1. Le variable Let est un type de donné qui est utilisé pour ne pas permettre la re déclaration

de la variable.

Let préserve le donné assigné.

Dans le fichier .html, les variables ne sont pas déclarées plusieurs fois, qui ne cause pas de

plantages.

Le log sur la console si on remplace var par let que dans la é région #Type Simples :

Jacques 80 something

Jacques 81 14

(3) ['apple', 'pear', 'cherry'] (3) ['potato', 'curlyflower', 'tomato']

(4) ['apple', 'pear', 'cherry', 'banana'] (2) ['potato', 'curlyflower']

{sound: true, music: false, graphics: 'high', resolution: Array(2)} {time: 384, score: 47, opponent: 'Jacques'}

{sound: true, music: true, graphics: 'high', resolution: Array(2)} {}

3 The number is 3

2. Une première ligne est affichée sur la console -> Jacques 80 something

Puis la session plante vu qu'un constant ne peut pas être incrémenté ou re assigné

Le plantage se produit lors de l'exécution de la ligne 19 : age += 1

3. Aucunes erreurs n’est constaté parce que const ne définit pas le tableau comme étant un constant mais plutôt la référence du tableau.

Pour être plus claire, on peut modifier le contenu de chaque indexe du tableau mais on ne peut pas redéfinir le tableau.

Exemple :  fruits.push("banana"); ne change pas la référence du tableau

* fruits[3] = « banana »

On peut modifier le contenu du tableau mais faire ceci :

const fruits = ["apple", "pear", "cherry"];

const vegetables = ["potato", "curlyflower", "tomato"];

fruits.push("banana");

console.log(fruits);

fruits = ["apple", "pear", "cherry"];

console.log(fruits)

Sur la console :

Une image contenant texte

Description générée automatiquement

5. Vu qu’on modifie les valeurs de l’objet savedGames au complèt, on a un plantage.

Même comportement que les tableaux.

6. <!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta http-equiv="X-UA-Compatible" content="ie=edge" />

    <title>constiables!</title>

  </head>

  <body>

    <script>

      //1.1 Types simples

      const name = "Jacques";

      let age = 80;

      let useless = "something";

      console.log(name, age, useless);

      age += 1;

      useless = 14;

      console.log(name, age, useless);

      //----const------------------//

      //1.2 Tableaux

      const fruits = ["apple", "pear", "cherry"];

      const vegetables = ["potato", "curlyflower", "tomato"];

      console.log(fruits, vegetables);

      fruits.push("banana");

      vegetables.pop();

      console.log(fruits, vegetables);

      //----------------------//

      //1.3 Objets

      const settings = {

        sound: true,

        music: false,

        graphics: "high",

        resolution: [1920, 1080],

      };

      let savedGame = {

        time: 384,

        score: 47,

        opponent: "Jacques",

      };

      console.log(settings, savedGame);

      savedGame = {};

      settings.music = true;

      settings.resolution = [3840, 2160];

      console.log(settings, savedGame);

      //----------------------//

      //1.4 Portée

      for (var i = 0; i < 3; i++) {

        setTimeout(function () {

          console.log("The number is " + i);

        }, 1000);

      }

    </script>

  </body>

</html>

7. Si on remplace le type de donnée de #Portée par let, on a un résultat différent :

La dernière ligne est affichée 3 fois qui est cohérent vu que Var i = 0 devient let i = 0 et i est

incrémenté.

Du coup, il re affiche la ligne pour le nombre de fois qu’on fait la boucle avec les nouvelles

valeurs de ‘i’.

The number is 0

The number is 1

The number is 2

8.

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta http-equiv="X-UA-Compatible" content="ie=edge" />

    <title>constiables!</title>

  </head>

  <body>

    <script>

      //1.1 Types simples

      const name = "Jacques";

      let age = 80;

      let useless = "something";

      console.log(name, age, useless);

      age += 1;

      useless = 14;

      console.log(name, age, useless);

      //----const------------------//

      //1.2 Tableaux

      const fruits = ["apple", "pear", "cherry"];

      const vegetables = ["potato", "curlyflower", "tomato"];

      console.log(fruits, vegetables);

      fruits.push("banana");

      vegetables.pop();

      console.log(fruits, vegetables);

      //----------------------//

      //1.3 Objets

      let settings = {

        sound: true,

        music: false,

        graphics: "high",

        resolution: [1920, 1080],

      };

      let savedGame = {

        time: 384,

        score: 47,

        opponent: "Jacques",

      };

      console.log(settings, savedGame);

      savedGame = {};

      settings.music = true;

      settings.resolution = [3840, 2160];

      console.log(settings, savedGame);

      //----------------------//

      //1.4 Portée

       for (var i = 0; i < 3; i++) {

        setTimeout(function () {

          console.log("The number is " + i);

        }, 1000);

      }

    </script>

  </body>

</html>