

TP de Javascript

TP2 : Ajout et suppression d'éléments dans la page web avec Javascript

Équipe pédagogique

Mars 2018

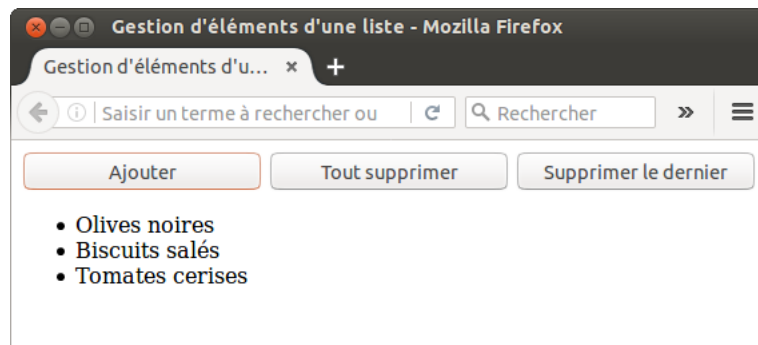
Pour rendre les pages accessibles en ligne, il faudra les copier sur le serveur **Forge** dans le dossier **forge_html**. La page **index.html** se trouvera à l'adresse **http://forge.info.univ-angers.fr/~login/index.html**. Toutes les pages, une fois testées localement (c'est-à-dire sur votre ordinateur), devront être validées puis copiées sur **Forge**.

1 Gestion d'une liste

Question 1.1 Reproduire la figure ci-dessous, sans la liste. Lorsqu'on clique sur Ajouter, une fenêtre pop-up s'ouvre et demande la saisie d'un élément à ajouter à la liste. Si l'on ne saisit rien, ou que l'on clique sur Annuler, aucun élément n'est ajouté à la liste. Initialement, la liste n'existe pas et doit être créée dans `<div id="ici"></div>`.

Question 1.2 Le bouton Tout supprimer doit supprimer la liste, si elle existe. On utilisera les fonctions `hasChildNodes()` et `removeChild()`.

Question 1.3 Le bouton Supprimer le dernier doit supprimer le dernier élément de la liste, s'il existe. On utilisera la méthode `lastChild`.



2 Suite de Fibonacci

Le but de cet exercice est de calculer les termes de la suite de Fibonacci, $(u_n)_{n \in \mathbb{N}}$, et de construire un histogramme des $n + 1$ premiers termes. La suite de Fibonacci est définie par la formule de récurrence suivante : $u_n = u_{n-1} + u_{n-2}$, pour tout $n \geq 2$. Les termes u_0 et u_1 sont donnés, de sorte que l'on peut calculer tous les autres termes à l'aide de la formule de récurrence.

Question 2.1 Reproduire la section **1 - Données** de la figure et faire en sorte qu'une fenêtre pop-up affiche « données invalides » quand on clique sur Calculer si n n'est pas un entier dans l'intervalle $[0, 20]$, ou si u_0 ou u_1 ne sont pas des nombres. On s'assurera également que les valeurs par défaut des données sont celles de la figure.

Question 2.2 Reproduire la section **2 - Résultats**, et écrire la fonction Javascript associée au bouton Calculer de manière à ce qu'elle calcule u_n avec la formule de récurrence, pour un entier n donné, ainsi que s_n , qui est la

somme des termes de la suite : $s_n = u_0 + u_1 + \dots + u_n$. La fenêtre des valeurs de la suite de Fibonacci affichera tous les termes calculés, qui seront stockés dans un tableau. Tous les résultats seront en lecture seule.

Question 2.3 Reproduire la section **3 - Histogramme** afin qu'un histogramme comme celui de la figure soit construit pour représenter graphiquement les termes de la suite. Cet histogramme sera construit dans un `<div id="ici"></div>`, il contiendra une grande table constituée d'une seule ligne, et de $n+1$ colonnes. Chaque colonne de la grande table correspond à un terme de la suite, et est constituée d'une petite table. Chacune de ces petites tables est constituée d'une seule colonne, et de trois lignes.

- La première ligne est la valeur du terme. Cette valeur doit être centrée, et affichée avec la police **Arial**.
- La seconde ligne est la barre de l'histogramme, sa hauteur doit être égale à la valeur du terme correspondant, en pixels, et sa largeur est de 40 pixels. La couleur de fond est **aquamarine**.
- La troisième ligne est le nom du terme (comme « u_0 »), elle doit être centrée.

On créera une classe en CSS pour chaque ligne de la petite table, et on affectera, en Javascript, la classe correspondante en construisant la table.

Question 2.4 Faire en sorte qu'à chaque appui sur le bouton Calculer, tout diagramme préexistant dans le `div` soit préalablement détruit avant qu'un nouveau diagramme ne soit construit. En cas de saisie de données invalides, tous les résultats doivent être effacés, et le diagramme détruit (s'il existe).

