# JavaScript Hoisting

« Previous Next Chapter »

Hoisting is JavaScript's default behavior of moving declarations to the top.

## JavaScript Declarations are Hoisted

In JavaScript, a variable can be declared after it has been used.

In other words; a variable can be used before it has been declared.

**Example 1** gives the same result as **Example 2**:

### Example 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
```

Try it yourself »

## Example 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
```

Try it yourself »



Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope (to the top of the current script or the current function).

# JavaScript Initializations are Not Hoisted

JavaScript only hoists declarations, not initializations.

**Example 1** does **not** give the same result as **Example 2**:

## Example 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
```

Try it yourself »

## Example 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
```

Try it yourself »

---

Does it make sense that y is undefined in the last example?

This is because only the declaration (var y), not the initialization (=7) is hoisted to the top.

Because of hoisting, y has been declared before it is used, but because initializations are not hoisted, the value of y is undefined.

Example 2 is the same as writing:

## Example

```
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
```

Try it yourself »

---

# Declare Your Variables At the Top !

Hoisting is (to many developers) an unknown or overlooked behavior of JavaScript.

If a developer doesn't understand hoisting, programs may contain bugs (errors).

To avoid bugs, always declare all variables at the beginning of every scope.

Since this is how JavaScript interprets the code, it is always a good rule.

JavaScript in strict mode does not allow variables to be used if they are not declared.
Study **"use strict"** in the next chapter.

« Previous Next Chapter »





Google Maps
Animated Buttons
Modal Boxes
JS Animations
Progress Bars
Dropdowns
HTML Includes
Color Palettes

REPORT ERROR
PRINT PAGE
FORUM
ABOUT

## Top 10 Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
SQL Tutorial
PHP Tutorial
jQuery Tutorial
Bootstrap Tutorial
Angular Tutorial
ASP.NET Tutorial
XML Tutorial

## Top 10 References

HTML Reference
CSS Reference
JavaScript Reference
Browser Statistics
HTML DOM
PHP Reference
jQuery Reference
HTML Colors
HTML Character Sets
AngularJS Reference

# Top 10 Examples

HTML Examples
CSS Examples
JavaScript Examples
HTML DOM Examples
PHP Examples
jQuery Examples
XML Examples
ASP Examples
SVG Examples

# Web Certificates

HTML Certificate
HTML5 Certificate
CSS Certificate
JavaScript Certificate
jQuery Certificate
PHP Certificate
Bootstrap Certificate
XML Certificate