

ReferenceError: Cannot access 'c' before initialization
    at Object.<anonymous> (c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\013-manipulandoLet.js:57:20)
    at Module._compile (internal/modules/cjs/loader.js:1072:14)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1101:10)
    at Module.load (internal/modules/cjs/loader.js:937:32)
    at Function.Module._load (internal/modules/cjs/loader.js:778:12)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:76:12)
    at internal/main/run_main_module.js:17:47
```