INF2005 – Programmation web

Javascript – partie 2

Jacques Berger

Objectifs

Approfondir le Javascript

Mieux travailler avec le Javascript

Prérequis

HTML

Javascript

Gestion des erreurs

Une erreur d'interprétation du Javascript peut lancer un message d'erreur à l'utilisateur du site

On peut attraper ces erreurs à l'exécution

Gestion des erreurs

Ce qui peut générer une erreur :

- Appel de fonction inexistante
- Utilisation d'une variable inexistante
- Instancier un objet d'une classe inconnue
- Appeler une méthode inexistante

Bref, tout ce qui provoquerait une erreur de compilation dans un langage compilé

Gestion des erreurs

Pour attraper ces erreurs : try/catch

```
try {
    …
} catch (errorcode) {
    …
}
```

Gestion des erreurs

Que faire en cas d'erreur?

- Ignorer l'erreur
- Avertir l'utilisateur
- Avertir le serveur

Gestion des erreurs

Ignorer l'erreur
- Passe l'erreur sous silence
- L'utilisateur ne s'en rend pas compte
- Peut prendre du temps avant que vous ne le sachiez
- Perte de fonctionnalité qui peut durer

Gestion des erreurs

Avertir l'utilisateur

- Donne la responsabilité à l'utilisateur de vous aviser de votre erreur
- Peut nuire à votre crédibilité

Gestion des erreurs

Avertir le serveur

- Se faire un système de logging d'erreur
- Dans le catch, le JS envoie un message au serveur pour décrire l'erreur à corriger
- Le serveur avise une personne de l'erreur

Exceptions

Vous pouvez lancer des exceptions avec throw et les attraper avec try/catch

```
function activateControl() {
    …
    if (value == null) {
        throw "E03";
    }
    …
}
```

Exceptions

```
try {
    activateControl();
} catch (errorcode) {
    if (errorcode == "E03") {
        …
    }
    …
}
```

Exceptions

Il existe un objet Error que l'on peut utiliser pour lancer des exceptions

```
throw new Error("Message décrivant l'erreur");
```

Fonctions libres

Quelques fonctions libres sont offertes avec le langage

- Lire un entier, un float
- Encoder/décoder un URI
- Vérifier des valeurs extrêmes

Voir la référence

Objets

Javascript permet certaines fonctionnalités des langages orienté-objets

- Instanciation d'objets
- Appels de méthodes

Objets

Certains objets sont fournis avec le langage, dont :

- Array
- Date
- Math
- RegExp
- String

Array

Permet des fonctionnalités pour :

- Ajouter des valeurs
- Enlever des valeurs
- Chercher des valeurs

Voir la référence

Date

Permet de faire des lectures et des conversions de dates. Permet également d'utiliser toutes les notions de temps (heure, minute, etc.)

Voir la référence

Math

Contient :

- Des constantes mathématiques
- Des fonctions mathématiques

Voir la référence

RegExp

Ce sont des expressions régulières

Très utile pour faire du pattern matching

Voir la référence

String

Permet toutes les manipulations de chaînes de caractères courantes

Voir la référence

Objets

Vous pouvez créer vos propres objets

Javascript ne possède pas de mécanisme orienté-objet avancé contenant :

- Polymorphisme
- Héritage

Objets

Les objets ont des propriétés et des méthodes

Les propriétés sont des variables publiques

Les méthodes sont des fonctions rattachées à l'objet

Créer un objet

Version longue

```
var vetement = new Object();
vetement.identifiant = "IDX777";
vetement.couleur = "Rouge";
vetement.grandeur = "XL";
```

Créer un objet

Constructeur implicite

```
var vetement = {identifiant:"IDX777", couleur:"Rouge",
grandeur:"XL"};
```

Créer un objet

Constructeur explicite

```
function Vetement(identifiant, couleur, grandeur) {
    this.identifiant = identifiant;
    this.couleur = couleur;
    this.grandeur = grandeur;
}

var chandail = new Vetement("IRS993", "Blanc", "S");
```

Ajouter une méthode

On assigne la méthode à l'objet

```
function obtenirCouleur() {
    return this.couleur;
}

chandail.obtenirCouleur = obtenirCouleur;
```

Ajouter une méthode

On peut assigner les méthodes dans le constructeur

```
function obtenirCouleur() {
    return this.couleur;
}

function obtenirGrandeur() {
    return this.grandeur;
}

function Vetement(identifiant, couleur, grandeur) {
    this.identifiant = identifiant;
    this.couleur = couleur;
    this.grandeur = grandeur;
    this.obtenirCouleur = obtenirCouleur;
    this.obtenirGrandeur = obtenirGrandeur;
}
```

Utiliser les méthodes

Ensuite, on peut utiliser les méthodes

Meilleure pratique : dans le constructeur

```
var bas = new Vetement("SSR994", "Noir", "1Size");
element.innerHTML = bas.obtenirCouleur();
```

Fonction anonyme

Une fonction sans nom qu'on code directement au niveau de l'appelant

```
xmlHttp.onreadystatechange = function() {
    var container = document.getElementById("page");
    container.innerHTML = xmlHttp.responseText;
}
```

Validation

La validation de formulaires est un cas d'utilisation très courant de Javascript

Elle permet de valider certains champs au niveau du client afin d'alléger la charge du serveur

Validation

Voir exemple

Timeout

Utiliser un compte à rebours pour lancer un événement

- setTimeout
- clearTimeout

Redirection

Rediriger l'utilisateur vers un autre site web de façon automatique

L'ancienne façon de faire avec les métadonnées

```
<meta http-equiv="refresh" content="5; URL=http://yougotrickrolled.com/">
```

Redirection

Maintenant, on le fait avec Javascript

```
window.location = "http://yougotrickrolled.com/";
```

Si on veut un délai, on utilise la méthode setTimeout de Javascript

Voir exemple

Maintenabilité

Le Javascript est du code de production tout comme du code Java ou C++

Il est appelé à vieillir et à être maintenu

Il est donc important de s'appliquer à écrire du code de qualité

Maintenabilité

Tout ce qu'on appliquerait à Java est aussi vrai avec Javascript :

- Réduire la portée des variables
- Bien nommer ses variables
- Bien nommer ses fonctions
- Écrire de petites fonctions
- Soigner l'indentation
- Etc.

Maintenabilité

Certains outils peuvent vous aider

JSLint
- http://www.jslint.com/
- Outil très restrictif
- Plugins disponibles

Maintenabilité

JSHint
- http://www.jshint.com/
- Moins restrictif que JSLint

Obfuscator

Le code Javascript est ouvert à tous

Tous les utilisateurs peuvent voir votre code Javascript dans le code source de la page web

Tout est ouvert sur le web...

Obfuscator

Certaines entreprises désirent garder leur code secret pour éviter que la concurrence ne copie leurs fonctionnalités

L'utilisation du Javascript est donc problématique pour eux

Obfuscator

Il existe des outils pour transformer le Javascript afin de le rendre complètement illisible pour un humain

On appelle ça un Javascript Obfuscator
http://javascriptobfuscator.com/

Obfuscator

Le résultat est pratiquement impossible à maintenir mais le code ne peut pas être lu et compris par un autre développeur (sauf s'il est très tenace)

Obfuscator

Pour cacher du code, l'idéal demeure toujours de placer ce code dans la technologie du backend

JS et PHP

Dans un script PHP, vous pouvez générer du code Javascript dans votre page HTML

Le Javascript peut être écrit directement dans la page HTML; le code ne sera pas interprété par PHP puisqu'il n'est pas dans une balise PHP

JS et PHP

Un script PHP peut devenir difficile à maintenir s'il contient plusieurs langages différents

- HTML
- CSS
- Javascript
- PHP

JS et PHP

Pour contrer ce problème, isolez votre code Javascript dans un fichier sur votre serveur

Dans le script PHP, faites un require de votre Javascript

Ainsi, le Javascript sera inclut dans le HTML sans polluer votre script PHP avec un langage supplémentaire

Plus loin...

Référence Objets JS
    http://www.w3schools.com/jsref/default.asp

JSLint
    http://www.jslint.com/

JSHint
    http://www.jshint.com/

Javascript Obfuscator
    http://javascriptobfuscator.com/

Javascript Compressor
    http://javascriptcompressor.com/