

JavaScript



TP 4: JavaScript

Partie 2: Manipulation du DOM



Spécifications

- L'objectif de ce TP est de vous familiariser les méthodes/propriétés Javascript pour la manipulation du DOM
- Vous allez mettre en pratique ces notions à travers la réalisation d'une mini-application « **CRUD-Persons** » à travers 6 étapes (chacun contenant déjà un fichier index.html)
- Le fichier **TP4_JS.pdf** vous guidera étape par étape pour la concrétisation de ce TP.
- Tous les scripts JS doivent être fait dans un fichier externe script.js
- Chaque script doit contenir au tout début : **"use strict";**
- Votre code doit impérativement être commenter, en mentionnant pour chaque concept évoqué, le numéro de slide correspondant dans le cours (ex. Ici on utilise une Boucle, vu dans le slide 45).
- L'énoncé du TP est déposé sur Github.
- Le lien de l'assignement: est donné dans google classroom.
- **Livrables attendus:**
 - 6 dossiers, un pour chaque étape.
 - Dans chaque dossier doit se trouver: Index.html (et style.css & style script.js quand c'est demandé)



Objectifs Spécifications

On voudrait réaliser l'application suivante

Nom:

Prénom:

Points:

Ajouter

N°	Nom	Prénom	Points	Select
1	nom1	prenom1	5	<input type="checkbox"/>
2	nom2	prenom2	10	<input type="checkbox"/>
3	nom3	prenom3	15	<input type="checkbox"/>

3 ligne(s)

Total point(s)= 30

Console Tableau

Supprimer

Objectifs Spécifications

On voudrait réaliser l'application suivante

1

Un formulaire permettant de remplir le nom, prénom, points d'une nouvelle personne qu'on souhaiterait rajouter.

2

Un tableau qui contient une liste de personnes, identifiées par leur nom, prénom, points.

3

Un espace résumant le contenu du tableau: nombre de lignes, total de points

4

Des boutons de contrôle:

Nom: <input type="text"/>		Prénom: <input type="text"/>		Points: <input type="text"/>		<input type="button" value="Ajouter"/>	
N°	Nom		Prénom		Points	Select	
1	nom1		prenom1		5	<input type="checkbox"/>	
2	nom2		prenom2		10	<input type="checkbox"/>	
3	nom3		prenom3		15	<input type="checkbox"/>	
3 ligne(s)					Total point(s)= 30		
<input type="button" value="Console Tableau"/>					<input type="button" value="Supprimer"/>		

Un tableau qui contient une liste de personnes, identifiées par leur nom, prénom, points.

Chaque ligne est identifiée par un numéro.

Chaque ligne contient une case à cocher, permettant de la sélectionner.

Une espace résumant le contenu du tableau: nombre de lignes, total de points

Des boutons de contrôle:

Ajouter: insérer les données saisies dans les champs.

Supprimer: supprimer les ligne sélectionnées

Console Tableau: afficher le contenu du tableau dans la console.



Plan

Ce TP consiste en 6 étapes qui visent à satisfaire les besoins exprimés :

- **Etape 1:** Tableau Statique
- **Etape 2:** Insertion d'une nouvelle ligne
- **Etape 3:** Insertion à partir d'un tableau d'objets
- **Etape 4:** Espace "Résumé"
- **Etape 5:** Insertion via des champs de saisie
- **Etape 6:** Suppression des lignes sélectionnées
- **Etape 7:** Observation du Flux Réseau




Etape 1

Tableau Statique



Etape 1 Tableau Statique

Dans un premier temps, on va construire le tableau suivant de manière statique, histoire de fixer la structure et le style CSS.



N°	Nom	Prénom	Points	Select
1	nom1	prenom1	5	<input type="checkbox"/>
2	nom2	prenom2	10	<input type="checkbox"/>
3	nom3	prenom3	15	<input type="checkbox"/>



Etape 1 Tableau Statique

À vous

1

Dans le fichier etape1/index.html, construire le tableau suivant:

N°	Nom	Prénom	Points	Select
1	Nom1	Prénom1	5	<input type="checkbox"/>
2	Nom2	Prénom2	5	<input type="checkbox"/>
3	Nom3	Prénom3	5	<input type="checkbox"/>

!

Aide:

Pour insérer une checkbox dans la dernière colonne, utilisez l'élément suivant:

```
<input type="checkbox"/>
```

Etape 1 Tableau Statique

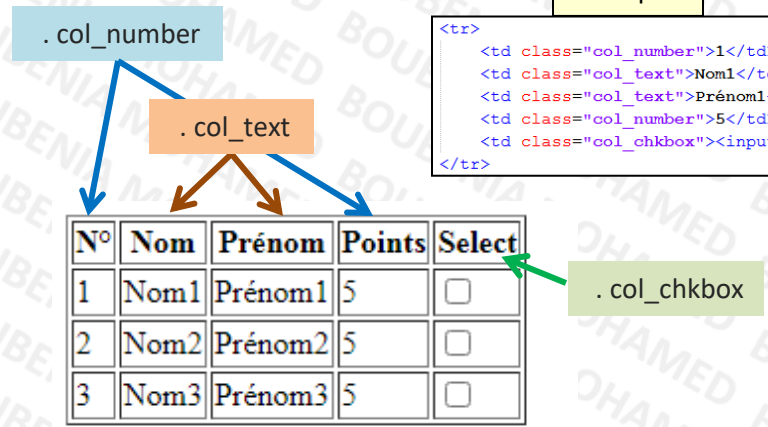
À vous

2 Dans un fichier style.css, rajoutez les règles suivantes, afin de mettre en forme notre tableau.

1 Attribuez aux colonnes les classe suivantes

Exemple:

```
<tr>
  <td class="col_number">1</td>
  <td class="col_text">Nom1</td>
  <td class="col_text">Prénom1</td>
  <td class="col_number">5</td>
  <td class="col_chkbox"><input type="checkbox"/></td>
</tr>
```



N°	Nom		Prénom	Points	Select
1	Nom1		Prénom1	5	<input type="checkbox"/>
2	Nom2		Prénom2	5	<input type="checkbox"/>
3	Nom3		Prénom3	5	<input type="checkbox"/>



Etape 1 Tableau Statique

À vous

1

Maintenant, on souhaite mettre notre tableau à l'intérieur d'un conteneur "div", auquel on donnera l'identifiant (table_container)

2

Puis, on voudrait utiliser ce conteneur pour :

- mettre le tableau au milieu.
- mettre une police d'écriture globale Calibri comme première police.

N°	Nom	Prénom	Points	Select
1	Nom1	Prénom1	5	<input type="checkbox"/>
2	Nom2	Prénom2	5	<input type="checkbox"/>
3	Nom3	Prénom3	5	<input type="checkbox"/>



Etape 2

Début de dynamisme/Interaction

Insertion d'une nouvelle ligne



Etape 2 Insertion d'une nouvelle ligne

Besoin

1

On souhaite rajouter le comportement suivant:

N°	Nom	Prénom	Points	Select
1	Nom1	Prénom1	5	<input type="checkbox"/>
2	Nom2	Prénom2	5	<input type="checkbox"/>
3	Nom3	Prénom3	5	<input type="checkbox"/>
99	nom	prenom	10	<input type="checkbox"/>
99	nom	prenom	10	<input type="checkbox"/>

Ajouter

2

Au clic sur le
bouton "Ajouter"

Une nouvelle ligne est créée avec le données
statiques suivantes:
num="99"
nom="nom"
prenom="prenom"
points=10



Etape 2 Insertion d'une nouvelle ligne

À vous

1

Recopiez etape1/index.html dans le fichier etape2/index.html

2

Dans le conteneur (table_container), et après le tableau, insérez un élément bouton qui porte le texte "Ajouter".

N°	Nom	Prénom	Points	Select
1	Nom1	Prénom1	5	<input type="checkbox"/>
2	Nom2	Prénom2	5	<input type="checkbox"/>
3	Nom3	Prénom3	5	<input type="checkbox"/>
99	nom	prenom	10	<input type="checkbox"/>
99	nom	prenom	10	<input type="checkbox"/>

Ajouter

!

Aide:

Pour insérer un bouton, utilisez l'élément suivant:

```
<button>text</button>
```



Etape 2 Insertion d'une nouvelle ligne

À vous

1

Attachez au bouton "Ajouter" un événement "au clic", qui exécute la fonction:
doInsertRowTable(99, 'nom', 'prenom', 10)

2

Rajouter au document HTML, un fichier script externe nommé: script.js

3

Dans le fichier script.js, déclarez la fonction **doInsertRowTable**, qui prend en paramètre:
num, nom, prenom, points
Puis, mettre le code nécessaire partout où c'est marqué "A COMPLETER".

```
function doInsertRowTable(num, nom, prenom, points){

    // Récupérer l'élément tableau
    // const table = (A COMPLETER)

    // Créer un élément de type tr (ligne de tableau)
    // row = (A COMPLETER)

    // Affecter à l'élément row, la valeur "row" à son attribut "class",
    // en utilisant la méthode setAttribute
    // row.(A COMPLETER)

    // Créer 5 éléments de type td (colonne de tableau)
    // col1 = (A COMPLETER)
    // col2 = (A COMPLETER)
    // col3 = (A COMPLETER)
    // col4 = (A COMPLETER)
    // col5 = (A COMPLETER)

    /*
    Remplir le contenu de chaque colonne avec les paramètres
    de la fonction num, nom, prenom, points, en utilisant innerText
    exemple:
        col1.innerText = num;

    remarque:
        la dernière colonne doit contenir un élément input,
        dont l'attribut type vaut checkbox
    */
    // A compléter: REMLISSAGE DU CONTENU

    /*
    En utilisant la méthode setAttribute
    affecter les classes correspondantes pour chaque colonne
    exemple:
        col1.setAttribute("class", "col_number");
    */
    // A compléter: AFFECTATION DES CLASSES

    // A compléter: RAJOUTER LES COLONNES A LA LIGNE row avec la méthode append()

    // A compléter: RAJOUTER LA LIGNE row AU TABLEAU table
}
```



Etape 2 Insertion d'une nouvelle ligne

À vous

N°	Nom	Prénom	Points	Select
1	Nom1	Prénom1	5	<input type="checkbox"/>
2	Nom2	Prénom2	5	<input type="checkbox"/>
3	Nom3	Prénom3	5	<input type="checkbox"/>
99	nom	prenom	10	<input type="checkbox"/>
99	nom	prenom	10	<input type="checkbox"/>

1

Cliquez sur le bouton ajouter, et assurez-vous qu'une ligne est effectivement insérée à la fin avec les données spécifiées.

Ajouter

2

Maintenant, actualisez la page

?

Que remarquez-vous ? Et comment l'expliqueriez-vous ?



Etape 3

Insertion à partir d'un tableau d'objets



Etape 3 Insertion à partir d'un tableau d'objet

Besoin

1

Jusqu'à présent, notre tableau était initialisé de manière statique, directement dans le document HTML

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

Ajouter

2

Maintenant, on souhaite qu'il soit initialisé à partir du tableau "persons", suivant, qui contient des objets "person" identifié chacun par un nom, prenom et points.
Le tableau doit contenir exactement les même données décrite sur l'image.



Etape 3 Insertion à partir d'un tableau d'objet

À vous

1

Recopiez etape2/index.html dans un nouveau fichier etape3/index.html

2

Supprimez les lignes du tableau, mis à part l'entête.

N°	Nom	Prénom	Points	Select
Ajouter				



Etape 3 Insertion à partir d'un tableau d'objet

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ☐ Au début, JavaScript va créer le tableau "persons".
- ☐ Puis il va exécuter une fonction init().
- ☐ init() va parcourir le tableau "persons", et pour chaque objet, elle va appeler une fonction doInsert(nom, prenom, points), avec les données de l'objet courant.
- ☐ doInsert(nom, prenom, points) est une fonction qui va appeler la fonction doInsertRowTable (qu'on a définie précédemment, et qui se contente d'insérer une ligne dans l'élément Table).
- ☐ JS doit aussi tenir compte du nombre de lignes insérées, afin de remplir la colonne N° (1, puis 2, puis 3). Donc il faudra tenir un compteur qu'on nommera "lignes", et qu'on initialisera à 0.



Etape 3 Insertion à partir d'un tableau d'objet

À vous

☐ Au début, JavaScript va créer le tableau "persons".

Déclarez l'objet "persons"
dans notre fichier script.js

```
persons = [
  // ...
]
```



Etape 3 Insertion à partir d'un tableau d'objet

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

✓ **Au début, JavaScript va créer le tableau "persons".**

- ☐ Puis il va exécuter une fonction `init()`.
- ☐ `init()` va parcourir le tableau "persons", et pour chaque objet, elle va appeler une fonction `doInsert(nom, prenom, points)`, avec les données de l'objet courant.
- ☐ `doInsert(nom, prenom, points)` est une fonction qui va appeler la fonction `doInsertRowTable` (qu'on a définie précédemment, et qui se contente d'insérer une ligne dans l'élément `Table`).
- ☐ JS doit aussi tenir compte du nombre de lignes insérées, afin de remplir la colonne N° (1, puis 2, puis 3). Donc il faudra tenir un compteur qu'on nommera "lignes", et qu'on initialisera à 0.



Etape 3 Insertion à partir d'un tableau d'objet

À vous

- ☐ Puis il va exécuter une fonction `init()`.
- ☐ `init()` va parcourir le tableau "persons", et pour chaque objet, elle va appeler une fonction `doInsert(nom, prenom, points)`, avec les données de l'objet courant.

Complétez le code suivant

```
// Appel de init()
init();

function init(){
    // Utilisez la boucle for..of vue en cours
    // pour parcourir les objets du tableau persons
    // et appeller doInsert sur chaque objet

    // A completer
    /*
        for(... of ...){
            doInsert(...);
        }
    */
}
```



Etape 3 Insertion à partir d'un tableau d'objet

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ Au début, JavaScript va créer le tableau "persons".
- ✓ Puis il va exécuter une fonction `init()`.
- ✓ `init()` va parcourir le tableau "persons", et pour chaque objet, elle va appeler une fonction `doInsert(nom, prenom, points)`, avec les données de l'objet courant.
- ❑ `doInsert(nom, prenom, points)` est une fonction qui va appeler la fonction `doInsertRowTable` (qu'on a définie précédemment, et qui se contente d'insérer une ligne dans l'élément `Table`).
- ❑ JS doit aussi tenir compte du nombre de lignes insérées, afin de remplir la colonne N° (1, puis 2, puis 3). Donc il faudra tenir un compteur qu'on nommera "lignes", et qu'on initialisera à 0.



Etape 3 Insertion à partir d'un tableau d'objet

À vous

- ☐ doInsert(nom, prenom, points) est une fonction qui va appeler la fonction doInsertRowTable (qu'on a définie précédemment, et qui se contente d'insérer une ligne dans l'élément Table).

Complétez le code suivant

```
function doInsert(nom, prenom, points){
    // Appelez doInsertRowTable avec les paramètres appropriés
    // pour l'instant on mettra num = 99
    num = 99;
    // A Compléter
}
```



Etape 3 Insertion à partir d'un tableau d'objet

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ Au début, JavaScript va créer le tableau "persons".
- ✓ Puis il va exécuter une fonction `init()`.
- ✓ `init()` va parcourir le tableau "persons", et pour chaque objet, elle va appeler une fonction `doInsert(nom, prenom, points)`, avec les données de l'objet courant.
- ✓ `doInsert(nom, prenom, points)` est une fonction qui va appeler la fonction `doInsertRowTable` (qu'on a définie précédemment, et qui se contente d'insérer une ligne dans l'élément `Table`).
- ❑ JS doit aussi tenir compte du nombre de lignes insérées, afin de remplir la colonne N° (1, puis 2, puis 3). Donc il faudra tenir un compteur qu'on nommera "lignes", et qu'on initialisera à 0.



Etape 3 Insertion à partir d'un tableau d'objet

À vous

- ☐ JS doit aussi tenir compte du nombre de lignes insérées, afin de remplir la colonne N° (1, puis 2, puis 3). Donc il faudra tenir un compteur qu'on nommera "lignes", et qu'on initialisera à 0.

Complétez le code suivant

```
// Au début de script.js
// Déclarez une variable "lignes", et l'initialiser à 0
```

Réajustez la fonction doInsert, définie précédemment, pour qu'elle utilise le compteur lignes

```
function doInsert(nom, prenom, points){
    // Incrémenter la variables lignes de 1,
    // puis l'affecter à num, au lieu de 99.
    // A compléter

    /*code précédent*/
}
```



Etape 3 Insertion à partir d'un tableau d'objet

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ Au début, JavaScript va créer le tableau "persons".
- ✓ Puis il va exécuter une fonction `init()`.
- ✓ `init()` va parcourir le tableau "persons", et pour chaque objet, elle va appeler une fonction `doInsert(nom, prenom, points)`, avec les données de l'objet courant.
- ✓ `doInsert(nom, prenom, points)` est une fonction qui va appeler la fonction `doInsertRowTable` (qu'on a définie précédemment, et qui se contente d'insérer une ligne dans l'élément `Table`).
- ✓ JS doit aussi tenir compte du nombre de lignes insérées, afin de remplir la colonne N° (1, puis 2, puis 3). Donc il faudra tenir un compteur qu'on nommera "lignes", et qu'on initialisera à 0.



Etape 3 Insertion à partir d'un tableau d'objet

À vous

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

Ajouter

1

Ouvrez etape3/index.html dans le navigateur, et assurez-vous que le tableau contient bien les éléments insérés à partir du tableau "persons".

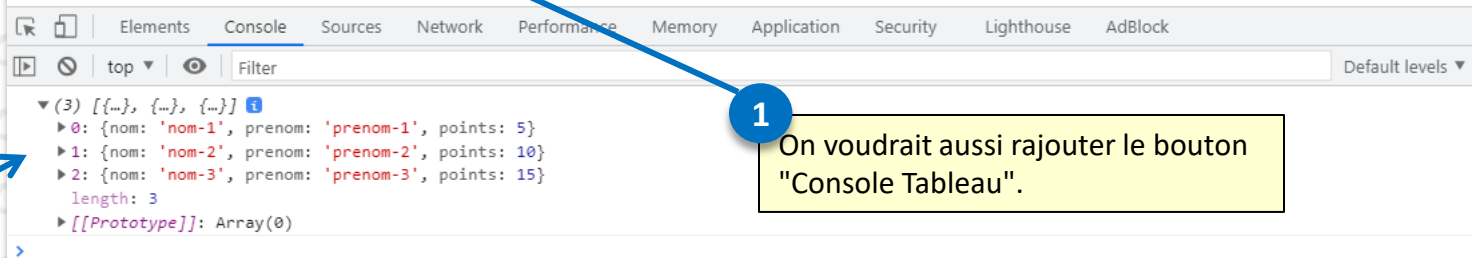


Etape 3 Insertion à partir d'un tableau d'objet

Un autre besoin

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

Ajouter Console Tableau



1

On voudrait aussi rajouter le bouton "Console Tableau".

2

Qui affichera le contenu du tableau "persons" dans la console.



Etape 3 Insertion à partir d'un tableau d'objet

À vous

1

Attachez au bouton "Tableau Console" un événement click, qui exécutera la fonction "ConsoleTableau()"

2

Dans script.js, complétez le code nécessaire pour le fonctionnement de la fonction consoleTableau()

```
function consoleTableau() {  
    // A compléter  
}
```



Etape 4

Espace "Résumé"



Etape 4 Espace "Résumé"

Besoin

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>
3 ligne(s)			Total point(s)= 30	
Ajouter		Console Tableau		

1

Au-dessous du tableau, on souhaite ajouter un espace qui résume son contenu: nombre de ligne, et le total des points.



Etape 4 Espace "Résumé"

À vous

1

Recopiez etape3/index.html dans un nouveau fichier etape4/index.html

2

Juste au-dessous du tableau, insérez le div suivant

```
<div id="container_summary">
  <p id="p1">0 ligne(s)</p>
  <p id="p2"></p>
  <p id="p3">Totale point(s)= 0</p>
</div>
```

3

Voyez apparaitre l'espace résumé

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

0 ligne(s)

Totale point(s)= 0

Ajouter

Console Tableau



Etape 4 Espace "Résumé"

À vous

1

On va rajouter un peu de CSS pour mettre en forme notre espace de résumé

2

Dans le fichier style.css, rajoutez les règles nécessaires

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>
0 ligne(s)			Totale point(s)= 0	
Ajouter	Console Tableau			

3

L'espace est maintenant mis en forme



Etape 4 Espace "Résumé"

Besoin

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>
0 ligne(s)			Totale point(s)= 0	
Ajouter	Console Tableau			

2

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ Pour le nombre de lignes, nous avons déjà la variable "lignes", qu'on a définie précédemment.
- ❑ Pour le total de points, on va utiliser une nouvelle variable "total_points", qu'on va déclarer et initialiser à 0.
- ❑ A chaque insertion dans `doInsert()`, on va mettre à jour les deux variables (lignes est déjà mise à jour, il reste à faire celle de "total_points").
- ❑ `doInsert()` appellera une fonction `update_summary()`.
- ❑ La fonction `update_summary()` va mettre à jour l'espace du résumé.

1

Maintenant, il faudra rajouter le code nécessaire pour que l'espace "résumé" reflète le contenu du tableau (nombre de ligne, total points).
Par exemple ici, on devrait avoir: "3 ligne(s)" et "Total point(s)= 30"



Etape 4 Espace "Résumé"

À vous

- ☐ Pour le total de points, on va utiliser une nouvelle variable "total_points", qu'on va déclarer et initialiser à 0.

Complétez le code suivant

```
// Au début de script.js  
// Déclarez une variable "total_points", et l'initialiser à 0
```



Etape 4 Espace "Résumé"

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ Pour le nombre de lignes, nous avons déjà la variable "lignes", qu'on a définie précédemment.
- ✓ Pour le total de points, on va utiliser une nouvelle variable "total_points", qu'on va déclarer et initialiser à 0.
- ☐ A chaque insertion dans `doInsert()`, on va mettre à jour les deux variables (lignes est déjà mise à jour, il reste à faire celle de "total_points").
- ☐ `doInsert()` appellera une fonction `update_summary()`.
- ☐ La fonction `update_summary()` va mettre à jour l'espace du résumé.



Etape 4 Espace "Résumé"

À vous

- ☐ A chaque insertion dans `doInsert()`, on va mettre à jour les deux variables (lignes est déjà mise à jour, il reste à faire celle de `"total_points"`).

Réajustez la fonction `doInsert`, définie précédemment, pour qu'elle additionne les points à `"total_points"`

```
function doInsert(nom, prenom, points){
    // Rajouter points (passé en paramètre)
    // à la variable total_points
    // A compléter

    /*code précédent*/
}
```



Etape 4 Espace "Résumé"

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ Pour le nombre de lignes, nous avons déjà la variable "lignes", qu'on a définie précédemment.
 - ✓ Pour le total de points, on va utiliser une nouvelle variable "total_points", qu'on va déclarer et initialiser à 0.
 - ✓ A chaque insertion dans `doInsert()`, on va mettre à jour les deux variables (lignes est déjà mise à jour, il reste à faire celle de "total_points").
- ☐ `doInsert()` appellera une fonction `update_summary()`.
 - ☐ La fonction `update_summary()` va mettre à jour l'espace du résumé.



Etape 4 Espace "Résumé"

À vous

☐ doInsert() appellera une fonction update_summary().

Complétez le code suivant

```
function doInsert(nom, prenom, points){

    /*code précédent*/

    // Appelez la fonction update_summary()
    // qu'on définira après
    // A compléter

}
```



Etape 4 Espace "Résumé"

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ Pour le nombre de lignes, nous avons déjà la variables "lignes", qu'on a définie précédemment.
- ✓ Pour le total de points, on va utiliser une nouvelle variable "total_points", qu'on va déclarer et initialiser à 0.
- ✓ A chaque insertion dans `doInsert()`, on va mettre à jour les deux variables (lignes est déjà mise à jour, il reste à faire celle de "total_points").
- ✓ `doInsert()` appellera une fonction `update_summary()`.
- ☐ La fonction `update_summary()` va mettre à jour l'espace du résumé.



Etape 4 Espace "Résumé"

À vous

☐ La fonction `update_summary()` va mettre à jour l'espace du résumé.

Complétez le code suivant

```
function update_summary(){
    // Récupérer l'élément id = p1
    // element_lignes = ...(A COMPLETER)

    // Récupérer l'élément id = p3
    // element_points = ...(A COMPLETER)

    // Avec innerText, modifiez le contenu de element_lignes
    // pour afficher le nombre de lignes (variables lignes)
    // element_lignes...(A COMPLETER)

    // Avec innerText, modifiez le contenu de element_points
    // pour afficher le total des points (variables total_points)
    // element_points...(A COMPLETER)
}
```



Etape 4 Espace "Résumé"

Besoin

Voici la logique de fonctionnement qu'on va implémenter:


- ✓ Pour le nombre de lignes, nous avons déjà la variable "lignes", qu'on a définie précédemment.
- ✓ Pour le total de points, on va utiliser une nouvelle variable "total_points", qu'on va déclarer et initialiser à 0.
- ✓ A chaque insertion dans `doInsert()`, on va mettre à jour les deux variables (lignes est déjà mise à jour, il reste à faire celle de "total_points").
- ✓ `doInsert()` appellera une fonction `update_summary()`.
- ✓ La fonction `update_summary()` va mettre à jour l'espace du résumé.



Etape 4 Espace "Résumé"

À vous

Finalement, dans le bouton "Ajouter", changer l'événement de sorte qu'il appelle la fonction "doInsert('nom', 'prenom', 10)"



```
<button onclick="doInsert('nom', 'prenom', 10)">
  Ajouter
</button>
```



Etape 4 Espace "Résumé"

À vous

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>
3 ligne(s)			Total point(s)= 30	
Ajouter		Console Tableau		

1

Ouvrez etape4/index.html dans le navigateur, et assurez-vous que l'espace résumé reflète le contenu du tableau.



Etape 5

Insertion via des champs de saisie



Etape 5 Insertion via des champs de saisie

Besoin

1

Jusqu'à présent, le bouton ajouter ne faisait que rajouter des données statiques spécifiées dans le code JS (voir le onclick du bouton Ajouter)
`doInsertRowTable(99, 'nom', 'prenom', 10)`

2

Mais maintenant, on voudrait que ces données soient saisies par l'utilisateur, via des champs de saisie (un formulaire)

Nom:

Prénom:

Points:

Ajouter

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

3 ligne(s)

Total point(s)= 30

Console Tableau



Etape 5 Insertion via des champs de saisie

À vous

1

Recopiez etape4/index.html dans un nouveau fichier etape5/index.html

2

Dans "form_container", et Juste au-dessus du tableau, insérer le div suivant

4

Résultat

3

Veillez à supprimer l'élément button qui existe déjà en bas

```
<div id="form_container">
  <div class="form_group">
    <label class="form_label">Nom:</label>
    <input type="text" id="form_nom" class="form_field"/>
  </div>
  <div class="form_group">
    <label class="form_label">Prénom:</label>
    <input type="text" id="form_prenom" class="form_field"/>
  </div>
  <div class="form_group">
    <label class="form_label">Points:</label>
    <input type="number" id="form_points" class="form_field"/>
  </div>
  <button onclick="doInsertRowTable(99, 'nom', 'prenom', 10)">
    Ajouter
  </button>
</div>
```

Nom:

Prénom:

Points:

Ajouter

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

3 ligne(s) Total point(s)= 30

Console Tableau

```
<div id="container_summary">
  <p id="p1">0 ligne(s)</p>
  <p id="p2"></p>
  <p id="p3">Totale point(s)= 0</p>
</div>
<del>
  <button onclick="doInsertRowTable(99, 'nom', 'prenom', 10)">
    Ajouter
  </button>
  <button onclick="consoleTableau()">
    Console Tableau
  </button>
</del>
```



Etape 5 Insertion via des champs de saisie

À vous

1

On va rajouter un peu de CSS pour mettre en forme notre formulaire

2

Dans le fichier style.css, rajouter les règles nécessaire

Nom: Prénom: Points:

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

3 ligne(s) Total point(s)= 30

3

Le formulaire est maintenant mis en forme

4

Maintenant, il faudra rajouter le code nécessaire pour que le bouton Ajouter insère les données saisies.



Etape 5 Insertion via des champs de saisie

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ☐ L'événement onclick du bouton Ajouter appellera désormais une fonction doNewData().
- ☐ La fonction doNewData() va:
 - ☐ lire les valeurs des champs de saisie,
 - ☐ puis passer ces valeurs à la fonction doInsert.
 - ☐ Ensuite, elle insère aussi ces nouvelles valeurs dans le tableau "persons".
 - ☐ Finalement, elle videra les champs pour préparer une autre saisie.
- ☐ Pour une saisie contrôlée, doNewData() vérifiera aussi qu'aucun champs de saisie n'est vide.



Etape 5 Insertion via des champs de saisie

À vous

- ☐ L'événement onclick du bouton Ajouter appellera désormais une fonction doNewData().

Apportez les modifications nécessaires pour que l'événement clic sur Ajouter appelle la fonction "doNewData()"

```
<button onclick="doInsertRowTable(99, 'nom', 'prenom', 10)">
    Ajouter
</button>
```



Etape 5 Insertion via des champs de saisie

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

✓ **L'événement onclick du bouton Ajouter appellera désormais une fonction doNewData().**

☐ La fonction doNewData() va:

- ☐ lire les valeurs des champs de saisie,
- ☐ puis passer ces valeurs à la fonction doInsert.
- ☐ Ensuite, elle insère aussi ces nouvelles valeurs dans le tableau "persons".
- ☐ Finalement, elle videra les champs pour préparer une autre saisie.

☐ Pour une saisie contrôlée, doNewData() vérifiera aussi qu'aucun champs de saisie n'est vide.



Etape 5 Insertion via des champs de saisie

À vous

- ☐ La fonction doNewData() va:
 - ☐ lire les valeurs des champs de saisie,
 - ☐ puis passer ces valeurs à la fonction doInsert.
 - ☐ Ensuite, elle insère aussi ces nouvelles valeurs dans le tableau "persons".
 - ☐ Finalement, elle videra les champs pour préparer une autre saisie.

Complétez le code suivant

```
function doNewData() {
    // Récupérer l'élément id = form_nom
    // const elt_nom = ...(A COMPLETER)

    // Récupérer l'élément id = form_prenom
    // const elt_prenom = ...(A COMPLETER)

    // Récupérer l'élément id = form_points
    // const elt_points = ...(A COMPLETER)

    // Accéder à l'attribut value de elt_nom et le stocker dans "nom"
    // nom = ...(A COMPLETER)

    // Accéder à l'attribut value de elt_prenom et le stocker dans "prenom"
    // prenom = ...(A COMPLETER)

    // Accéder à l'attribut value de elt_points et le stocker dans "points"
    // points = parseInt(...A COMPLETER)
    // ATTENTION: il faut pas oublier parseInt, qui sert à convertir un texte en entier

    // Appelez doInsert avec les nouvelles données récupérées
    // doInsert(...A COMPLETER)

    // Insérer également les nouvelles données dans le tableau "persons"
    // exemple d'insertion d'un objet dans le tableau "persons":
    //     persons.push({"BOB", "BINGO", 16});
    // persons...(A COMPLETER)

    // En manipulant les éléments elt_nom, form_prenom, form_points
    // Remettez leur contenu à vide (attribut value)
    // elt_nom...(A COMPLETER)
    // form_prenom...(A COMPLETER)
    // form_points...(A COMPLETER)
}
```



Etape 5 Insertion via des champs de saisie

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ L'événement onclick du bouton Ajouter appellera désormais une fonction `doNewData()`.
- ✓ La fonction `doNewData()` va:
 - ✓ lire les valeurs des champs de saisie,
 - ✓ puis passer ces valeurs à la fonction `doInsert`.
 - ✓ Ensuite, elle insère aussi ces nouvelles valeurs dans le tableau "persons".
 - ✓ Finalement, elle videra les champs pour préparer une autre saisie.
- ❑ Pour une saisie contrôlée, `doNewData()` vérifiera aussi qu'aucun champs de saisie n'est vide.



Etape 5 Insertion via des champs de saisie

À vous

- ☐ Pour une saisie contrôlée, `doNewData()` vérifiera aussi qu'aucun champs de saisie n'est vide.

Apportez les modifications nécessaires pour que `doNewData()` vérifie que les données saisie ne sont pas vides.

```
function doNewData() {
    /*
    if(nom est une chaine vide
    ou prenom est une chaine vide
    ou points est vide)
        Aide:(pour points, utilisez (Number.isNaN(points))
            qui retourne true si l'entier est vide, false sinon))
    {
        // corps du if
        utilisez la méthode alert de window pour afficher le message
        "Formulaire incomplet !"
    } else{
        // corps du else
        Faire le traitement normal d'ajout
    }
    */
}
```



Etape 5 Insertion via des champs de saisie

Besoin

Voici la logique de fonctionnement qu'on va implémenter:

- ✓ L'événement onclick du bouton Ajouter appellera désormais une fonction `doNewData()`.
- ✓ La fonction `doNewData()` va:
 - ✓ lire les valeurs des champs de saisie,
 - ✓ puis passer ces valeurs à la fonction `doInsert`.
 - ✓ Ensuite, elle insère aussi ces nouvelles valeurs dans le tableau "persons".
 - ✓ Finalement, elle videra les champs pour préparer une autre saisie.
- ✓ Pour une saisie contrôlée, `doNewData()` vérifiera aussi qu'aucun champs de saisie n'est vide.



Etape 5 Insertion via des champs de saisie

À vous

Testez la nouvelle
fonctionnalité d'ajout via
des champs de saisie

1

Nom: Prénom: Points:

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>
4	nom-4	prenom-4	20	<input type="checkbox"/>

4 ligne(s) Total point(s)= 50

3

Appuyez sur Console Tableau pour
vous assurez que les données ont
également été insérées dans le
tableau "persons"

2

Assurez-vous que l'espace
résumé est toujours
cohérent



Etape 6

Suppression des lignes sélectionnées

Etape 6 Suppression des lignes sélectionnées

Besoin

1 On souhaite rajouter une dernière fonctionnalité, comme suit:

Nom:

Prénom:

Points:

Ajouter

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input checked="" type="checkbox"/>
3	nom-3	prenom-3	15	<input checked="" type="checkbox"/>

3 ligne(s)

Total point(s)= 30

Console Tableau

Supprimer

3 Les lignes sélectionnées seront supprimées du tableau

2 Lorsque l'utilisateur clique sur le bouton supprimer



Etape 6 Suppression des lignes sélectionnées

À vous

1

Recopiez etape5/index.html dans un nouveau fichier etape6/index.html

Nom: Prénom: Points:

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input checked="" type="checkbox"/>
3	nom-3	prenom-3	15	<input checked="" type="checkbox"/>

3 ligne(s) Total point(s)= 30

2

Insérez le bouton Supprimer après le bouton "Console Tableau"

3

Attachez au bouton supprimer l'événement au click, qui exécutera la fonction `deleteRow()`, qu'on définira après



Etape 6 Suppression des lignes sélectionnées

À vous

1

On définit la logique de fonctionnement de la fonction `deleteRow()`, comme suit :

2

Complétez le code

```
function deleteRow() {

    // Étape 1: Vérifier si le tableau est vide
    // Si le nombre de lignes (variable "lignes") est égal à 0,
    // afficher un message d'alerte et sortir de la fonction.

    // Étape 2: Récupérer le tableau et la liste des checkbox cochées
    // Utiliser querySelector() et querySelectorAll()
    // (pense à cibler les checkbox à l'intérieur de .col_chkbox)

    // Étape 3: Vérifier si aucune checkbox n'est cochée
    // Si aucune sélection, afficher un message "Sélectionnez au moins une ligne !"
    // et sortir de la fonction.

    // Étape 4: Demander une confirmation à l'utilisateur avant suppression
    // si l'utilisateur clique sur "Annuler", quitter la fonction.

    // Étape 5: Récupérer toutes les lignes du tableau

    // Étape 6 : Parcourir les lignes
    // → Pour chaque ligne, vérifier si la checkbox est cochée.
    // Si oui :
    // - récupérer la valeur des points dans la 4e cellule
    // - la soustraire de total_points
    // - décrémenter la variable lignes
    // - supprimer la ligne du DOM
    // - supprimer aussi la même entrée dans le tableau persons

    // Étape 7: Afficher un message de succès et mettre à jour le résumé
    // Appeler update_summary() à la fin
}
```



Etape 6 Suppression des lignes sélectionnées

À vous

Nom:

Prénom:

Points:

Ajouter

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input checked="" type="checkbox"/>
3	nom-3	prenom-3	15	<input checked="" type="checkbox"/>

3 ligne(s)

Total point(s)= 30

Console TableauSupprimer

2

Appuyez sur Console Tableau pour vous assurez que les données ont également été supprimées du tableau "persons"

1

Exécuter le bouton supprimer pour s'assurer que les lignes sont bien supprimées

3

Maintenant, actualisez la page

?

Que remarquez-vous ? Et comment l'expliqueriez-vous ?



Etape 6 Suppression des lignes sélectionnées

À vous

1

On va rajouter un peu de CSS pour mettre en forme nos boutons

4

Dans le fichier style.css, rajoutée les règles nécessaire

2

Affecté les classes suivantes aux boutons:
"btn-add" pour Ajouter
"btn-del" pour Supprimer
"btn-consol" pour Console Tableau

3

Mettez les boutons Supprimer et Console Tableau à l'intérieur d'un div qui portera l'identifiant "container_btn"

5

Assurez-vous d'obtenir le rendu suivant

Nom:
 Prénom:
 Points:
Ajouter

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

3 ligne(s)
 Total point(s)= 30

Console Tableau
 Supprimer



Etape 7

Observation du Flux Réseau

Etape 7 Observation du Flux Réseau

À vous

1 Ouvrez etape6/index.html dans le navigateur

2 Puis ouvrez les outils de développement, et positionnez vous sur l'onglet "Network".

3 Actualisez la page Web. Vous devriez voir apparaître la liste des fichiers reçus par le navigateur

Name	Status	Type	Initiator	Size	Time
index2.html	Finis...	docu...	Other	2.1 kB	2 ms
style.css	Finis...	style...	index2.html	1.5 kB	9 ms
script.js	Finis...	script	index2.html	7.4 kB	9 ms

Form with fields: Nom: , Prénom: , Points: , and a button Ajouter.

N°	Nom	Prénom	Points	Select
1	nom-1	prenom-1	5	<input type="checkbox"/>
2	nom-2	prenom-2	10	<input type="checkbox"/>
3	nom-3	prenom-3	15	<input type="checkbox"/>

3 ligne(s)Total point(s)= 30

Console TableauSupprimer

4 Cliquez sur le fichier script.js, puis sur l'onglet "Preview". Vous devriez voir le contenu de votre fichier JavaScript, en intégralité.

Nameindex2.htmlstyle.cssscript.js

PreviewResponseInitiatorTiming

```
1
2 // Au début de script.js
3 // Déclarez une variable "lignes", et l'initialisez à 0;
4 lignes = 0;
5
6 // Au début de script.js
7 // Déclarez une variable "total_points", et l'initialisez à 0;
8 total_points=0;
9
10 persons = [
11   {
12     nom: "nom-1",
13     prenom: "prenom-1",
14     points: 5,
15     select: false
16   },
17   {
18     nom: "nom-2",
19     prenom: "prenom-2",
20     points: 10,
21     select: false
22   },
23   {
24     nom: "nom-3",
25     prenom: "prenom-3",
26     points: 15,
27     select: false
28   }
29 ]
```

? Vous en pensez quoi ?
Imaginez que le code contienne des instructions sensibles (comme des mots de passe)

JavaScript



TP 4: JAVASCRIPT

1. Manipulation du langage JavaScript
2. Manipulation du DOM
3. Utilisation du BOM
4. Un Site Statique Interactif