

Leçon 2.1 – Introduction à JavaScript

s1 -----

Dans cette partie, nous allons introduire le langage JavaScript qui permet de manipuler le document HTML : création ou repositionnement d'éléments de structure, modification du contenu et du style d'un élément, etc. Dans une première partie, nous verrons comment mettre en oeuvre ce langage ; vous ferez plusieurs exercices autour des notions centrales que sont les variables, les tableaux, les fonctions et les structures conditionnelles. JavaScript est vaste et je rappelle que ce module n'a pour objectif qu'une initiation à JavaScript. Je suppose cependant que vous avez déjà une petite expérience en matière de programmation et que la notion d'algorithme ne vous est pas inconnue.

s2 -----

JavaScript est un langage de programmation à l'origine développé chez Netscape Communications 1995 et standardisé par l'organisme ECMA en 1996 et 1997.

JavaScript est interprété par le navigateur. Qu'est-ce que ça veut dire ?

Nous avons vu que le fichier HTML transmis par le serveur contient à la fois des balises HTML définissant à la fois sa structure, son contenu et éventuellement son aspect et des informations de type CSS, contenues soit dans des balises script, soit dans des fichiers supplémentaires.

On va voir que ce fichier HTML va pouvoir contenir également des instructions JavaScript.

Le navigateur va traiter toutes ces informations dans un certain ordre et exécuter les instructions JavaScript qui on va le voir sont de nature à compléter et modifier la structure, le contenu et l'aspect de la page affichée.

s3 -----

JavaScript est extrêmement facile à apprendre si on se cantonne à de la programmation basique. Pour les développeurs avancés, il fonctionne également comme un langage orienté objet. Mais cet aspect ne sera pas abordé ici.

Comment fonctionne JavaScript ? À l'intérieur du langage HTML, il existe des mécanismes (on parle de méthodes et de propriétés) qui permettent de lire, modifier, mettre à jour et supprimer des éléments du document examiné. Cet ensemble de mécanismes est appelé DOM (pour Document Object Model).

JavaScript s'appuie sur cet ensemble de méthodes pour transformer le document HTML : repositionnement des éléments, création de nouveaux éléments, modification du contenu et du style d'un élément, suppression d'un élément, etc. On parle de manipulation statique quand toutes ces opérations sont effectuées par le client (c'est-à-dire par le navigateur) sans échange avec le serveur.

JavaScript peut également être un des éléments d'une architecture permettant un affichage dynamique de tout ou partie du contenu HTML, c'est-à-dire une architecture qui permet l'échange de données avec le serveur. AJAX (qui associe JavaScript, les CSS et le DOM) et PHP (langage qui s'exécute sur le serveur) sont les autres éléments de cette architecture. Cet aspect ne sera pas étudié ici.

s4 -----

Comment met-on en oeuvre JavaScript ?

Très simplement, de la même façon que pour CSS : soit au sein même de la page HTML à l'aide d'une balise `<script>` contenant le code à exécuter, soit en définissant un fichier extérieur contenant ce code à exécuter.

s5 -----

Le premier exemple ci-dessous que je vous demande de reproduire permet de changer le titre dans le conteneur identifié par la chaîne titre. Le code contenu entre les balises script effectue à peu près ceci : au niveau du document, va chercher l'élément dont l'identifiant a pour valeur titre et donne à sa propriété `innerHTML` (qui représente le contenu de cet élément) la valeur 'nouveau titre'.

Remarque : on devine que le navigateur ne peut exécuter ce code qu'après que le conteneur div ait été créé. Pour vous en convaincre, faites un essai en mettant l'élément div après le code. Nous verrons plus tard comment nous affranchir de cette contrainte.

s6 -----

Le deuxième exemple ci-dessous que je vous demande de reproduire permet de changer le titre dans le conteneur identifié par la chaîne titre, mais le code à exécuter se trouve dans un fichier extérieur `code_exercice2.js`

s7 -----

Vous trouverez sur le web énormément de tutoriels sur JavaScript. Plutôt que de reproduire ces tutoriels et se lancer dans une leçon exhaustive sur la syntaxe et les instructions JavaScript, je vous propose de découvrir un peu mieux JavaScript à travers la création d'un petit jeu : un quiz autour d'une liste de pays. Cependant, avant d'attaquer le développement de ce jeu, je vous propose quelques exercices préparatoires autour des notions centrales que sont les Variables, les Tableaux, les Fonctions et les Structures conditionnelles (de type `if else`).

La semaine prochaine nous découvrirons d'autres notions qui nous seront utiles pour développer ce jeu.

s8 -----

Les variables

Dans cet exercice que je vous demande de reproduire et de tester ligne après ligne, le code HTML est réduit à un conteneur vide portant l'identifiant "element".

Le code JavaScript effectue les opérations suivantes

- première ligne : on définit une variable appelée `variable1` ; cette variable est une chaîne de caractères reconnaissable aux guillemets : remarque 1 on pourrait se passer des trois petites lettres `var`. Nous verrons plus loin, quand nous étudierons les fonctions, les implications de ces trois lettres `var` ; remarque 2, cette variable n'est pas typée c'est-à-dire qu'on laisse JavaScript deviner que c'est une chaîne de caractères, ce qui peut parfois conduire à des erreurs ; remarque 3, on peut utiliser de simples quotes pour définir une chaîne de caractères ; remarque 4 : un nom de variable doit commencer par une lettre ou un `"_"` ; il peut comporter des lettres, des chiffres et les caractères `_` et `&` ; en revanche, les espaces ne sont pas autorisés ; de plus, il y a des noms interdits dont vous trouverez la liste sur la page

ci-dessous.

- deuxième ligne dont la signification est : dans le document, écrit (à la position de l'appel du script dans le document, c'est à dire ici à la suite de l'existant) la chaîne constituée de variable1 et une chaîne contenant la balise HTML
 qui signifie saut de ligne
- troisième ligne : on redéfinit la valeur de variable1 : à nouveau, on remarque que cette variable n'est pas typée : on laisse JavaScript deviner que c'est un nombre
- quatrième ligne dont la signification est déjà connue : au niveau du document, va chercher l'élément qui s'appelle "element" et donne à sa propriété innerHTML (qui représente le contenu de cet élément) le résultat du calcul variable1+6. Comme JavaScript a identifié variable1 comme un nombre, ce résultat est 10.

s9 -----

Les tableaux

Les tableaux sont une structure de données permettant de stocker des données en général de même nature, exemple, une liste d'auteurs, de marques, de villes ou, comme dans le projet qui clôt cette partie, une liste de pays.

Dans cet exercice que je vous demande de reproduire et de tester ligne après ligne, le code HTML est réduit à un conteneur vide portant l'identifiant "element".

Le code JavaScript effectue les opérations suivantes

- création de trois tableaux appelés tab1, tab2 et tab3 de trois façons différentes
- redéfinition du premier élément de tab3 tab3[0] prend la valeur "Amsterdam"
- définition d'un quatrième enregistrement pour tab3
- présentation de deux méthodes utiles : affichage du nombre d'enregistrements d'un tableau et affichage d'une chaîne de caractères concaténant l'ensemble des valeurs d'un tableau

s10 -----

Les fonctions

À quoi servent les fonctions. Généralement, une fonction rassemble une liste d'instructions que l'on souhaite réutiliser plusieurs fois dans un même développement.

Par exemple dans la première partie de l'exercice proposé, la fonction majA multiplie la valeur courante de la variable A par 3, lui ajoute 4 et met à jour le conteneur "element" avec cette nouvelle valeur de A. Remarque 1 : il est essentiel que la fonction soit déclarée avant d'être utilisée ; Remarque 2 : la fonction ne s'exécute que si elle est appelée ; Remarque 3 : on peut évidemment l'appeler plusieurs fois ; Remarque 4 : bien faire attention à la syntaxe : parenthèses, accolades, point virgule

Dans la deuxième partie de l'exercice proposé, on déclare une fonction possédant deux paramètres a et b. Que fait la fonction sommeAB ?

Elle effectue le calcul a + b et affiche le résultat dans la page. Remarque : si vous transmettez à la fonction sommeAB un paramètre a qui est un tableau, JavaScript ne saura pas le faire, à vous de bien faire attention ! Je vous conseille d'afficher la console des erreurs du navigateur pour qu'il affiche les éventuelles erreurs JavaScript

Autre exemple avec la fonction creeNouveauDiv

Cette fonction crée dans la page web un nouveau conteneur de type div et lui donne comme identifiant la chaîne de caractère passée en paramètres ;

Remarque 1 : le paramètre passé à la fonction `document.write` est une chaîne ; cette chaîne commence par la balise ouvrante `div` puis `id =`, puis des guillemets ; vient ensuite entre guillemets le nom donné à l'identifiant, ce nom qui vient justement d'être passé en paramètre ; remarquez donc bien la façon dont il faut échapper par des backslashes les caractères guillemets pour que JavaScript comprenne que ce guillemet fait partie de la chaîne à afficher et qu'il n'est pas le guillemet fermant la chaîne à afficher.

Dernier exemple avec une fonction qui calcule la somme des entiers jusqu'à `n`.

Le mot clé "return" permet de renvoyer le résultat du traitement ; dans le code JavaScript, l'appel à la fonction `sommeDesN` est en quelque sorte remplacé par la valeur calculée

Une remarque : cet exemple nous permet d'illustrer la notion de portée d'une variable : dans la fonction `sommeDesN` la variable `N` est définie par le mot-clé `var` ceci à pour conséquence qu'elle sera connue seulement à l'intérieur de la fonction ; ainsi la dernière ligne qui demande l'affichage de `N` conduit à une erreur ;

refaites l'exercice en enlevant le mot-clé `var` ! `N` devient connu hors de la fonction

revenons sur la variable `A` définie plus haut sans le mot-clé `var` - on dit que sa déclaration est implicite : sa portée est étendue à l'ensemble du code JavaScript, y compris n'importe quelle fonction

Il est important de bien connaître la portée de chaque variable. On peut ainsi souhaiter que certaines d'entre elles aient une portée globale à tout le code, ou locale, afin éviter les interférences qui ne manqueraient pas de se produire si deux variables portaient le même nom !

s11 -----

Structures conditionnelles

Dans cet exercice, nous allons voir d'une part comment tester si une condition est vraie et d'autre part comment mettre en place des boucles.

Commençons par les boucles.

Dans la première partie de l'exercice proposé, on définit un tableau d'auteurs

ensuite on écrit dans le document la première balise d'une liste à puces, la balise ``

ensuite, on fait une boucle qui parcourt tous les enregistrements du tableau depuis l'enregistrement d'indice 0 jusqu'au dernier qui est l'enregistrement d'indice 2, car la longueur du tableau étant 3, la dernière valeur est bien entendu `tab1[2]`

pour chaque enregistrement, on écrit dans le document une chaîne de caractères correspondant à une ligne de la liste à puces : cette chaîne commence par la balise ``, continue par l'enregistrement courant `tab1[i]` et finit par la balise fermante ``

enfin, il faut écrire la balise fermante ``

Vous pouvez tester le résultat. Remarque : dans l'exercice qui va suivre tout à l'heure vous aurez l'occasion de créer une liste déroulante à partir d'un tableau

Dans la seconde partie, on introduit une condition de type `if / else`

ainsi à l'intérieur de la boucle, c'est-à-dire pour chaque valeur du tableau, on effectue le test suivant : si la longueur de la chaîne de caractères `tab[i]` est `<=7` et si le premier caractère de la chaîne `tab[i]` est différent de la lettre "P" alors on affiche cette ligne

sinon, on affiche une ligne avec un simple tiret

Au passage, nous venons de découvrir deux éléments utiles concernant les chaînes de caractères :

l'attribut length qui donne leur longueur et la méthode charAt()

Vous trouvez facilement sur internet ou dans le lien ci-dessous comment tester une égalité et comment tester l'opérateur logique OU

s12 -----

Je vous propose à présent une leçon qui vous permettra de découvrir des outils associés aux principaux navigateurs et qui vont grandement vous faciliter les tâches de programmation. À tout de suite !