

wawiwa

VARS, LET, CONST

JAVASCRIPT

let var y const

La declaración **var** permite definir una variable con ámbito (scope) de función o global, opcionalmente le se puede asignar un valor.

La declaración **let** permite definir una variable con ámbito de bloque, opcionalmente se le puede asignar un valor.

Constants también están limitadas a ámbito de bloque, de manera similar a las variables declaradas usando la clave “let”. El valor de una constante no puede ser cambiada mediante la reasignación y tampoco puede ser volver a ser declarada.

var

```
<html>
<body>
<script>
  var x = 1;
  if (x == 1) {
    var x = 2;
    console.log(x);
    // resultado esperado: 2
  }
  console.log(x);
  // resultado esperado: 2
</script>
</body>
</html>
```

La declaración var permite definir una variable con ámbito de función o global, opcionalmente le se puede asignar un valor.

Las declaraciones var son procesadas antes que cualquier código sea ejecutado.

El ámbito de una variable declarada con var es el contexto de su ejecución actual y sus clausuras, que pueden ser la función contenedora y las funciones declaradas dentro de ella, o global, para variables declaradas fuera de cualquier función.

Las declaraciones duplicadas de variables usando var no van a disparar un error, incluso en el modo “strict”, y la variable no perderá su valor, a menos que se realice otra asignación.

Modo Strict (estricto) mdn

Nota: Algunas veces verás que el modo predeterminado es el modo no estricto (non-strict mode) referido como "sloppy mode". Este no es un término oficial, pero sé consciente de eso, por si acaso.

El modo Strict, introducido en ECMAScript 5, es una manera para optar por una variante restrictiva de JavaScript.

Los navegadores que no son compatibles con el modo strict correrán el código de modo strict con un comportamiento distinto que los navegadores que sí son compatibles, por lo tanto, no confíes en el modo estricto sin realizar pruebas de compatibilidad de funciones para soportar los aspectos relevantes del modo strict.

Los códigos de modo Strict y de non-strict pueden coexistir, para que los scripts puede optar por el modo strict de manera incremental.

Más sobre el modo strict -

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode

var - 'use strict'

```
<html>
<body>
  <script>
    'use strict';
    function foo() {
      var x = 1;
      function bar() {
        var y = 2;
        console.log(x); // 1 (la función `bar` cierra sobre `x`)
        console.log(y); // 2 (`y` está en el ámbito)
      }
      bar();
      console.log(x); // 1 (`x` está en el ámbito)
      console.log(y); // Error de referencia en modo estricto, `y` está en el
        ámbito de `bar`.
    }
    foo();
  </script>
</body>
</html>
```

Las declaraciones `var`, cuando ocurren, son procesadas antes de que cualquier código sea ejecutado. A esto se le llama “elevación” (hoisting), y es discutido más adelante.

El ámbito de una variable declarada con `var` es su contexto de ejecución actual y sus clausuras, que pueden ser la función contenedora o las funciones declaradas dentro de ella, o para variables declaradas fuera de cualquier función: global.

Las declaraciones duplicadas de variables “`var`” no desencadenan un error, incluso en modo “strict”, y la variable no pierde su valor, a menos que otra asignatura sea realizada.

let

```
<html>
<body>
<script>
  let x = 1;
  if (x == 1) {
    let x = 2;
    console.log(x);
    // resultado esperado: 2
  }
  console.log(x);
  // resultado esperado: 1
</script>
</body>
</html>
```

Let te permite declarar variables que están limitadas al ámbito de una declaración de bloque (block statement), o la expresión donde esté siendo usada, distinto a la claba var keyword, que declara una variable globalmente, o localmente a una función entera independientemente del ámbito del bloque. La otra diferencia entre var y let es que el último se inicializa solamente cuando un analizador lo evalpua (ver abajo)



Let - reglas de ámbito

Las variables declaradas como `let` tienen su ámbito en el bloque donde son declaradas, también en cualquier sub-bloque contenido. De esta manera `let` trabaja muy parecido a `var`. La diferencia principal es que el ámbito de una variable `var` es la función contenedora entera.

```
<html>
  <body>
    <script>
      function varTest () {
        var x = 1;

        {
          var x = 2; // la misma variable!
          console.log(x); // 2
        }

        console.log(x); // 2
      }
      function letTest () {
        let x = 1;

        {
          let x = 2; // variable distinta
          console.log(x); // 2
        }

        console.log(x); // 1
      }
    </script>
  </body>
</html>
```


const

```
<html>
<body>
  <script>
    const number = 42;

    number = 99;

    console.log(number);
  </script>
</body>
</html>
```

Las constantes tienen ámbito de bloque, de manera similar a las variables declaradas usando la palabra clave **let**. El valor de una constante no puede ser cambiado mediante reasignación y no puede ser redeclarada.



const - reglas de ámbito

```
<html>
<body>
  <script>
    // define MY_FAV como constante y dale un valor de 7
    const MY_FAV = 7;
    // esto va a arrojar un error - Uncaught TypeError: Assignment to constant
    variable.

    MY_FAV = 20;
    // MY_FAV is 7
    console.log('my favorite number is: ' + MY_FAV);
    // tratando de redeclarar una constante arroja un error
    // Uncaught SyntaxError: Identifier 'MY_FAV' has already been declared
    const MY_FAV = 20;
    // el nombre MY_FAV está reservado por una constante arriba, entonces esto
    también va a fallar

    var MY_FAV = 20;
    // esto también arroja error
    let MY_FAV = 20;
  </script>
</body>
</html>
```



Las constantes pueden ser declaradas con mayúsculas o minúsculas, pero la convención más común es usar todas mayúsculas.

A nighttime photograph of a city skyline. In the foreground, a large, curved skyscraper with a grid-like facade is illuminated with blue and white lights. Behind it, another tall building with a similar grid pattern is visible. To the right, a street with traffic lights and other buildings is shown. The sky is dark, and the city lights create a vibrant scene.

wawiwa

¿Preguntas?