

Programmation Web avancée

Cahier de laboratoire

Année académique 2018-2019

Contenu

Exercices de laboratoire	2
Exercice 1 : vérification de formulaire.....	2
Exercice 2 : Ajax.....	3
Exercice 3 : Ajax requête générique.....	4
Exercice 4 : Modification de l'arbre DOM	5
Exercice 5 : « drag and drop »	6
Exercice 6 : Stockage d'information sur le client	7
Exercice 7 : Sélection d'entrées dans une table html	8
Projet de laboratoire : Implémentation d'un utilitaire web de gestion des tâches.....	9
Consignes générales.....	9
Consignes.....	10

Exercices de laboratoire

Exercice 1 : vérification de formulaire

Ecrire un formulaire permettant de se créer un compte sur un site. Le formulaire comprend les champs suivants :

- Nom : uniquement des caractères
- Prénom : uniquement des caractères
- Date de naissance : JJ/MM/AAAA
- Adresse email : les adresses mail doivent normalement respecter un format décrit dans une RFC. <https://tools.ietf.org/html/rfc3696>

Vous trouverez dans le lien ci-dessous un exemple d'expression régulière permettant de vérifier une adresse mail qui aurait été saisie. Dans le cadre de l'exercice, vous pouvez écrire également votre propre fonction qui vérifiera la validité d'une adresse, éventuellement selon un format simplifié.

<http://stackoverflow.com/questions/46155/validate-email-address-in-javascript>

- Numéro de téléphone : +suivi de 11 chiffres
- Mot de passe souhaité : minimum 5 caractères quelconques.

On vous demande :

- a. Ecrire le code JavaScript permettant de tester que les données fournies par l'utilisateur du formulaire respectent bien le format attendu.
- b. Ecrire une fonction permettant de calculer un hash du mot de passe au moment où le formulaire est soumis.

Documentation additionnelle : <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript/tp-un-formulaire-interactif-1>

Exercice 2 : Ajax

L'exercice 2 consiste à charger une page de connexion à un site. En repartant du formulaire de l'exercice 1, il s'agit de vérifier sur le serveur avec Ajax que l'adresse mail n'est pas déjà utilisée.

L'exercice est à réaliser

- avec et sans JQuery ;
- En simulant un temps d'attente côté serveur pour obtenir la réponse. L'affichage d'un indicateur côté client de votre choix (image sablier ou autre) sera affiché le temps de l'attente.

Documentation : <https://openclassrooms.com/courses/ajax-et-l-echange-de-donnees-en-javascript/l-objet-xmlhttprequest-1>

Exercice 3 : Ajax requête générique

Pour cet exercice, il faut réaliser une interface permettant d'interroger une base de donnée quelconque.

L'interface proposera 5 champs :

- Serveur,
- Utilisateur,
- Mot de passe,
- Base de données,
- Requête (exclusivement des SELECT – à vérifier)

et un bouton pour envoyer la requête en utilisant AJAX.

L'interface affichera l'ensemble des tuples retournés par la requête sous la forme d'un tableau.

Exercice 4 : Modification de l'arbre DOM

1. Ecrire le code html d'une page contenant un bouton et un paragraphe et permettant d'insérer la date dans le contenu du paragraphe lors d'un clic sur le bouton.
2. Ecrire une page html et le code JS se rapportant à la problématique suivante :
 - a. La page html contient une liste de catégories d'articles.
 - b. La sélection d'une catégorie provoque le chargement et l'affichage de sous catégories associées. Dit autrement, une partie de l'arbre DOM de votre page initiale est modifiée et les sous-catégories sont donc insérées dans l'arbre.
 - c. Un choix opéré dans la sous-catégorie provoque l'affichage d'images d'articles relatifs à cette sous-catégorie.

Exemple :

- Catégorie : Jardinage
- Sous catégories : outils, tondeuse, produits, ...
- Dans outils, on retrouve Bêche, fourche, brouette, ...

Documentation : <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript/manipuler-le-code-html-partie-1-2>

Exercice 5 : « drag and drop »

L'exercice consiste à développer un jeu de classification de formes destiné à des enfants.

Principe du jeu : Pour faire simple, il consiste à ranger via la technique du « drag & drop » des formes (des ronds, des carrés ou des triangles) dans un panier présentant une des formes précitées.

Au départ, le jeu comporte 10 formes à ranger.

Comptage du nombre de points acquis :

- 1 point est acquis si la forme est placée dans le panier adéquat
- 1 point est perdu si la forme n'est pas placée dans le panier adéquat.

Fin du jeu : Le jeu se termine lorsque toutes les formes ont été rangées.

Documentation : <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript/le-drag-drop-2>

Exercice 6 : Stockage d'information sur le client

Cet exercice s'inscrit dans la continuité du précédent. Il s'agit d'implémenter :

- le stockage de l'état du jeu à chaque action réalisée (chaque forme mis dans un des paniers – sauvegarde en permanence) .
- lorsqu'une partie est terminée, le stockage d'informations générales concernant la partie achevée, c.-à-d. la date, le score obtenu et le nom du joueur.

Pour cet exercice, vous utiliserez la technique du localStorage.

Documentation : https://www.w3schools.com/html/html5_webstorage.asp

Exercice 7 : Sélection d'entrées dans une table html

L'exercice consiste à pouvoir sélectionner plusieurs entrées dans un tableau en cliquant et en maintenant enfoncé la souris lors du survol sur d'autres cases (voir l'exemple en rouge ci-dessous).

	19/03/2018	20/03/2018	21/03/2018	22/03/2018	23/03/2018	24/03/2018	25/03/2018
1							
2							
3							
4							
1							

Projet de laboratoire : Implémentation d'un utilitaire web de gestion des tâches.

Consignes générales

Objectif :

Implémentation en JS d'un outil de réservation de locaux au sein de pages HTML.

Consignes :

Un dossier (une archive contenant l'ensemble de votre projet avec des schémas et explications additionnelles que vous jugez nécessaire pour faciliter la compréhension de votre travail) à envoyer selon les modalités suivantes :

Comment ? Par mail à l'adresse de votre titulaire de laboratoire (@hepl.be).

Pour quand ? Le vendredi 30 mai 2019 au plus tard.

Quoi ? L'archive porte le nom GroupeNomPrenom_JSProjet20182019.rar ou zip

Exemple : 210xDupontJean_JSProjet20182019.rar

Evaluation : votre travail sera évalué **oralement** en session. Veuillez-vous munir de l'énoncé et d'une page de garde pour l'évaluation.

La présentation et le look de votre site est laissé à votre propre appréciation pour vous permettre un maximum de créativité. Seuls les points imposés en JS sont à respecter impérativement. En cas de doute, n'hésitez pas à demander...

Consignes

L'objectif du dossier est d'implémenter un système de gestion de tâche. Celui-ci doit permettre de classer les tâches en 3 catégories :

- A faire,
- En cours et
- Terminé.

Une tâche reprend les informations suivantes :

- Nom
- Description
- Attribution de la tâche (permis les utilisateurs inscrits sur le site)
- Priorité (avertissement si choix d'un tâche de moindre priorité)
- Etat d'avancement de la tâche
- Attribution à un projet

Lorsqu'une tâche se trouve dans l'état « à faire » elle peut être éditée totalement sauf pour l'état d'avancement qui n'est accessible que dans « en cours ».

Dans l'état « en cours », on pourra exclusivement faire évoluer son état de progression.

Finalement lorsqu'une tâche est marquée comme « terminée », on ne peut plus modifier celle-ci.

L'accès à l'ensemble des fonctionnalités du site nécessite l'identification via login et mot de passe.

L'interface doit permettre de choisir le projet pour lequel on affiche l'ensemble des tâches. L'utilisateur doit pouvoir créer un nouveau projet et de nouvelles tâches qui devront être attribuées à un projet.

Lorsque l'on choisi une tâche pour la passer en « en cours », il faut vérifier que l'on respecte les priorités fixées sur les tâches. Si ce n'est pas le cas, un message avertira l'utilisateur qui devra expressément choisir de maintenir ou non son choix.

L'ensemble des fonctionnalités ne sont accessibles qu'après connexion. Il existe 2 types de compte ayant des droits différents :

- Soit on est simple utilisateur (sur un projet) et ne peut voir que les tâches qui nous sont assignées ;
- Soit on est un gestionnaire de projet (celui qui crée le projet), dans ce cas on peut voir toutes les tâches du projet.

Tout le monde peut créer un projet dont il devient automatiquement gestionnaire.

L'interface utilisateur doit être pensée de manière à faciliter l'utilisation de celle-ci selon la situation. Il est impératif d'utiliser les possibilités offert par JS pour rendre l'expérience utilisateur la plus agréable possible. L'interface doit être mono-page et utiliser JS pour les éventuelles fenêtres complémentaires.