



Compte Rendu : Atelier Pratique

JavaScript

Établissement : École Supérieure de Technologie de Nador

Filière : ILCS | **Module :** Technologie du Web

Feit par: Naoual aouissi et omayma elmir encadrée par prof: ouzayr rabhi

Introduction

Ce rapport présente les solutions et concepts abordés lors des séances de travaux pratiques sur le langage JavaScript, couvrant les structures de contrôle, les boucles, les fonctions, les objets et la manipulation de données JSON.

TP 1 : Structures de Contrôle (if, else, switch)

Exercice 1 : Calculateur de Réduction

- **Objectif :** Appliquer un taux de remise selon le montant des achats.
- **Logique appliquée :**
 - Moins de 200 Dh : 5% de réduction.
 - Entre 200 Dh et 500 Dh : 10% de réduction.
 - Plus de 500 Dh : 20% de réduction.
- **Concepts utilisés :** prompt() pour la saisie, Number() pour la conversion de type, et console.log() pour le résultat.

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing a project structure with files like 'exercice1.html', 'exercice1.js', 'exercice2.html', etc. In the center is the main editor area with the following code:

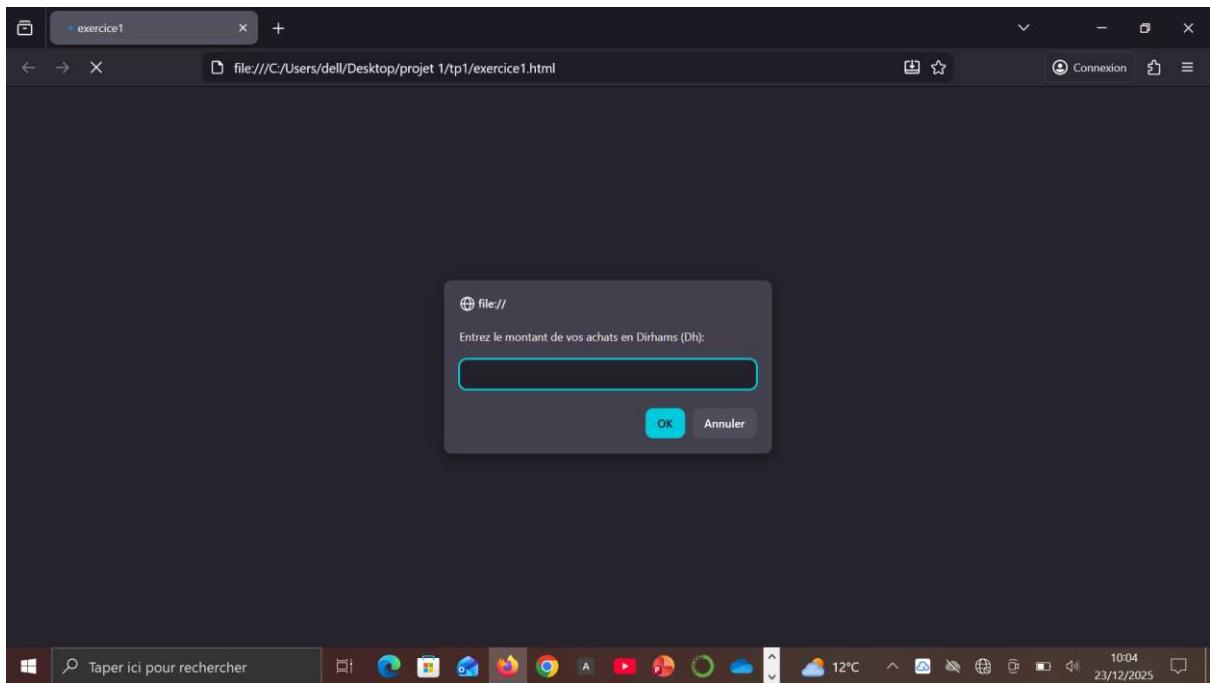
```
// Étape 1: Demander le montant des achats
let montantAchats = Number(prompt("Entrez le montant de vos achats en Dirhams (Dh):"));

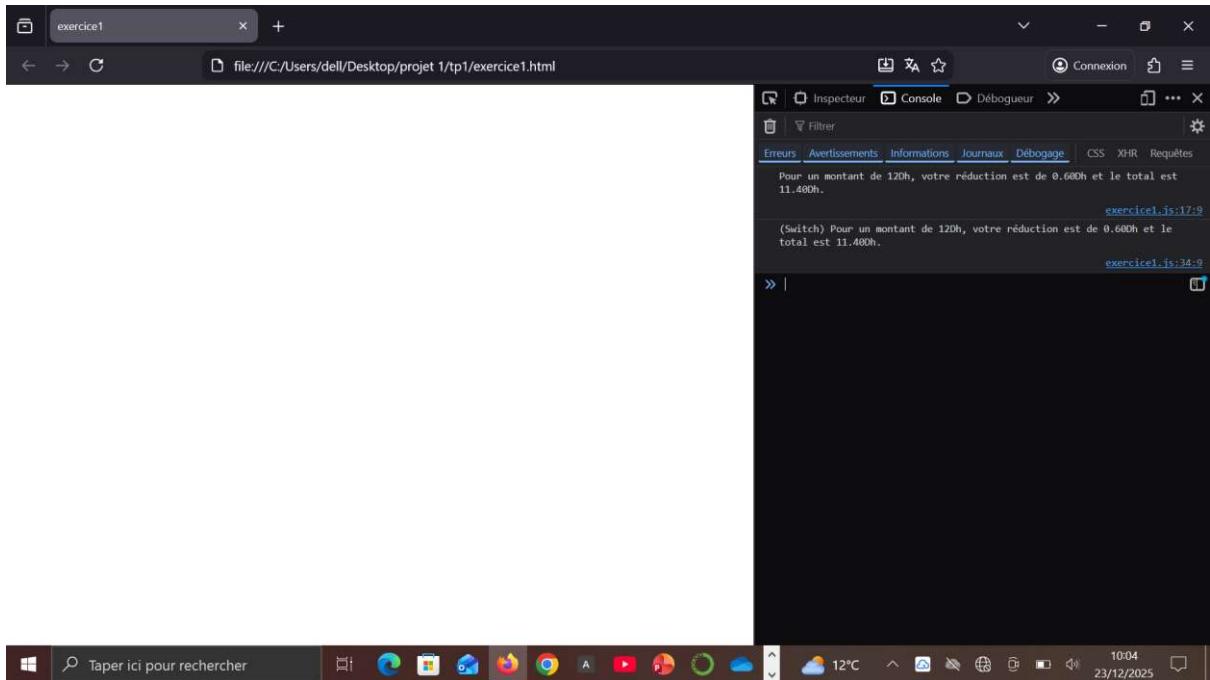
// Étape 2: Calculer la réduction
let reduction = 0;

if (montantAchats < 200) {
    reduction = montantAchats * 0.05;
} else if (montantAchats >= 200 & montantAchats < 500) {
    reduction = montantAchats * 0.10;
} else {
    reduction = montantAchats * 0.20;
}

// Étape 3: Afficher le résultat
let total = montantAchats - reduction;
console.log(`Pour un montant de ${montantAchats} Dh, votre réduction est de ${reduction} Dh. Votre total à verser est ${total} Dh`);
```

On the right side, there's a "Build with Agent" panel with a message about AI responses being inaccurate and a "Generate Agent Instructions" button. Below it are "SUGGESTED ACTIONS" like "Build Workspace" and "Show Config". A tooltip suggests building the file "exercice1.js". The status bar at the bottom shows the file has 25 lines, 11 spaces, is in UTF-8 encoding, and was last saved on 22/12/2025 at 15:39.





Exercice 2 : Validation d'Utilisateur

- Objectif :** Évaluer la force d'un mot de passe via sa propriété `.length`.
- Critères :** Faible (<8), Moyen (8-12), Fort (>12).
- Structure :** Utilisation d'une boucle `while` pour permettre à l'utilisateur de tester plusieurs mots de passe jusqu'à la saisie de "exit".

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows a project structure for "PROJET 1" containing subfolders "tp1", "tp2", "tp3", and "tp4", each with various files like "exercice1.html", "exercice1.js", etc.
- EDITOR:** The main editor pane displays the content of "exercice2.js". The code is as follows:

```

//Exercice 2:
let motDePasse = "";

while (motDePasse !== "exit") {
    motDePasse = prompt("Entrez un mot de passe (ou tapez exit pour quitter)");
    if (motDePasse === "exit") {
        alert("Au revoir !");
        break;
    }

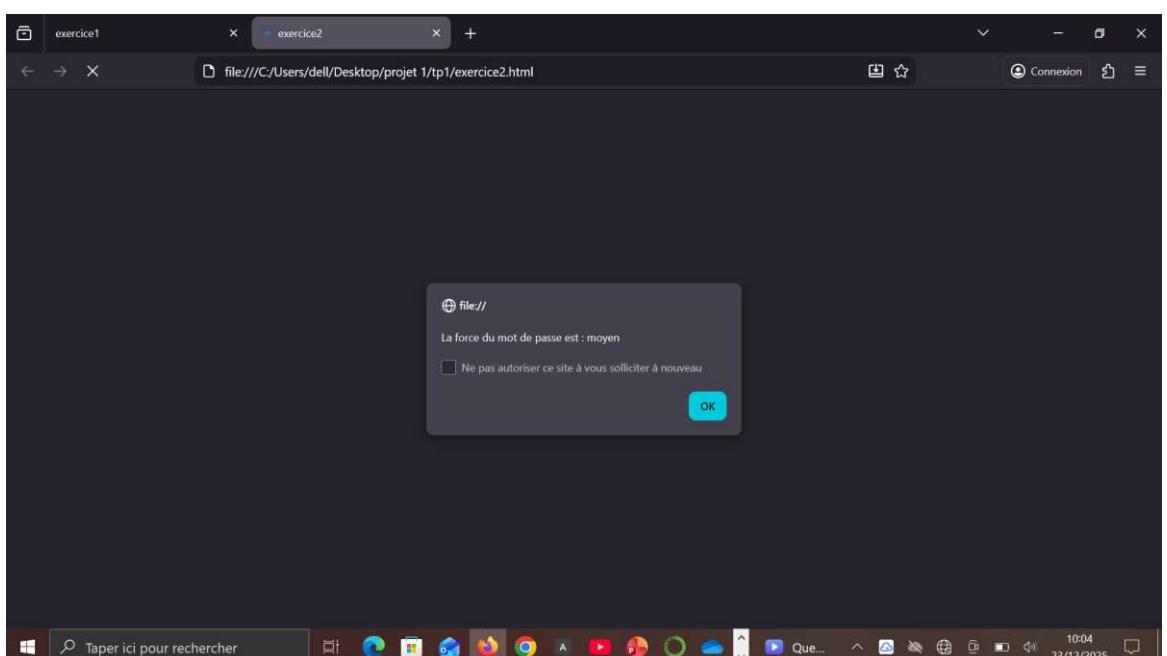
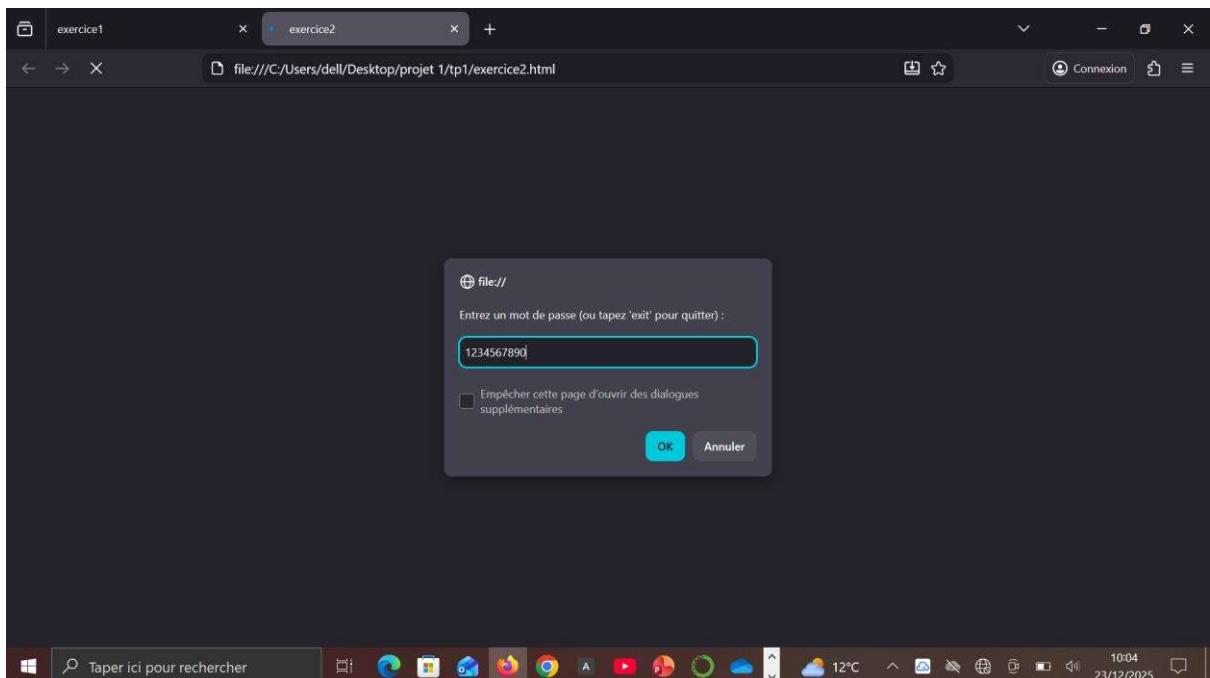
    let longueur = motDePasse.length;
    let force = "";

    if (longueur < 8) {
        force = "faible";
    } else if (longueur <= 12) {
        force = "moyen";
    } else {
        force = "fort";
    }

    alert("La force du mot de passe est : " + force);
}

```

- OUTPUT:** A "Build with Agent" panel is visible on the right, providing AI-generated instructions for building the workspace.
- STATUS BAR:** Shows the file path "exercice2.js", line 24, column 6, and other standard status information.



-
- **TP 2 : Boucles Avancées (for, while, do-while)**
- **Exercice 3 : Tableaux Multiples**
- **Objectif :** Générer la table de multiplication d'un nombre jusqu'à 10.
- **Particularité :** Utilisation du mot-clé continue pour ignorer les résultats impairs et n'afficher que les produits pairs dans la console.

The screenshot shows a Windows desktop environment with a code editor (VS Code) and a web browser window.

Code Editor (VS Code):

- File Explorer:** Shows a project structure named "PROJET 1" containing several files: exercice1.html, JS exercice1.js, exercice2.html, JS exercice2.js, tp1, tp2, tp3, tp4, and tp5. The file "exercice3.js" is selected.
- Editor:** Displays the content of "exercice3.js". The code is as follows:

```
1 //Exercice3;
2 // Demander un nombre à l'utilisateur
3 let nombre = parseInt(prompt("Entrez un nombre :"));
4
5 // Vérification simple
6 if (isNaN(nombre)) {
7     console.log("Veuillez entrer un nombre valide.");
8 } else {
9     // Boucle for de 1 à 10
10    for (let i = 1; i <= 10; i++) {
11        let resultat = nombre * i;
12
13        // Ignorer les résultats impairs
14        if (resultat % 2 !== 0) {
15            continue;
16        }
17
18        // Afficher uniquement les résultats pairs
19        console.log(nombre + " x " + i + " = " + resultat);
20    }
21 }
22 }
```

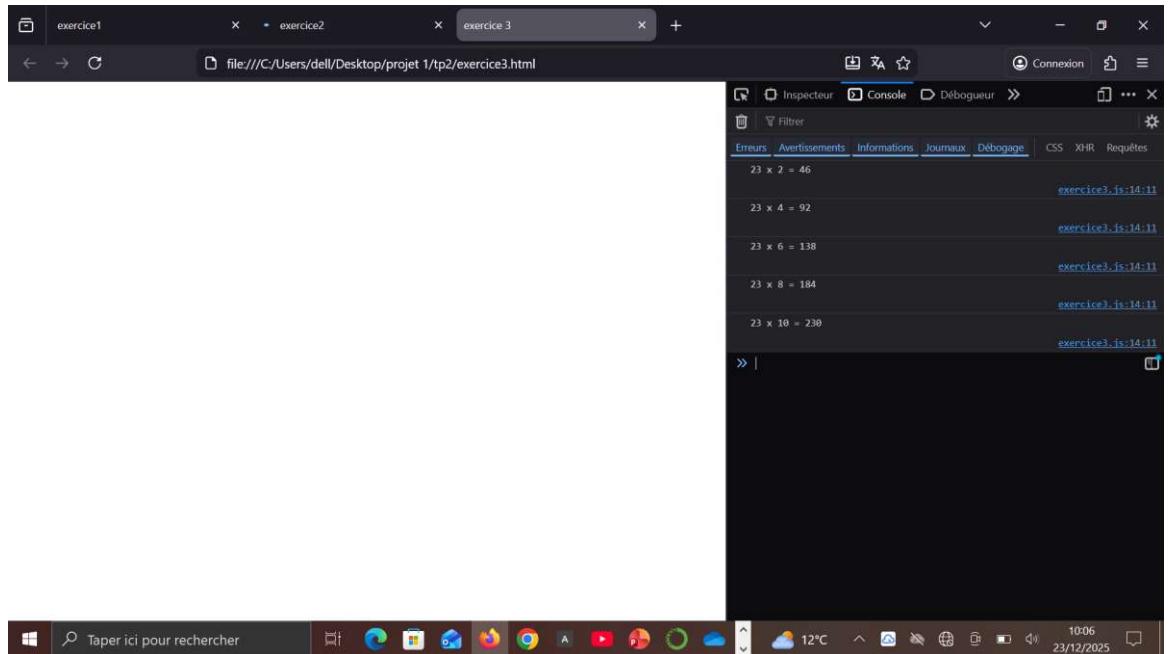
- Suggested Actions:** Includes "Build Workspace" and "Show Config".
- Bottom Status Bar:** Shows "Ln 22, Col 1" and other system information.

Browser Window:

- Address Bar:** Shows "file:///C:/Users/dell/Desktop/projet 1/tp2/exercice3.html".
- Content Area:** Displays a modal dialog box with the following text:

file:///
Entrez un nombre pour afficher sa table de multiplication:
23|

OK Annuler
- Taskbar:** Shows the Windows Start button, a search bar, and various pinned application icons.

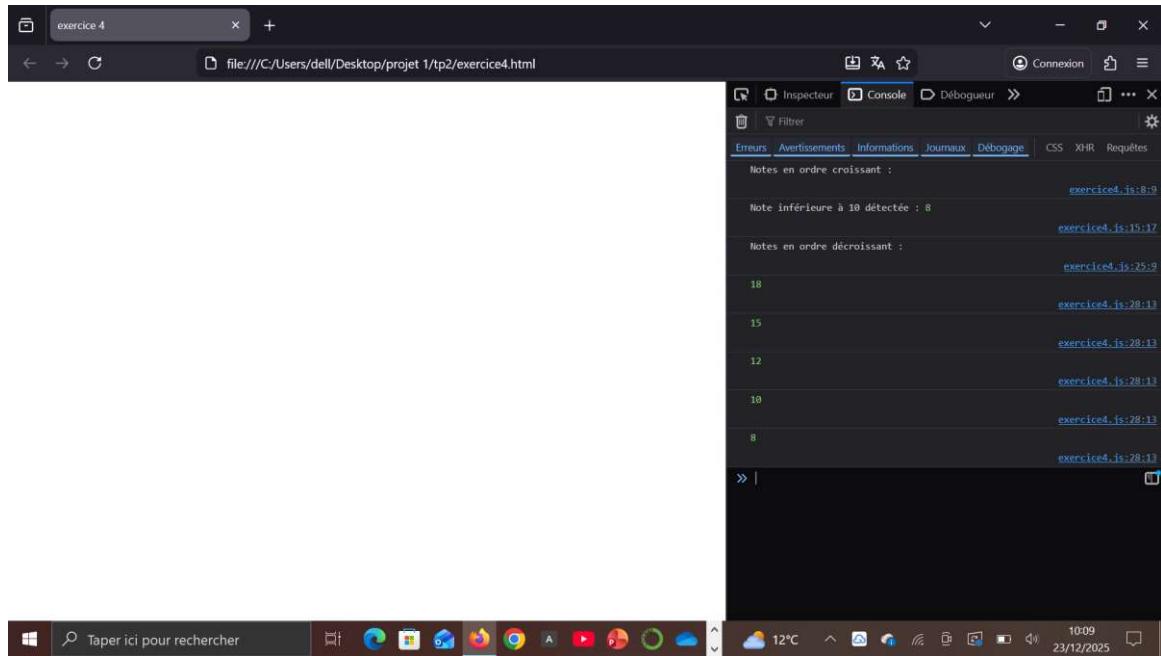


- Exercice 4 : Tri des Notes
- Données : Un tableau prédéfini [10, 15, 8, 18, 12].
- Manipulation : Emploi des méthodes `sort()` pour l'ordre croissant et `reverse()` pour l'ordre décroissant.
- Sécurité : Arrêt immédiat du script avec `break` si une note est inférieure à 10.

The screenshot shows a code editor interface with the project structure on the left and the code editor on the right. The code editor has tabs for 'exercice3.html', 'exercice3.js', 'exercice4.html', and 'JS exercice4.js'. The 'JS exercice4.js' tab is active, displaying the following JavaScript code:

```
//Exercice 4:
// Tableau prédéfini des notes
let notes = [10, 15, 8, 18, 12];
// Tri par ordre croissant
notes.sort((a, b) => a - b);
console.log("Notes en ordre croissant :");
// Parcours avec boucle
for (let i = 0; i < notes.length; i++) {
    // Si une note est inférieure à 10, arrêter le script
    if (notes[i] < 10) {
        console.log("Note inférieure à 10 détectée !");
        break;
    }
    console.log(notes[i]);
}
// Tri par ordre décroissant
notes.reverse();
console.log("Notes en ordre décroissant :");
for (let i = 0; i < notes.length; i++) {
    console.log(notes[i]);
}
```

The code uses the `sort()` method to sort the array in ascending order and the `reverse()` method to sort it in descending order. It includes a `for` loop to iterate through the array and a `if` statement to check if a note is less than 10, in which case it breaks out of the loop. The code editor also shows an AI assistant panel on the right with options like 'Build with Agent' and 'SUGGESTED ACTIONS'.



- TP 3 : Fonctions et Expressions Lambdas
- Exercice 5 : Calcul des Moyennes
- Procédure : Récupération de 5 notes successives stockées dans un tableau via `push()`.
- Calcul : Création d'une fonction classique, puis transformation en fonction fléchée (lambda) pour calculer la moyenne.
- Contrôle : Ignorer les notes négatives avec un message d'erreur.

```

File Edit Selection View Go ... < > Q projet 1
EXPLORER ... dice 5.html exercice6.html JS exercice6.js JS exercice 5 js CHAT + ...
PROJET 1 tp3 > JS exercice 5 js ...
1 //Exercice 5:
2 // Tableau pour stocker les notes
3 let notes = [];
4
5 // Saisie des 5 notes
6 for (let i = 1; i <= 5; i++) {
7   let note = parseFloat(prompt("Entrez la note " + i));
8   notes.push(note);
9 }
10
11 // Fonction fléchée pour calculer la moyenne
12 const calculerMoyenne = (tableauNotes) => {
13   let somme = 0;
14   let compteur = 0;
15
16   for (let i = 0; i < tableauNotes.length; i++) {
17     if (tableauNotes[i] >= 0) {
18       somme += tableauNotes[i];
19       compteur++;
20     } else {
21       alert("Erreur : la note " + tableauNotes[i]);
22     }
23
24   }
25
26   if (compteur === 0) {
27     return "Aucune note valide";
28   }
29
30   return somme / compteur;
}

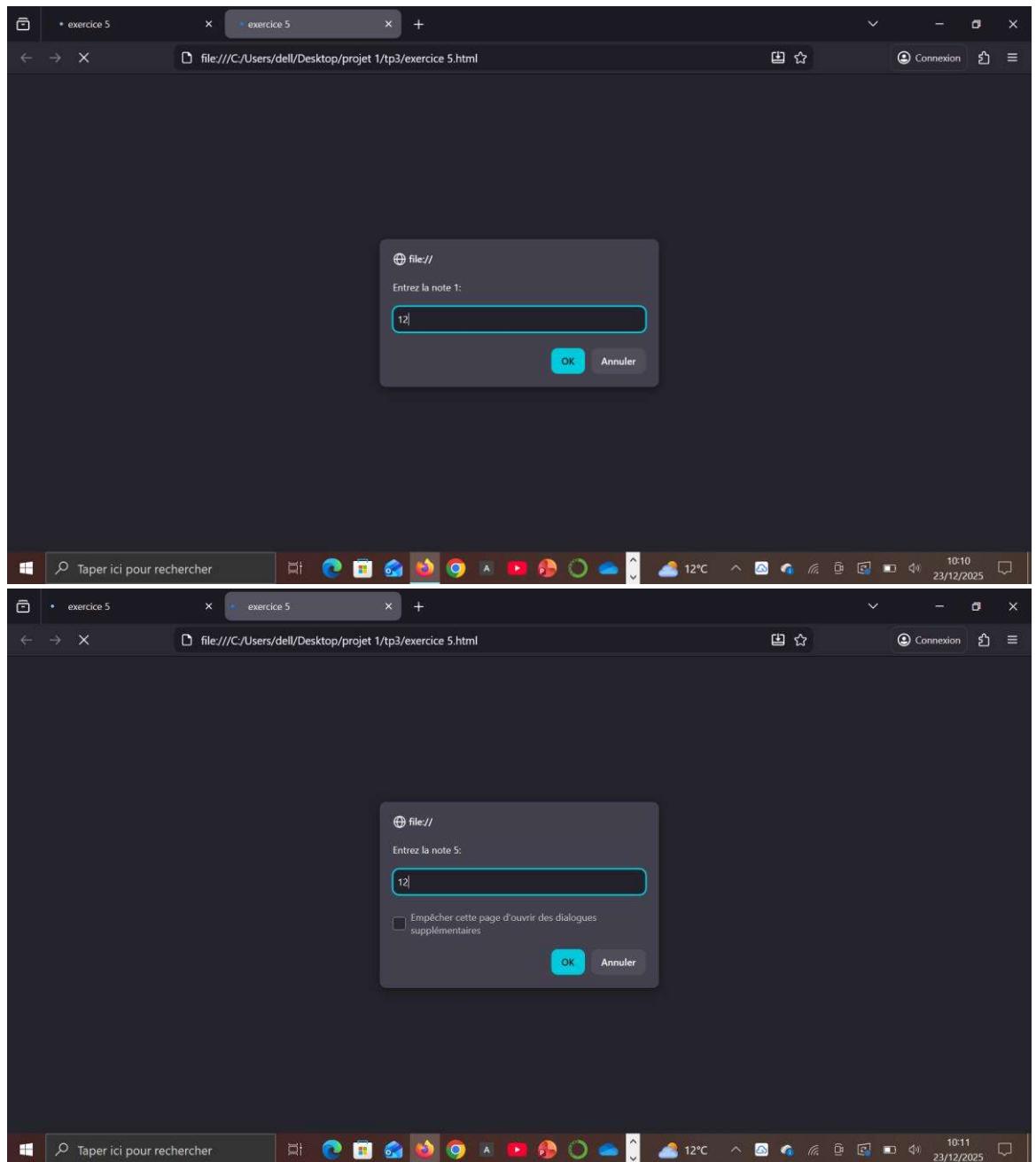
```

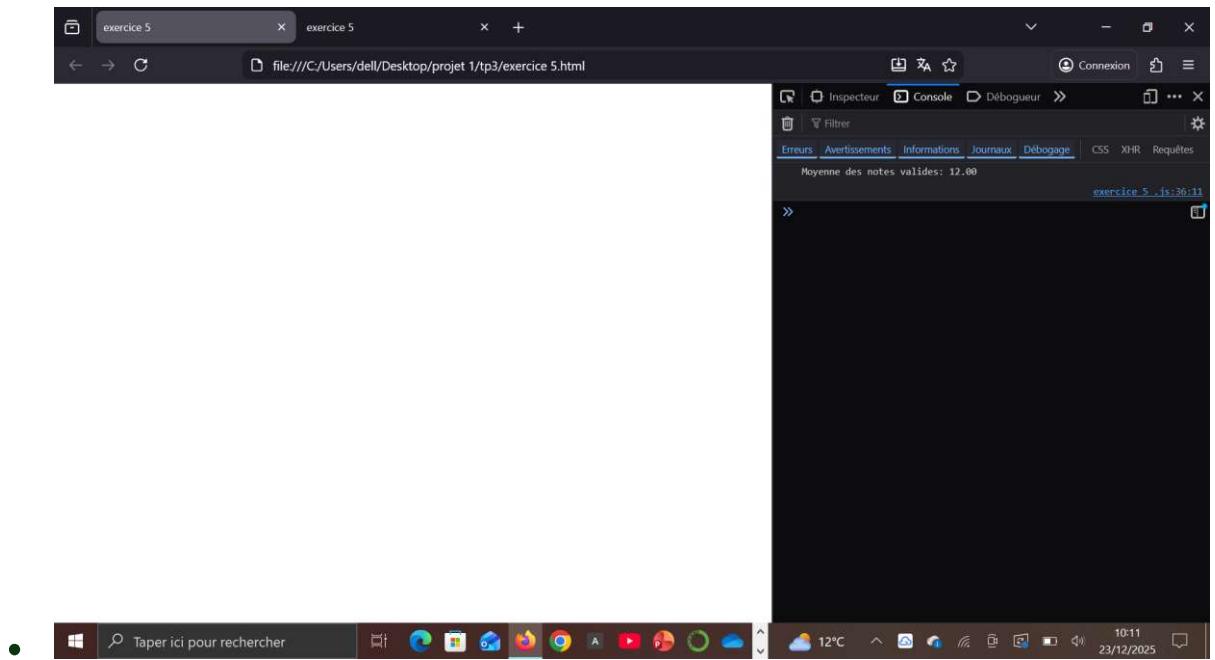
SUGGESTED ACTIONS

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

Build Workspace Show Config

JS exercise 5.js +
Describe what to build next
Agent Auto 1540 22/12/2025





TP 4 : Objets, Tableaux et Sécurité

- **Exercice 6 : Catalogue de Produits**
- **Modèle :** Tableau d'objets contenant les propriétés : nom, prix, et quantité.
- **Méthodes :**
 - **afficherProduits()** : Parcourt la liste avec une boucle `for...of`.
 - **vendreProduit()** : Diminue le stock ou affiche une erreur si l'inventaire est insuffisant.

The screenshot shows a Windows desktop environment. At the top, a browser window titled "exercice 6" displays the URL "file:///C/Users/dell/Desktop/projet 1/tp3/exercice6.html". The page content is a simple list of products: Clavier (Prix: 150 DH, Quantité: 10), Souris (Prix: 80 DH, Quantité: 20), and Écran (Prix: 1200 DH, Quantité: 5). Below the browser is a code editor window titled "exercice6.js" with the following code:

```

25 }
26
27 // 3. Fonction pour vendre un produit
28 function vendreProduit(nomProduit, quantiteVendue) {
29   for (let produit of produits) {
30     if (produit.nom === nomProduit) {
31
32       if (quantiteVendue <= 0) {
33         console.log("Erreur : la quantité doit être positive");
34         return;
35       }
36
37       if (produit.quantite >= quantiteVendue) {
38         produit.quantite = produit.quantite - quantiteVendue;
39         console.log(`Quantité vendue : ${quantiteVendue} unité(s) de ${produit.nom}`);
40       } else {
41         console.log(
42           `Erreur : stock insuffisant pour ${produit.nom}. Il y a ${produit.quantite} unité(s) disponibles.`);
43       }
44     }
45   }
46   return;
47 }
48
49
50 }
51
52 }
53

```

The code editor also includes a "CHAT" panel with the message "Build with Agent" and a "SUGGESTED ACTIONS" panel with the option "Build Workspace". The status bar at the bottom of the code editor shows "Ln 53, Col 1 Spaces: 4 UTF-8 CRLF" and the date "22/12/2025".

- **Exercice 7 : Saisie Sécurisée**
- **Objectif :** Validation d'un nombre entre 1 et 100.
- **Gestion d'erreurs :** Implémentation d'un bloc `try...catch` avec `throw` pour lever des exceptions si la saisie est `isNaN()` ou hors plage.

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with various icons. Below the taskbar, a code editor window titled "projet 1" is open, showing several files in the sidebar and the content of "exercice7.html" in the main pane. The code uses JavaScript to handle user input for a number between 1 and 100. In the bottom right corner of the code editor, there is an AI integration panel titled "Build with Agent". The browser window below shows the result of running the code, displaying a prompt asking for a number between 1 and 100, with the number "12" entered.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Exercice 7 - Saisie Sécurisée</title>
6 </head>
7 <body>
8
9   <h2>Exercice 7 : Saisie Sécurisée</h2>
10  <button onclick="saisieSecurisee()">Entrer un nombre</button>
11
12 <script>
13   function saisieSecurisee() {
14     try {
15       let saisie = prompt("Entrez un nombre entre 1 et 100 :");
16       let nombre = Number(saisie);
17
18       if (isNaN(nombre)) {
19         throw "Erreur : Vous devez entrer un nombre.";
20       }
21
22       if (nombre < 1 || nombre > 100) {
23         throw "Erreur : Le nombre doit être entre 1 et 100.";
24       }
25
26       alert("Nombre valide : " + nombre);
27
28     } catch (erreur) {
29       alert(erreur);
30     }
31   }
32 </script>
```

Exercice 7 : Saisie Sécurisée

Entrer un nombre

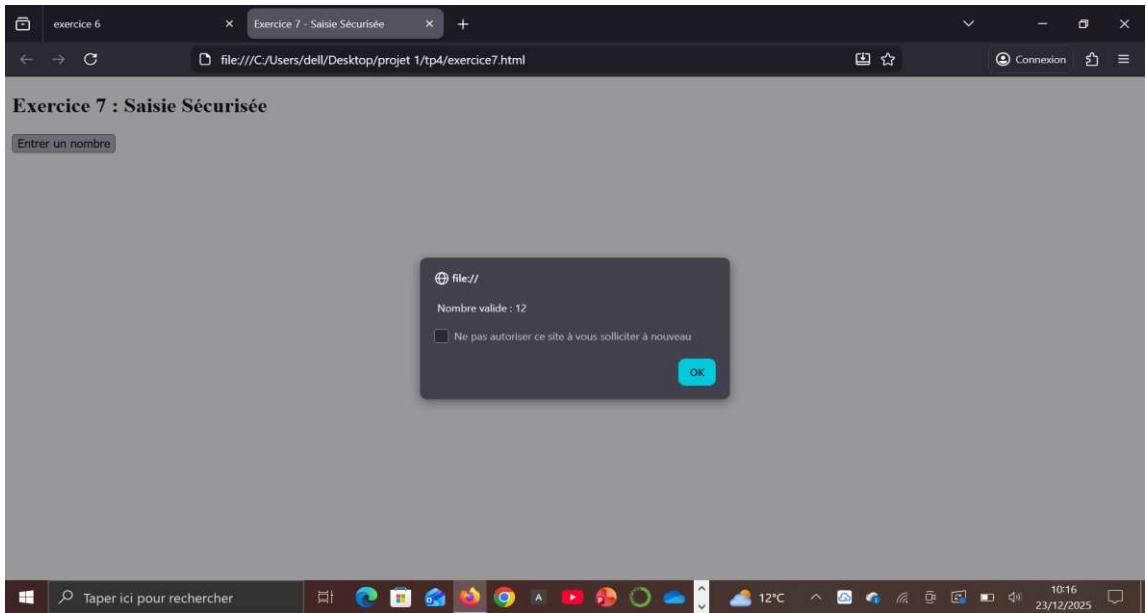
file:///C:/Users/dell/Desktop/projet 1/tp4/exercice7.html

file://

Entrez un nombre entre 1 et 100 :

12

OK Annuler



- Exercice 8 : Manipulation JSON

- Flux de données :

- Saisie des infos (nom, âge, hobbies).
- Conversion vers JSON avec `JSON.stringify()`.
- Analyse d'un JSON reçu avec `JSON.parse()` pour simuler une réponse serveur.

```

File Edit Selection View Go ...
exercice1.html exercise1.js exercice2.html exercise2.js exercice7.html exercise8.html CHAT
tp4 > exercise8.html ...
1 <html lang="fr">
2   <body>
3     <script>
4       function gestionJSON() {
5         // 1. Saisie des informations
6         let nom = prompt("Entrez votre nom :");
7         let age = prompt("Entrez votre âge :");
8         let hobbies = prompt("Entrez vos hobbies (séparés par des virgules) :");
9
10        let utilisateur = {
11          nom: nom,
12          age: Number(age),
13          hobbies: hobbies.split(",")
14        };
15
16        // 2. Conversion en JSON
17        let jsonUtilisateur = JSON.stringify(utilisateur);
18        console.log("JSON envoyé : ", jsonUtilisateur);
19
20        // 3. Simulation de réception JSON
21        let jsonReçu = '{"nom":"Sara","age":22,"hobbies":["Lecture","Sport","Musique"]}'
22        let utilisateurReçu = JSON.parse(jsonReçu);
23
24        console.log("JSON reçu : ", utilisateurReçu);
25        console.log("Nom : ", utilisateurReçu.nom);
26        console.log("Âge : ", utilisateurReçu.age);
27        console.log("Hobbies : ", utilisateurReçu.hobbies.join(", "));
28      }
29    </script>
30
31
32
33
34
35
36
37
38
39

```

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

SUGGESTED ACTIONS
Build Workspace Show Config
Describe what to build next
Agent Auto

