

Web (Persistence)



Andrea G. B. Tettamanzi

Université de Nice Sophia Antipolis

Département Informatique

andrea.tettamanzi@unice.fr

CM - Séance 9

Introduction à JavaScript

Plan

- Dans cette séance
 - Éléments du langage JavaScript
 - JavaScript et HTML
 - Objets, prototypes et héritage
- Dans les prochaines séances
 - Objets standard et gestion des erreurs
 - Document Object Model
 - Ajax
 - Exemple d'encapsulation d'Ajax

Histoire



- Créé en 1995 par Brendan Eich pour Netscape
- En décembre 1995 Sun et Netscape annoncent sa sortie
- En mars 1996, moteur JavaScript dans NS Navigator 2.0
- Microsoft réagit en développant Jscript, en août dans IE
- En 1997, JavaScript devient un standard ECMA : ECMAScript
- Choix du nom
 - Des raisons de marketing
 - Complément à Java
 - Confusion auprès du publique

Vue d'ensemble du langage

- Orienté objet, mais basé sur les prototypes. Pas de classes
- Un programme est un regroupement d'objets communicants
- Objet = collection de propriétés
- Valeurs primitives : undefined | null | Boolean | Number | String
- Collection d'objets intégrés :
 - Objet global
 - Object, Function, Array, String, Boolean, Number, Math, Date, RegExp, JSON
 - Un certain nombre d'objets erreur : Error, EvalError, ...
- Opérateurs : arithmétiques, relationnels, logiques, etc.
- Syntaxe intentionnellement similaire à Java, mais moins stricte

Types et Expressions

- JavaScript est un langage au typage faible et dynamique
- Le type d'une expression est déterminé pas son résultat
 - Nombres : entiers ou à virgule
 - Chaînes de caractères
 - Booléens (false, true)
 - « null » un type à une seule valeur = absence de données
- Expressions simples : formées par un seul élément
 - Littéral
 - Identifiant d'une variable
 - Le mot-clé « this »
- Expressions composées : avec opérateurs / fonctions / méthodes

Instructions

Le standard ECMAScript prévoit 15 types différents d'instructions :

- Block
- VariableStatement
- EmptyStatement
- ExpressionStatement
- IfStatement
- IterationStatement
- ContinueStatement
- BreakStatement
- ReturnStatement
- WithStatement
- LabelledStatement
- SwitchStatement
- ThrowStatement
- TryStatement
- DebuggerStatement

Block Statement

- Un bloc d'instructions est délimité par des accolades : { ... }

Déclaration de variable

- « Variable statement »
var <identifiant> ;
var <identifiant> = <expression>;
- En fait, complètement facultative !

Expressions

- Exemple : $x + 2 * \text{Math.sin}(\text{theta})$

Structures de contrôle

- Les mêmes que Java – c'est fait exprès !
- Conditionnelles
 - `if(cond) ... [else ...]`
 - `switch(expr) { case ... : ... break ; ... default : ... }`
- Itératives
 - `while(cond) ...`
 - `do ... while(cond) ;`
 - `for(instr ; cond ; instr) ...`

Définition des Fonctions

- Définition de fonction (c'est une « lambda ») :
function(*arguments*) { ... }
- Définition de fonction nommée :
function *nom*(*arguments*) { ... }
- Renvoi d'un résultat :
return(*expression*);
- Appel de fonction :
nom(*arguments*)

JavaScript et HTML

- JavaScript interagit avec sa pages HTML via :
 - La balise `<script>`
 - Les fonctions qui font apparaître des boîtes-éclair
 - Les événements
 - Le modèle orienté objet de documents (DOM)
 - L'API de HTML5
 - Stockage local
 - Géolocalisation
 - Glisser-déposer
 - Web Sockets

La balise `<script>`

- Le code JavaScript est attaché à un document HTML avec la balise `<script>`, qui peut être placée n'importe où
- Il y a deux manières de faire cela (en HTML5) :
 - script « embarqué » (c'est-à-dire, le code fait partie du document HTML),
`<script> code embarqué </script>`
par exemple :
`<script>document.write("Hello World!")</script>`
 - script externe (c'est-à-dire, le code est dans un fichier séparé),
`<script src="URL du fichier "> </script>`
- En HTML 4, type obligatoire : `<script type="text/javascript">`

Les boîtes-éclair

- JavaScript peut faire apparaître trois types de boîte-éclair
- Boîte avertissement
 - `alert(message)`
 - Affiche une boîte-éclair avec le message et un bouton [OK]
- Boîte confirmation
 - `confirm(message)` → true | false
 - Affiche une boîte-éclair avec le message et [Cancel] [OK]
- Boîte invite
 - `prompt(message, texte_par_défaut)` → saisie utilisateur
 - Affiche le message, une case invite initialisée avec le texte par défaut dans laquelle l'utilisateur peut écrire et les boutons [Cancel] et [OK]

Les événements

- Les événements HTML sont des actions de l'utilisateur susceptibles de donner lieu à une interaction
- Par exemple : le clic de souris, les mouvements de la souris, etc.
- On peut associer des fonctions JavaScript à des événements
- L'association d'un événement HTML à une fonction se fait par le biais des *gestionnaires d'événements*
- Exemple :

```
<button type="button" onclick="calc_input(1)">1</button>
```
- Chaque balise accepte un certain nombre d'événements

Événements de la balise `<body>`

<code>onafterprint</code>	Le document vient d'être imprimé
<code>onbeforeprint</code>	Le document va être imprimé
<code>onbeforeunload</code>	Le document va être déchargé
<code>onerror</code>	Une erreur s'est produite
<code>onhaschange</code>	Le document a changé
<code>onload</code>	La page vient de finir d'être chargée
<code>onmessage</code>	Le message est déclenché
<code>onoffline</code>	Le document passe hors-ligne
<code>ononline</code>	Le document passe en ligne
<code>onpagehide</code>	La fenêtre est cachée
<code>onpageshow</code>	La fenêtre devient visible
...	...

Événements de la souris

onclick	L'utilisateur clique dans ou sur un élément
ondblclick	L'utilisateur clique deux fois dans ou sur ...
ondrag	L'utilisateur glisse un élément
ondragend	L'utilisateur termine de glisser un élément
...	...
onmousedown	Bouton de la souris pressé
onmousemove	L'utilisateur bouge la souris
onmouseout	La souris sort d'un élément
onmouseover	La souris bouge sur un élément
onmouseup	Bouton de la souris relâché
onmousewheel	Molette de la souris en rotation
onscroll	La barre de défilement de l'élément défile

Merci de votre attention

