

Le langage Javascript et D.O.M.

Introduction

- Une archive contenant les fichiers mentionnés dans cette fiche est à récupérer sur la liste de diffusion.
- Vous trouverez une référence pour HTML DOM à cette adresse : <http://www.w3schools.com/jsref/>, et pour XML DOM http://www.w3schools.com/dom/dom_nodetype.asp
- Nous nous assisterons de deux outils de **Firefox** accessibles sous le menu *Outils* :
 1. **la console d'erreur**, (**ctrl+J**) : permet de lire les messages d'erreurs et d'avertissement. ⚠ Lorsque une erreur survient en Javascript, aucune information n'est portée à la connaissance de l'utilisateur, le programme s'arrête simplement sans notification. Pour connaître la nature de l'anomalie, il faut consulter la console d'erreurs. La console d'erreur est commune à toutes les instances du navigateur, pour une meilleure lisibilité, il convient de l'effacer puis de recharger la page à tester,
 2. **l'ardoise Javascript**, (**maj+F4**) : permet de saisir du code et de l'exécuter (**ctrl+r**) dans le contexte de la page chargée (dans l'onglet actif de la fenêtre active). Cette ardoise vous permettra de faire quelques expérimentations sans avoir à écrire le code à tester dans le fichier. **Attention** : tout ce qui a été fait dans l'ardoise est perdu lors du rechargement de la page. Veillez donc à copier-coller vos résultats dans un fichier pour les consigner au fur et à mesure.

Exercice 1 – Faisons connaissance avec l'environnement

Question 1.1 : Ouvrez une nouvelle fenêtre de navigateur, puis ouvrez l'ardoise Javascript (**maj+F4**).

- Tapez le code suivant dans l'ardoise et exécutez le (**ctrl+r**) :

```
1 function coucou() {  
2     alert('Coucou !');  
3 }  
4  
5 coucou();
```

Que se passe-t-il ?

- Effacez le contenu de l'ardoise et exécutez le code suivant :

```
1 coucou();
```

Que se passe-t-il ?

- Exécutez le code suivant :

```
1 alert(coucou);
```

Quel est le résultat de l'affichage ?

- Sans fermer l'ardoise Javascript, ouvrez une nouvelle fenêtre de navigateur, retournez dans l'ardoise Javascript et ré-exécutez les deux dernières instructions. Que se passe-t-il ?
- Sans fermer l'ardoise Javascript, fermez la fenêtre que vous venez d'ouvrir et retournez dans la fenêtre originale et exécutez à nouveau les deux instructions.
- Rechargez la page complètement dans le navigateur (**F5**), puis exécutez à nouveau les deux instructions.

Expliquez ce que toutes ces observations mettent en évidence ?

Exercice 2 – Programmation Javascript pure.

Récupérez le contenu du répertoire `jslab`, et ouvrez le fichier `jslab.html`. celui-ci contient seulement un squelette de document avec quelques éléments pré-définis pour nos expériences. Pour toutes les questions de cet exercice, nous travaillerons dans l'ardoise JavaScript (maj+F4).

Question 2.1 : Écrivez une fonction `printObject()` qui affiche (dans la div de sortie) la structure d'un objet passé en paramètre avec par exemple un affichage de cette forme :

```
Object [nom objet] : [type objet] =  
  attribut1 : type1 = valeur1  
  attribut2 : type2 = valeur2  
  ...  
  attributn : typen = valeurn
```

Note : vous aurez besoin d'utiliser un peu l'API DOM pour écrire à l'intérieur de la div de sortie.

Appliquez la fonction sur les objets prédéfinis `document`, `window` et sur une fonction.

Question 2.2 : En utilisant la notation JSON, créez et initialisez (avec les valeurs de votre choix) un objet contenant : une chaîne de caractères `nom`, un entier `age`, un booléen `majeur`, un objet `adresse` contenant un entier `numéro` et une chaîne `rue`, et enfin une fonction `travail` qui renvoie (toujours) `false`.

Affichez cet objet avec votre fonction `printObject()`.

Question 2.3 : Nous allons étudier le comportement des prototypes :

- Écrivez un constructeur `rect(lon, lar)` qui initialise un objet avec une longueur et une largeur.
- Appelez `printObject()` sur ce constructeur puis sur `rect.prototype`.
- Instanciez un objet `r` et affichez le.
- Ajoutez un attribut `color` avec la valeur `blue` au prototype de `rect`.
- Instanciez un objet `r2` et affichez le.
- Afficher à nouveau `r`.

Question 2.4 : Écrivez un script qui imprime l'arbre du document dans la div de sortie. On utilisera une simple indentation pour la profondeur et on utilisera la propriété `nodeName` pour imprimer le nom des nœuds. (Pensez à la récursivité)

Question 2.5 : Écrivez une instruction qui change l'étiquette du bouton1 en "Clear!", puis affectez lui une fonction anonyme qui effacera la div de sortie.

Question 2.6 : Écrivez une instruction qui change le style (en fait la classe) des modules (éléments de classe `module`) en `module_bis`. Notez que tous ces modules sont des `div` !

Exercice 3 – Premiers pas avec DOM

Question 3.1 : Récupérez le contenu du répertoire `bart`, modifiez le fichier `bart.html` afin que lorsque l'utilisateur clique sur le bouton, la phrase qu'il a saisie soit copiée 10 fois sur le tableau. Pour ce premier essai, nous n'allons pas créer de nouveaux nœuds dans le document nous même. Nous utiliserons la propriété `innerHTML` des éléments.

Question 3.2 : Cette méthode est-elle robuste, testez des injections HTML (ex : `blablaboldblabla`) et Javascript (`blabla<script type="text/javascript">alert('Owned !... or not');</script>blabla` puis `<p onmouseover="alert('coucou')"> Gotcha ! !</p>`)

Question 3.3 : Remplacez la propriété `innerHTML` par la propriété `textContent` et testez à nouveau vos injections.

Exercice 4 – DOM Only : un sommaire automatique

Question 4.1 : Récupérez le contenu du répertoire `sommaire`. Ajoutez un script au fichier `sommaire.html` de telle sorte qu'un sommaire cliquable soit automatiquement généré au chargement du fichier. Pour simplifier,

le fichier CSS fourni prend en charge la numérotation automatiquement, il suffit simplement d'ajouter des div de classes respectives `somH2`, `somH3` et `somH4`. On notera le nommage judicieux des classes. Pour cet exercice, on s'interdit l'emploi des attributs `textContent`, `innerHTML`, etc.. Les fonctions `document.createTextNode` et `document.createElement` devront être utilisées à la place. ATTENTION de bien ajouter le sommaire au document un fois le calcul terminé afin d'éviter que le sommaire ne s'indexe lui-même (récursion infinie)!

Question 4.2 : Ajoutez un titre de niveau 2 intitulé "Introduction" devant le premier paragraphe et constatez que le sommaire se met à jour.

Exercice 5 – Validation de formulaire

Question 5.1 : Récupérez le contenu du répertoire `validation` et ajoutez un script au fichier `validation.html` afin que le formulaire ne soit envoyé que si les conditions suivantes sont respectées :

- Le nom et le prénom doivent chacun comporter au moins 3 caractères ;
- L'utilisateur doit avoir choisi un des deux formats ;
- L'adresse email doit être valide. Pour faire simple elle doit être de la forme au moins un mot (une ou plusieurs lettres) suivie du signe arobase puis un mot, un point et un mot. Pensez aux expressions régulières (regexp) de javascript.

Lorsqu'un champs sera en erreur, on lui appliquera le style `error` (voir le css) afin de le mettre en évidence. On lui appliquera le style `none` s'il est valide (en cas de correction).

Question 5.2 : Ajoutez au formulaire l'image `subscribe.jpg` et ajoutez un script qui, lorsque le curseur de la souris survole l'image provoquera l'envoi du formulaire validé.