There are three kinds of declarations in JavaScript.

Var, let, const

Identifiers : all the variables, functions and properties name known as identifier.

You use variables as symbolic names for values in your application. The names of variables, called [identifiers](https://developer.mozilla.org/en-US/docs/Glossary/Identifier), conform to certain rules.

A JavaScript identifier must start with a letter, underscore (\_), or dollar sign ($); subsequent characters can also be digits (0-9). Because JavaScript is case sensitive, letters include the characters "A" through "Z" (uppercase) and the characters "a" through "z" (lowercase).

You can declare a variable in two ways:

* With the keyword [var](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/var). For example, var x = 42. This syntax can be used to declare both **local** and **global** variables, depending on the *execution context*.
* With the keyword [const](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const) or [let](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let). For example, let y = 13. This syntax can be used to declare a block-scope local variable. See [Variable scope](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_Types#Variable_scope) below.

You can also simply assign a value to a variable For example, x = 42. This form creates an [**undeclared global**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/var#Description) variable. It also generates a strict JavaScript warning. Undeclared global variables can often lead to unexpected behavior. Thus, it is discouraged to use undeclared global variables.

A variable declared using the var or let statement with no assigned value specified has the value of [undefined](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/undefined).

**Variable hoisting**

[**Section**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_Types#Variable_hoisting)