

JavaScript - Listes II

Note : consulter le cours (js/manipuler le DOM) et la doc @ MDN !

Exo 1 – Parents/enfants

En HTML, créer une liste vide (OL), un input[type=text], un bouton d'action «**ajouter**», un bouton d'action «**racine**».

EN JS, au click sur «ajouter», insérer un élément de liste (LI) contenant la valeur de INPUT à la racine de UL. Associer également un input[type=radio] à chaque élément créé.

Dès lors, le bouton ajouter se comportera différemment si un élément radio est sélectionné. Le nouveau LI, la valeur de INPUT et RADIO seront insérés comme enfants du LI actif. (Attention, le code HTML doit rester valide: OL > LI > OL || UL > LI).

Au click sur racine, si un RADIO est actif, on le désactive pour pouvoir insérer à la racine.

Si le champs texte est vide, indiquer une erreur à l'utilisateur et empêcher l'insertion.

Exo 2 – Glisser/Déposer

En HTML, créer 2 listes : une vide et une autre remplie de LI contenant du texte.

En JS, écrire un programme permettant de glisser les LI d'une liste et de le déposer dans l'autre.

Attention : on ne peut déposer un LI uniquement dans un UL.

Variante : on pourra ou non choisir la position de l'insertion dans la liste.

Indices :

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Drag_and_drop

http://www.html5rocks.com/en/tutorials/dnd/basics/?redirect_from_locale=fr

Exo 3 – Filtrer

En HTML, créer un UL contenant et un input[type=text] d'action «**filtrer**».

UL correspond à une collection sur un thème au choix (sport, bd, fooding, cocktails, etc.).

Ajouter n LI correspondant aux éléments de votre collection. Deux options:

- Easy: remplir la liste *en dur* en HTML avec des LI.
- Badass: parcourir un array d'objets JS pour peupler la liste de LI.

À minima, chaque LI aura cette structure:

```
< li id="id" class="class" data-category="category" > label < / li >
```

Exemple:

En JS, à la saisie dans INPUT, utiliser l'event pour effectuer un tri et ne conserver que les LI dont le label et/ou la catégorie correspond(ent) à la valeur courante de INPUT.

Pour cela, utiliser [Array.prototype.filter](#).

Indice: utiliser les id des objet/LI filtrés pour repeupler la liste.

Aller plus loin: ajouter des catégories et sous-catégories, customiser l'UI avec des icônes et images, ajouter d'autre méthodes de filtres (checkbox, select/option etc.)