



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

Cuestionario

Tecnologías Web

ALUMNO:

Melgoza de la Torre Abraham

PROFESORA

Dra. Georgina Salazar Partida

6-C

ISC

1. Menciona las 4 maneras de declarar variables en JavaScript
Automáticamente, var, let y const.
2. Traduce la siguiente imagen
It is considered good programming practice to always declare variables before use.
Está considerado una buena práctica de programación el siempre declarar variables antes de usarlas.
3. La declaración de variables con var ¿De que fecha a que fecha fue utilizada?
1996 a 2015
4. El let y const ¿A partir de qué fecha inició su uso?
2015
5. ¿Cuándo deberías usar var en las declaraciones de variables?
Cuando se usa para navegadores antiguos
6. Menciona las 5 recomendaciones de cuándo deberías usar var, let o const
 - Siempre declarar variables.
 - Utilizar const si el valor no cambia.
 - Usar const const si el tipo no debe de cambiar (arreglos y objetos).
 - Solo usar let si no puedes usar const.
 - Solo usar var si es un REQUERIMIENTO usarlo en navegadores antiguos.
7. Menciona las 5 reglas generales para escribir identificadores en JavaScript.
 - Los identificadores pueden contener letras, dígitos, barrabajas y signo de dolar
 - Los identificadores TIENEN que empezar con una letra
 - Pueden empezar con \$ y _
 - Son sensibles al capitalizado
 - Palabras reservadas
8. Traduce la siguiente imagen/
JavaScript identifiers are case-sensitive
Los identificadores de JavaScript son case-sensitive.

9. Ejemplifica dos formas de declarar variables en JavaScript

```
let var1 = 0;  
  
const PI = 3.1416;
```

10. Traduce la siguiente imagen

It's a good programming practice to declare all variables at the beginning of a script.

Es una buena práctica de la programación el declarar todas las variables al inicio de un script.

11. ¿Qué valor tiene una variable, que solo está declarada SIN asignarse valor?
undefined

12. Traduce la siguiente imagen

You cannot re-declare a variable declared with `let` or `const`.

This will not work:

```
let carName = "Volvo";  
let carName;
```

No puedes re-declarar variables con let o const.
esto no funciona:

```
let carName = "Volvo";  
  
let carName;
```

13. Traduce la siguiente imagen y ejemplifica con línea(s) de código

If you put a number in quotes, the rest of the numbers will be treated as strings, and concatenated.

Si colocas un número dentro de comillas, el resto de los números van a ser tratados como una cadena y serán concatenados.

14. Traduce la siguiente imagen

Using the underscore is not very common in JavaScript, but a convention among professional programmers is to use it as an alias for "private (hidden)" variables.

Utilizar la barra baja no es común dentro de JavaScript, pero es una convención dentro de los programadores profesionales para utilizarla para variables "privadas (escondidas)".

15. Desarrollar el tema JavaScript Hoisting, a partir del siguiente link

En JavaScript una variable puede ser declarada después de ser usada, en otras palabras; una variable puede ser utilizada antes de haber sido declarada:

```
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
```

```
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
```

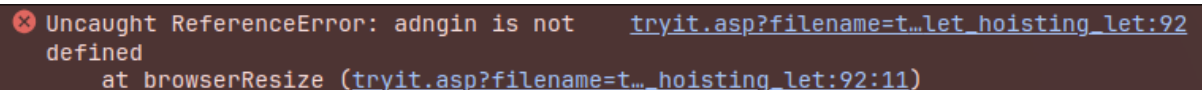
Resultan lo mismo.

Para comprender esto, tenemos que entender el término “hoisting”, el cual es un comportamiento natural de JS de mover todas las declaraciones a la parte superior dentro del alcance actual.

En otras palabras, las variables definidas con *let* y *const* son “hoisted” al tope del bloque, pero no son *inicializadas*.

Pero no van a poder ser usadas hasta que no sean inicializadas.

Utilizar *let* antes de ser declarada va a resultar en un Reference Error

A screenshot of a browser's developer console showing a red error message. The message reads: "Uncaught ReferenceError: adngin is not defined". Below this, it shows the file and line number: "tryit.asp?filename=t...let_hoisting_let:92". At the bottom, it indicates the location within the file: "at browserResize (tryit.asp?filename=t..._hoisting_let:92:11)". There is a small icon of a red 'x' in a circle to the left of the error message.

```
Uncaught ReferenceError: adngin is not defined
    tryit.asp?filename=t...let_hoisting_let:92
    at browserResize (tryit.asp?filename=t..._hoisting_let:92:11)
```

Y aquellas declaradas con *const* simplemente resulta en un error de sintaxis y el código no se podrá ejecutar.

Es importante remarcar que javascript solo realizará *hoist* a las declaraciones y no inicializaciones de variables.

```
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
```

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

var y = 7; // Initialize y
```

No resultan lo mismo, la segunda entrega indefinido en y ya que aun no se inicializa con el valor 7. Por lo tanto resultaría como escribir lo siguiente:

```
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
```