

JavaScript

...les bases

Par Richard Vigneux

Table des matières

- Introduction
- ADD-ON pour Firefox
- JavaScript et HTML
- JavaScript version 1.5
- Syntaxe de base
- Objets du navigateur
- Compatibilité
- Exercice du jour

2

Introduction

- Qu'est-ce que JavaScript?
- Quelle est la syntaxe de JavaScript?
- Quelle est la différence entre Java et JavaScript?
- Pourquoi utiliser des scripts avec HTML?
- En résumé
- Références
 - ◆ <http://www.ecma-international.org/>
 - ◆ ECMA : <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - ◆ Mozilla : <http://www.mozilla.org/js/language/E262-3.pdf>

3

ADD-ON pour Firefox

Le navigateur Firefox vous offre la possibilité de lui ajouter des applications complémentaires.

- Valideur de code HTML
 - ◆ <https://addons.mozilla.org/fr/firefox/addon/249>
- Modificateur d'en-tête HTTP
 - ◆ <https://addons.mozilla.org/fr/firefox/addon/3829>
- Débugueur JavaScript
 - ◆ Firebug: <https://addons.mozilla.org/fr/firefox/addon/1843>
 - ◆ Tuto : <http://www.digitalmediaminute.com/screencast/firebug-js/>
 - ◆ venkman : <https://addons.mozilla.org/firefox/addon/216>

De base, pour voir les erreurs de votre code dans Firefox, utilisez la console (Menu: Outils > Console d'erreurs)

4

JavaScript et HTML

Insertion de code JavaScript dans le HTML

- Comme pour les CSS, on doit inclure une balise <meta > pour indiquer le type de script utilisé:
 - ◆ `<meta http-equiv="Content-Script-Type" content="text/javascript">`
- Pour ajouter des scripts aux html on peut utiliser quatre approches:
 - ◆ Feuille de script externe:
 - ◆ `<script type="text/javascript" src="fichier.js"></script>`

Attention, on ne peut ajouter du code entre les balises.
 - ◆ Bloc de script interne:
 - ◆ `<script type="text/javascript">...votre code JavaScript...</script>`

Mettre le code dans un commentaire HTML <!-- /-->.
 - ◆ Par l'entremise d'un attribut événementiel
 - ◆ `<p onmouseover="codeJavaScript;">Texte du paragraphe</p>`
 - ◆ Par l'entremise de l'attribut « href »:
 - ◆ ``

5

JavaScript version 1.5

JavaScript existe en plusieurs versions, mais la version 1.5 est la mieux supportée par les navigateurs en général

- Références
 - ◆ W3 : <http://www.w3.org/TR/html4/interact/scripts.html>
 - ◆ Mozilla: <http://developer.mozilla.org/fr/docs/JavaScript>
 - ◆ Intro complète: https://developer.mozilla.org/fr/Guide_JavaScript_1.5
 - ◆ Base: https://developer.mozilla.org/fr/Référence_de_JavaScript_1.5_Core
 - ◆ Microsoft IE: <http://msdn.microsoft.com/en-us/library/ms919026.aspx>
 - ◆ JScript : <http://www.host-web.com/iishelp/JScript/htm/JSoc.htm>
 - ◆ <http://msdn.microsoft.com/en-us/library/ms950396.aspx>
 - ◆ Ce qui est standard : <http://msdn2.microsoft.com/en-us/library/d33e43t3.aspx>
 - ◆ Compatibilité IE – Firefox
 - ◆ https://developer.mozilla.org/fr/Migration_d'applications_d'Internet_Explorer_vers_Mozilla

6

Syntaxe de base (1/2)

Voici les éléments qui distinguent JavaScript du « C »

- Il n'existe pas de pointeurs n'y de classe.
- Variables:
 - ◆ Non typée
 - ◆ Déclaration: `var variable = valeur;`
- Tableaux:
 - ◆ Il n'existe que des vecteurs.
 - ◆ Déclaration Explicite: `var vecteur = new Array (nbElem);`
 - ◆ Déclaration Init: `var vecteur = new Array (elem1, elem2, elem3);`
 - ◆ Déclaration Implicite: `var vecteur = [elem1, elem2, elem3];`
 - ◆ Pour faire des tableaux, on fait des vecteurs de vecteur.
 - ◆ Déclaration: `var tableau = new Array (new Array (), new Array ());`
 - ◆ Utilisation: `var contenu00 = tableau[0][0];`

Syntaxe de base (2/2)

- Concaténation de chaîne de caractères:
 - ◆ "bon" + "jour" → "bonjour"
 - ◆ "5" + "5" → "55" (5 + 5 → 10 → avec les chiffres, + fait son travail normal)
- Objet:
 - ◆ Même si le mot clé « class » n'existe pas, il est possible de simuler la création de classes.
 - ◆ fonction nomClasse (paramPourConstructeur, param1)


```

          {
            /* PROPRIÉTÉS */
            this.nomPropriete = valeur;
            this.param      = param1;
            /* MÉTHODES */
            this.NomMethode = NomClasse_NomMethode;
            /* CONSTRUCTEUR */
            this.NomDuConstructeur();
          }
          
```

Utilisation
`var objet = new nomClasse (param);`
`objet.NomMethode(param);`

Attention: Il n'y a pas de parenthèses lorsqu'on fait référence à une méthode externe. Utilisez une fonction générique pour définir directement à l'intérieur.
`function(param) { code... }`

Objets du navigateur (1/3)

La première chose à savoir lorsqu'on manipule une page Web avec du JavaScript est que l'objet de base est « window ».

- ◆ window
 - ◆ Représente l'onglet ou la fenêtre du navigateur.
 - ◆ Offre tout ce qu'il est possible de consulter ou de modifier.
 - ◆ Référence: <https://developer.mozilla.org/en/DOM/window>
- ◆ Sous-objets principaux:
 - ◆ document → le document HTML.
 - <https://developer.mozilla.org/en/DOM/document>
 - ◆ navigator → le navigateur.
 - <https://developer.mozilla.org/en/DOM/window.navigator>
 - ◆ screen → l'écran de l'ordinateur.
 - <https://developer.mozilla.org/en/DOM/window.screen>

9

Objets du navigateur (2/3)

- ◆ Objets principaux:
 - ◆ defaultStatus → message par défaut dans la barre de statut.
 - ◆ history → donne accès à l'historique.
 - ◆ location → donne accès à la barre d'adresse.
 - ◆ status → donne accès à la barre de statut.
 - ◆ scroll... → donne accès à la barre de défilement.
- ◆ Propriétés principales:
 - ◆ innerHeight → hauteur disponible pour le document HTML.
 - ◆ innerWidth → largeur disponible pour le document HTML.

10

Objets du navigateur (3/3)

- Méthodes principales:
 - ◆ alert → Affiche une boîte de dialogue (utile pour déboguer)
 - ◆ confirm → Affiche une boîte de dialogue (demande réponse)
 - ◆ prompt → Affiche une boîte de dialogue (demande un texte)
 - ◆ setInterval → permet d'exécuter du code à intervalle régulier.
 - ◆ setTimeout → permet d'exécuter un code après un délai.
 - ◆ escape → Encode les caractères spéciaux
 - ◆ getComputedStyle → Styles réels appliqués à un élément
- window.document
 - ◆ Est le sous-objet le plus important.
 - ◆ Il donne accès au document HTML et ses composants.
 - ◆ Principale méthode: write → permet d'écrire dans le document. ¹¹

Compatibilité

- La compatibilité est l'une des choses les plus difficiles à atteindre lorsqu'on fait du JavaScript. La raison est que chaque navigateur a sa propre façon d'accéder aux éléments et de les structurer. De plus, il faut aussi tenir compte des différences entre les versions.
- Pour ces raisons, dans le cadre du cours, nous nous limiterons à la version du navigateur Firefox disponible au Cégep. C'est cette version qui servira de référence lors des corrections de vos travaux.
- Bien entendu, dans la vraie vie, on va devoir rendre nos codes compatibles. Pour certains éléments, vous pouvez utiliser des bibliothèques existantes telles « JQuery ». Par contre, dans la majorité des cas, vous devrez assurer la compatibilité par vous-même. À ce moment, je vous invite à utiliser toutes les sources disponibles pour vous aider. Ci-dessous, un lien qui devrait être un bon point de départ:
 - ◆ https://developer.mozilla.org/en/DOM_Client_Object_Cross-Reference

12

Exercice du jour (1/2)

- L'exercice d'aujourd'hui est pour vous permettre de vous familiariser avec JavaScript et ses éléments de base.
- Pour pouvoir le réaliser, vous devrez, préalablement, faire une lecture rapide de la documentation sur JavaScript 1.5 pour repérer les éléments du langage qui vous seront nécessaires.
- Bien entendu, tout le code doit être impeccable.
- N'hésitez pas à vous créer des sous-fonctions pour rendre votre code plus structuré.
- Pour vos variables et procédures, utilisez des noms significatifs. Pour savoir si un nom de variable est significatif, demandez-vous ce qui suit:
 - ◆ Si quelqu'un d'autre lisait votre code, serait-il capable de déterminer le contenu possible d'une variable, seulement en lisant son nom? Si oui, le nom est significatif.
- Note: Vous être limité à 255 caractères pour vos noms.

13

Exercice du jour (2/2)

- Pour réaliser l'exercice, vous devez minimalement réaliser la fonction demandée. Cette fonction doit faire un « return » d'une « string » html. Je vous propose de vous créer une variable dans cette fonction qui contiendra tout ce que vous générerez.
 - ◆ Par exemple:


```
var var_HTML = "";

...
>" + "\n"
var_HTML += "<table>" + "\n"
           + " <thead >" + "\n"
           + ...

...

return var_HTML;
```

Attention, « \n » ne modifie que le code source. Pour modifier l'affichage à l'écran on doit utiliser

Assurez-vous que le code source généré est impeccable (indentation). Même si présentement on ne le voit pas, il est important de prendre de bonnes habitudes pour lorsqu'on générera le code avec PHP à la prochaine session.

14

