TP Frontend
EBM 2017/2018 - S3

Objectif : Mettre en œuvre les bonnes pratiques de développement javascript (fonctions anonymes, arguments passés comme des objets JSON…). Découvrir les requêtes asynchrones en Ajax, m

Exercice 1 : Rappels de javascript = librairie utils.js

- Améliorer la fonction debug pour éliminer les variables globales à l'aide d'une fermeture

Exercice 2 : Fonctions de rappel, JSON = librairie boucle.js

- Déclencher un traitement périodique boucle(iPeriode,fCbTraitement,fCbContinuer) :
  - qui effectue un appel à une fonction callback de traitement
  - puis un appel à une fonction gérant la poursuite de boucle
  - NB : On n'utilisera pas les arguments supplémentaires de window.setTimeout
- Transformer votre code en une librairie de fonctions
  - Comment rendre le 3ème argument optionnel ?
  - Passer les arguments en JSON
    - On définira une fonction enrichir(oDefaut,oConfig) qui renvoie un nouvel objet en ayant complété les propriétés absentes de l'objet de configuration par les valeurs de l'objet par défaut.
  - Permettre plusieurs traitements périodiques en parallèle à l'aide d'une closure

Exercice 3 : Cas d'étude TryIT, fonctions anonymes

- Refaire TryIT Editor (S1) à l'aide de boucle:
  - Un champ textarea permet d'éditer du code HTML
  - Un div placé à côté montre le rendu suite à un clic
    - Ou à intervalles réguliers (fonctionnalité débrayable)
  - Un appui sur la touche Esc annule la mise à jour automatique

- Utiliser html, val, show, hide, boucle
  - Comment ne pas avoir à nommer la fonction de poursuite lors de l'appel de la fonction boucle ?

Exercice 4 : Objet XMLHttpRequest : refactoring de code = librairie ajax.js

- Améliorer les fonctions envoiRequete et traiteReponse pour que la fonction de callback choisie soit destinataire de la réponse du serveur comme paramètre, lorsque celle-ci est disponible
- L'intégrer dans une librairie de fonctions
  - Quelle restriction entraîne votre solution technique ?
- Définir un handler 'ajax' pour la fonction envoiRequete
  - Il admettra un paramètre sous forme de JSON
    - type, callback, data seront facultatifs, url devra être fourni
  - Les données seront passées par json également, par exemple {"debutNom":"T"} au lieu de «debutNom=T». Vous utiliserez la structure for … in

Exercice 5 : Cas d'étude Web2 Suggest

- Un formulaire permet de saisir le début du nom de famille d'un étudiant dans un champ input
  - Les données sont stockées dans un fichier
  - Au fur et à mesure de la saisie, une liste de propositions s'affiche
  - On peut cliquer sur chaque proposition pour remplir le champ automatiquement
- IMPORTANT : l'expérience utilisateur !
  - Afficher un gif animé lors du déclenchement d'une requête asynchrone
  - Différer le réponse du serveur en utilisant sleep

Exercice 6 : Améliorations du Web2 Suggest

- Améliorer le Web2Suggest pour consommer un flux de données encodé en JSON avec plusieurs types de propriétés : tableau d'étudiants, avec leur formation, lus depuis un fichier
- Les données sont lues depuis une base de données avec les librairies php du TP backend (SQLSelect et ParcoursRs)
  - Utiliser json_encode pour transformer un tableau associatif en objet JSON
  - Comment ajouter des champs supplémentaires à un résultat mySQL pour préparer l'objet JSON ?
- On ne remplit pas forcément le champ avec ce qui est suggéré : notion de cache
  - Au fur et à mesure, le cache se remplit... éviter de faire des requêtes inutiles !