# Levantamento de JavaScript

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_debugging.asp&usg=ALkJrhjtWTqsZkqzXfzCzRFVGT6sTloInQ)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_strict.asp&usg=ALkJrhjXghOhXawZGq1egZkwH53oTnfVyw)

Hoisting é o comportamento padrão do JavaScript de declarações móveis para o topo.

## Declarações de JavaScript são elevadas

Em JavaScript, uma variável pode ser declarada depois de ter sido usada.

Em outras palavras; uma variável pode ser usada antes de ter sido declarada.

**O exemplo 1** dá o mesmo resultado que o **Exemplo 2** :

### Exemplo 1

x = 5; // Assign 5 to x  
  
elem = document.getElementById("demo"); // Find an element   
elem.innerHTML = x;                     // Display x in the element  
  
var x; // Declare x

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_hoisting1&usg=ALkJrhgbECBymlmkY7Vi8SIAotvH84rqgg)

### Exemplo 2

var x; // Declare x  
x = 5; // Assign 5 to x  
  
elem = document.getElementById("demo"); // Find an element   
elem.innerHTML = x;                     // Display x in the element

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_hoisting2&usg=ALkJrhg0_izVqI5yNEIvv3EC5NPJDJBwoQ)

Para entender isso, você precisa entender o termo "hoisting".

Hoisting é o comportamento padrão do JavaScript de mover todas as declarações para o topo do escopo atual (para o topo do script atual ou da função atual).

## As iniciais do JavaScript não são forçadas

O JavaScript apenas exibe declarações, não inicializações.

**O exemplo 1** **não** dá o mesmo resultado que o **Exemplo 2** :

### Exemplo 1

var x = 5; // Initialize x  
var y = 7; // Initialize y  
  
elem = document.getElementById("demo"); // Find an element   
elem.innerHTML = x + " " + y;           // Display x and y

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_hoisting4&usg=ALkJrhjLhEN5yNVrXJDWLAktPzYuSWRVqQ)

### Exemplo 2

var x = 5; // Initialize x  
  
elem = document.getElementById("demo"); // Find an element   
elem.innerHTML = x + " " + y;           // Display x and y  
  
var y = 7; // Initialize y

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_hoisting3&usg=ALkJrhgsCERFSNbLOHQkJr3KA3vhRAaUhA)

Faz sentido que y seja indefinido no último exemplo?

Isso ocorre porque somente a declaração (var y), e não a inicialização (= 7) é içada para o topo.

Por causa do levantamento, y foi declarado antes de ser usado, mas, como as inicializações não são içadas, o valor de y é indefinido.

O exemplo 2 é o mesmo que escrever:

### Exemplo

var x = 5; // Initialize x  
var y;     // Declare y  
  
elem = document.getElementById("demo"); // Find an element   
elem.innerHTML = x + " " + y;           // Display x and y  
  
y = 7;    // Assign 7 to y

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_hoisting5&usg=ALkJrhh0XpLKENmbM5YcTnq7EHshS49hOA)

## Declare suas variáveis ​​no topo!

Hoisting é (para muitos desenvolvedores) um comportamento desconhecido ou negligenciado do JavaScript.

Se um desenvolvedor não entender o levantamento, os programas podem conter erros (erros).

Para evitar erros, sempre declare todas as variáveis ​​no início de cada escopo.

Como é assim que o JavaScript interpreta o código, é sempre uma boa regra.

O JavaScript no modo estrito não permite que variáveis ​​sejam usadas se não forem declaradas.   
Estude **"use strict"** no próximo capítulo.