Javascript : Hosting

1. Introduction

Hoisting peut se traduire comme remonté.

Cela correspond au déplacement de la déclaration d’un fonction, classes, variables etc… en haut de leur portée avant l’execution du script.

Ce n’est pas un terme utilisé dans la specification ECMAScript.

Ce système de remonté est souvent associé aux variables déclarées avec var.

On retrouve ce type de comportement :

* Possibilité d’utilisé une variable dans sa portée avant la ligne ou elle est déclarée. (var hoisting)
* Possibilté de référence d’un variable dans sa portée avant la ligne ou elle est déclarée sans que cela déclanche une ReferenceError. (declaration hoisting)
* La déclaration de la variable entraine des changements de comportement dans la portée avant la ligne ou elle est déclarée.

Sans ‘use strict’ les variables déclarées sans mot-clé sont remontées selon le premier comportement.

Avec var avec selon le deuxieme comportement.

Let, const, class, se comporte comme le troisieme comportement.

On peut considéré que let, const, class ne déclenchent pas de hoisting (zone morte temporaire) qui interdit toute utilisation de la variable avant sa déclaration.

Cependant cela peut engendrer des comportements particuliers :

const x = 1;

{

console.log(x); // ReferenceError

const x = 2;

}