Temps de réalisation : 4 heures

Barème :

- Exercice 1 : 12 points
- Exercice 2 : 4 points
- Exercice 3 : 4 points

Critères d'évaluation :

- Maîtriser les bases de JavaScript.
- Savoir créer une page internet permettant de réserver une salle de réunion.

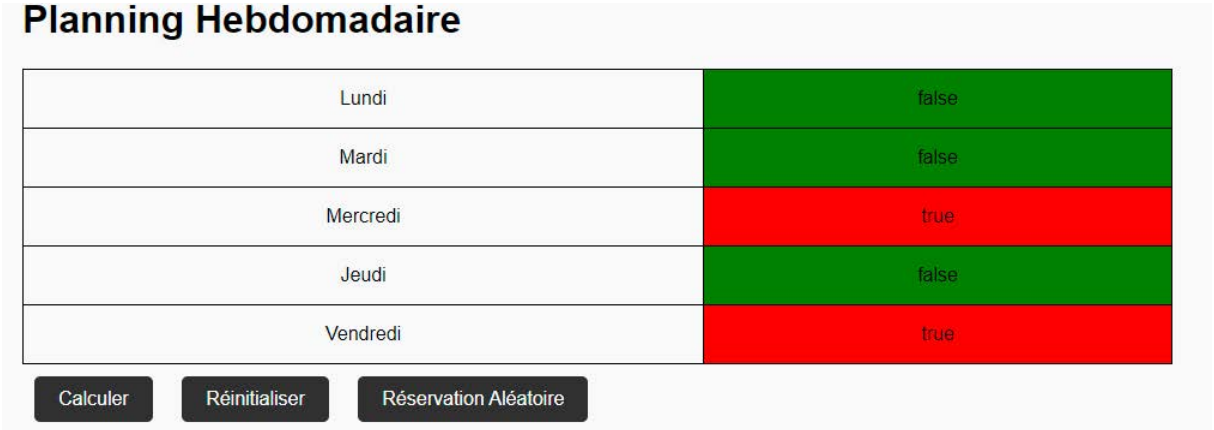
Consignes

L'objectif de cette évaluation est de créer un système de planning interactif en utilisant HTML, CSS et JavaScript.

Le planning permettra aux utilisateurs de visualiser et de gérer leurs réservations pour une salle, pour chaque jour, du lundi au vendredi.

Ce système affichera l'état actuel de réservation pour chaque jour de la semaine et offrira la possibilité de modifier facilement l'état de réservation (de « réservé » à « libre », et inversement) au clic sur une cellule de statut.

Dans cet exemple, après avoir cliqué sur « mercredi » et « vendredi », leur statut est passé à « réservé » (true). Un nouveau clic inversera le statut à « libre » (false), permettant un ajustement alternatif des réservations.



Exemple de planning hebdomadaire © Florian Rambeau

Exercice 1 : Création et interaction avec le planning

Création du fichier HTML

Créez un fichier index.html.

À l'intérieur des balises < body >, insérez une balise < table > avec l'identifiant planning.

Initialisation du tableau JavaScript

Créez un tableau JavaScript **planningJours**, composé d'un objet pour chaque jour de la semaine.

Chaque objet est composé de deux propriétés : jour, contenant le nom du jour, et statut, contenant le statut de réservation de la salle, qui peut être un booléen à true (réservé) ou false (libre). Le statut de réservation est initialisé à false pour tous les jours de la semaine.

Ce tableau sera la source principale d'informations pour connaître l'état de réservation d'une salle, il est donc nécessaire de le mettre à jour lors de changements et de se servir ensuite de ses données pour appliquer les changements esthétiques.

Génération du tableau HTML

Au chargement de la page, utilisez une boucle JavaScript pour parcourir **planningJours** et peupler le tableau HTML dynamiquement.

Avec un tableau structuré ainsi, l'utilisation de la boucle for…of… semble la plus adaptée.

Pour chaque jour, créez une ligne contenant deux cellules : une pour le nom du jour et l'autre pour le statut de réservation. Initialisez les cellules de statut à false avec un fond vert, indiquant que les salles sont libres.

Interaction utilisateur

Attachez un gestionnaire d'événements 'click' à chaque cellule de statut.

Lors d'un clic, inversez le statut de réservation du jour correspondant dans le tableau **planningJours**, qui sert de référentiel. Ensuite, ajustez l'affichage de la cellule pour refléter le nouveau statut basé sur les informations de **planningJours**. La couleur de fond doit correspondre au statut actualisé : rouge pour « réservé » (true), vert pour « libre » (false). Les modifications esthétiques des cellules doivent donc refléter fidèlement l'état du tableau **planningJours**.

MÉTHODES JS UTILES

- **document.getElementById(id) :** cette méthode permet de récupérer un élément du DOM correspondant à l'identifiant unique fourni(id), où id doit correspondre à l'attribut id d'un élément HTML.
- **.appendChild(node) :** cette méthode ajoute un nœud (généralement créé préalablement avec document.createElement) comme dernier enfant d'un nœud parent.
- **document.getElementsByTagName(tagName) :** renvoie une collection en direct (HTMLCollection) de tous les éléments enfants qui correspondent au nom de balise donné.
- **.textContent :** cette propriété permet de récupérer ou de définir le contenu textuel d'un nœud. Contrairement à innerHTML, il traite le texte littéralement, sans interpréter les balises HTML ou les entités.
- **.style.backgroundColor :** permet de modifier la couleur de fond d'un élément. La valeur assignée doit être une couleur valide en CSS (nom de couleur, code hexadécimal, rgba, etc.).

Exercice 2 : Gestion des créneaux

Ajoutez un bouton **« Calculer »** sous le tableau HTML. Après avoir cliqué, comptez et montrez le nombre de créneaux libres et occupés via une alerte.

Ajoutez un bouton **« Réinitialiser »** sous le tableau HTML permettant de réinitialiser tous les statuts de réservation des salles à false dans le **planningJours**, puis répercutez la mise à jour dans le tableau HTML.

PROPRIÉTÉ UTILE

.length : fournit la longueur d'une chaîne de caractères ou le nombre d'éléments dans un tableau.

Exercice 3 : Réservation aléatoire

Ajoutez un bouton **« Réservation Aléatoire »**, qui, au clic, invite l'utilisateur à saisir le nombre de jours à réserver via un prompt, puis réserve aléatoirement ce nombre de jours dans la semaine.

Avant cette action, réinitialisez tous les statuts à false pour garantir que la réservation part d'un état vierge.

MÉTHODE UTILE

.indexOf() : recherche un élément dans un tableau ou une sous-chaîne dans un texte, renvoyant l'indice de sa première occurrence ou –1 si non trouvé.

.Math.random() : méthode statique renvoie un nombre pseudo-aléatoire à virgule flottante supérieur ou égal à 0 et inférieur à 1. Voir documentation