Francisco Ramírez Ruiz

VARIABLES / ÁMBITOS VAR, LET / CONST, HOISTING

ÍNDICE

Francisco Ramírez Ruiz

ENUNCIADO

Lee los siguientes artículos y responde:

- Let, la nueva forma de declarar variables en Javascript
- From var to const/let
- Hoisting en JavaScript
- Variables and scoping
- Speaking JavaScript

1. **Indica verdadero o falso:**
    a. En JavaScript, podemos hacer uso de las variables sin haberlas declarado previamente. En ese caso, dicha variable es global, independientemente de donde se haya usado.
    b. Las variables sin declarar se convierten en propiedad del objeto document
    c. const tiene ámbito de bloque
2. **Contenido de una variable cuando todavía no se ha inicializado.**
3. **Ámbito de la declaración let.**
4. **En qué casos tiene un claro sentido la declaración de variables mediante let.**
5. **Indica si el hoisting le sigue afectando al let. ¿Y al const?**
6. **Indica en qué casos puede repetirse la declaración de una variable: funciones anidadas, bloques anidados, ámbito global...**
7. **Qué limita la directiva 'use strict'; a la declaración de variables.**
8. **Indica qué es la TDZ y a qué declaraciones afecta.**

Francisco Ramírez Ruiz

1.- VERDADERO O FALSO

- En JavaScript, podemos hacer uso de las variables sin haberlas declarado previamente. En ese caso, dicha variable es global, independientemente de donde se haya usado. **Verdadero**
- Las variables sin declarar se convierten en propiedad del objeto document. **Falso, se convierte en propiedad del objeto global**
- const tiene ámbito de bloque. **Verdadero**

2.-CONTENIDO DE UNA VARIABLE SI NO SE HA INICIALIZADO.

Tiene el contenido de: **undefined.**

3.-ÁMBITO DE LA DECLARACIÓN LET

Tiene un ámbito de: **bloque.**

4.-EN QUÉ CASOS TIENE UN CLARO SENTIDO LA DECLARACIÓN DE VARIABLES LET

En las condiciones de bloque por ejemplo de un if

```
if(condición){
        let x = “Hola”;
}
```

Si intentamos acceder a x nos saldrá un error de que no está definida.

5.-INDICA SI HOISTING SIGUE AFECTANDO AL LET ¿Y AL CONST?

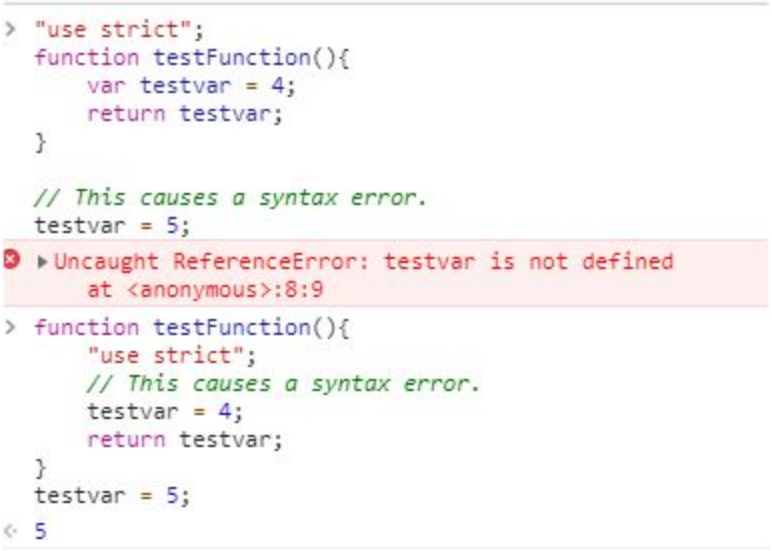
Si, sigue afectado a let y a const dejando las declaraciones en zona muerta temporalmente hasta que estas son declaradas dependiendo del bloque.

6.-INDICA EN QUÉ CASOS PUEDE REPETIRSE LA DECLARACIÓN DE UNA VARIABLE.

De forma global sin nada o con var, siempre lo permite a no ser que esté declarada por let o const y en las declaraciones de ámbito local siempre que estén en distintos bloques.

7.-QUÉ LIMITA LA DIRECTIVA 'USE STRICT' A LA DECLARACIÓN DE VARIABLES.

Si se declara fuera del ámbito de una función todo el programa está en modo strict. Si es en una función la función estaría en modo strict.



8.-INDICA QUE ES LA TDZ Y A QUE DECLARACIONES AFECTA.

Afecta a let y a const

La diferencia entre var con let y const es que tienen ámbito de bloque y además acceder a un var antes de declararlo da undefined sin embargo en let y const da ReferenceError

Por lo cual la TDZ Temporal Dead Zone, es el tiempo muerto que tiene la variable en ser declarada.