Langage JavaScript

Diapo 5

JavaScript est vraiment lié au code HTML d’une page web. La différence entre JavaScript et PHP, JavaScript est exécuté coté client, PHP, ASP et JSP, exécution du script coté serveur et renvoie la page au client.

Diapo 6

Cheminement d’une requête avec un codage PHP, ASP, JSP …

Diapo 7

JavaScript, c’est le navigateur (coté client) qui possède l’interpréteur et permet d’afficher la page demandée.

Diapo 8

Fonctionnement pas à pas pour une requête avec du JavaScript.

Construction d’un arbre DOM (Document Object Model). JavaScript peut utiliser l’arbre DOM, si on clique sur un bouton, JavaScript reprend la main et fait ce qu’il a à faire. L’interaction peut être locale ou sur le serveur.

Bouquin intéressant à lire (en bas de diapo).

Diapo 9

JS permet de faire des vérifications avant l’envoie au serveur, par exemple vérifier que l’adresse email est juste (présence @ et « .quelquechose »).

Permet vraiment d’alléger les requêtes vis-à-vis du serveur et augmenter la vitesse d’affichage vu que c’est exécuté localement.

Diapo 10

Il existe d’autres techno qui se font coté client comme flash, applet java …

API : application program interface. API webstorage (en JS) permet de travailler hors connexion sauvegarde locale et quand à nouveau du réseau se synchronise automatiquement, très utile pour les appareils mobiles n’ayant pas un accès réseau 100% du temps.

JavaScript coté serveur aussi, notamment avec les frameworks node.js et angular.js. Très facile de gérer la partie serveur avec ces frameworks (node.js). Angular.js permet de bien séparer les contenus client-serveur et gère le lien entre les deux.

Diapo 11

Outils pour JavaScript directement dans Firefox en mode développeur. Permet d’avoir accès à une console, au code HTML et CSS, au débogueur …

Diapo 12

Le débogueur est indispensable. Savoir bien l’utiliser permet de résoudre pas mal de soucis.

Diapo 14

Pour que mettre du code JavaScript on utilise les balises <script></script> directement dans un page HTML. Avec le code entre ces deux balises, ou alors juste un lien pour un document externe contenant le ou les scripts de la page.

Diapo 15

Les scripts peuvent être placé n’importe où sur la page. Souvent le script et placé dans l’entête de la page, mais certaine fois il peut être intéressant de placer le script juste avant la balise </body> car JS peut avoir besoin de jouer avec l’arbre DOM, donc c’est mieux s’il est entièrement chargé.

Attention, on ne peut pas faire de travail en parallèle si le script est en cours d’exécution.

Diapo 16

Le type d’une variable, n’est pas indispensable, notamment on peut facilement changer le type d’une variable. Cette possibilité est à la fois puissante, mais dangereuse. La variable peut avoir un booléen, puis int, puis char ….

Pour une chaine de caractère on l’entoure de «  », on peut aussi mettre des ‘ ‘. L’utilisation de ‘ ‘ permet de facilement afficher une chaine de caractère contenant des «  ».

Diapo 22

== et != autorise à comparer des opérandes de différents types.

Par contre l’utilisation de === et !== font des comparaisons que sur des éléments de même types.

Important toujours utiliser === et !== pour éviter les problèmes de règles de conversions

Diapo 23

Bien regarder pour voir particularité des fonctions en JS. Possibilité de plusieurs return dans la fonction. Un return peut être présent dans une boucle. On n’indique pas le type de la fonction. Bien regarder les petits détails différents de ce que l’on connait en C et en algo. Mettre un return dans une boucle c’est dépendant de la fonction, si gde fonction ça peut être dangereux si petite ça peut être très efficace !!

Comme en C une variable donnée à une fonction devient locale et donc disparait une fois la fonction finie. Une fonction peut être définie n’import ou dans le document.

Diapo 24

Une fonction est comme une variable donc on peut passer une fonction en tant que variable d’une autre fonction.

Diapo 25

Pareil bien voir l’intérêt d’utiliser une fonction en paramètre.

Diapo 26

Pas de surcharge de fonction (plusieurs fonctions avec le même nom). Car JS utilise la fonction définit en dernière pour effectuer l’ « action ». Si la dernière fonction ne prend qu’un seul paramètre et qu’on l’appel avec plusieurs elle va tout simplement ignorer les suivants.

Diapo 27 (sautée)

Diapo 28

Recap fonction

Diapo 30

Portée des variables (scope). Dans une fonction les variables déclarées sont locales et leurs durée de vie est dépendant du temps d’exécution de la fonction.

Diapo 31

Si on a une variable x globale et une variable x locale à la fonction f(), alors f() modifie le x global par la valeur du x local puis l’utilise.

Diapo 32

Une variable définie (sans la présence du var) pour la première fois dans une fonction devient une variable globale.

Diapo 35

Si le résultat d’un script n’est pas comme attendu, il est intéressant de jeter un œil à la console de mozilla (en plus du debogueur).

JS « remonte » que les initialisations, mais pas les affectations. Pour ça que dans le 3eme cas de la diapo on a deux undefined, car les déclarations sont remontées, mais le y = 1 reste ou il est.

Diapo 36

Exercice à faire sur cette diapo

Diapo39

JS n’est pas un langage réellement orienté objet, plutôt prototype

Canvas est un objet, il possède des dimensions et des fonctions d’opération (clear((), rect()…)

Diapo 40

Objet JS est équivalent à un struct en C. l’objet possède attribut. JSON : JavaScript Object Notation objet utilisable directement par JS.

Objet, ces attributs sont séparés par des « , » mais pas de « , » après le dernier. Le nom d’une propriété (attribut) doit être entre « » si c’est un nom système.

Diapo 41

Trois objets reliés par des attributs. L’objet peut être définie hors de l’objet auquel il est lié. Mais il est aussi possible de le définir directement dans l’objet « parents »

Diapo 42

Les références sont une sorte de pointeur. Une référence n’est pas un objet, c’est un nom pour accéder à un objet.

Diapo 43

Alert fait apparaitre un pop up avec un bouton ok pour fermer

Accès aux propriétés d’une ref pointant sur un objet deux possibilités

ref.propriété ou ref[« propriété »]

diapo 44  
méthode push permet d’ajouter une valeur à un tableau

ajout/supp d’une propriété directement dans le script

diapo 45

passage de paramètres de fonctions est un passage de valeurs (comme en C)

ici l’objet félicité est modifié car dans la fonction maigrir on utilise un pointeur vers l’objet félicité

diapo 46

diapo 47

this.poids cela veut dire que l’on utilise la var poids de l’objet dans lequel on se trouve

diapo 48

créer plusieurs objets du même type, ce n’est pas une bonne idée que de créer des objets identiques mais de nom différent

diapo 49

un constructeur permet de créer des objets ayant les même champs.

La fonction (constructeur) que l’on créée doit par convention commencer par un Majuscule. Pour créer des nouveaux objets de ce type on fait : var felicite = new Chat(attr, attr, attr, attr…) ;

Rq : Il existe typescript du javascript typé (mais pour le moment JS toujours très utilisé et typescript juste entrain de percer).

Diapo 50

Les pointeurs vers les fonctions miauler de félicité et de felix ne sont pas les mêmes. Les fonctions sont dupliquées pour chaque chat créé.

Diapo 51

Utilisation d’un pototype du constructeur. Le prototype pointe vers une fonction commune à tous les objets créé pour l’objet chat.

Donc ici l’utilisation de prototype permet d’avoir la même fonction miauler() utilisée par félicité et felix

Diapo 52

On peut utiliser le prototypes pour définir une propriété commune à tous les chats sans avoir à le demander dans les propriétés à chaque création de chat.

Diapo 53

Il est possible de redéfinir une méthode issu d’un prototype pour avec le nom de l’objet en question (ici felix).

Diapo 54

Modification de la méthode prototype cas général (avec Chat). Toutes les fonctions suivantes utiliseront la méthode modifiée.

Diapo 56

Les problèmes du à la non encapsulation des données.

Diapo 57

Classe = objet

Diapo 58

L’opérateur « + » est surchargé, il joue le role du « + » mathématique, mais aussi la concaténation de chaines de caractères. Donc il faut bien vérifier les opérandes donnés au signe « + ».

Diapo 59

Expressions régulières (chercher des patterns dans des chaines de caractères, changer des caractères spécifiques)

Diapo 60

Info sur les tableaux. Il existe le constructeur array pour faire des tableaux. En JS la taille du tableau est dynamique, elle peut évoluer au cours du temps

On peut mélanger les types dans un même tableau

Diapo 61

Quleques méthodes prédéfinie pour les tableaux

Diapo 62

Splice prend en paramètre

Début : indice à partir duquel le changement va commencer

Nb d’élément à supprimer : peut avoir des argument optionnels pour spécifier des éléments à insérer à cet endroit.

Splice revoit les éléments supprimés, donc dans l’exemple tabRes contient « deux » et « trois » (éléments supprimés du tableau).