

Javascript

Thierry Lecroq

Université de Rouen
FRANCE

Plan

1 Généralités sur Javascript

2 Les bases

3 Les objets prédéfinis

4 Les évènements

Exemple

```
<html>
  <head>
    <title>Page statique</title>
  </head>
  <body>
    <div>
      Nous sommes le 2/10/2008
    </div>
  </body>
</html>
```

Exemple

```
<html>
  <head>
    <title>Page dynamique</title>
  </head>
  <body>
    <script type = "text/javascript">
      //
        date = new Date();
        document.writeln("Nous sommes le ", date);
      //]]&gt;
    &lt;/script&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="875 877 988 920" data-label="Page-Footer"><img alt="University of Rouen logo" data-bbox="875 877 988 920"/><p>UNIVERSITÉ DE ROUEN</p></div><div data-bbox="637 933 988 955" data-label="Page-Footer"><img alt="Navigation icons" data-bbox="637 933 988 955"/></div><div data-bbox="47 966 279 993" data-label="Page-Footer"><p>Thierry Lecroq (Univ. Rouen)</p></div><div data-bbox="457 966 537 992" data-label="Page-Footer"><p>Javascript</p></div><div data-bbox="932 966 987 992" data-label="Page-Footer"><p>4 / 36</p></div>
```

Script

- Portion de code qui vient s'insérer dans une page HTML
- Le code du script n'est toutefois pas visible dans la fenêtre du navigateur car il est compris entre des balises spécifiques qui signalent au navigateur qu'il s'agit d'un script écrit en langage JavaScript
- Interprété du côté client

- interne

```
<script type = "text/javascript">  
//<![CDATA[  
code javascript  
//]]>  
</script>
```

- externe

```
<script type = "text/javascript" src = "url/script.js">  
</script>
```

- les éléments situés dans l'en-tête se comportent comme des déclarations, ils ne s'exécutent pas directement
- les éléments situés dans le corps s'exécutent au fur et à mesure du chargement de la page

Javascript n'est pas Java

- Java a été développé par Sun
- JavaScript a été développé par Netscape en 1995 sous le nom de LiveScript
- Microsoft développe le langage Script en 1995
- \Rightarrow norme des langages de script par l'ECMA (European Computer Manufacturers Association)
- basé sur les objets pas de classe
- Java est compilé (applets), Javascript est interprété (scripts)
- ne peut pas lire/écrire dans les fichiers
- ne peut pas exécuter d'autres programmes

- Utilisation de la balise `<script>...</script>` :
 - ▶ déclaration de fonctions dans l'en-tête HTML/XHTML (entre `<head>` et `</head>`)
 - ▶ appel d'une fonction ou exécution d'une commande JavaScript dans `<body>...</body>`
 - ▶ insertion d'un fichier externe (usuellement `'.js'`)
- Utilisation dans une URL, en précisant que le protocole utilisé est du JavaScript ex :
`Texte`
- Utilisation des attributs de balise pour la gestion événementielle :
`<balise onEvenement="instructionJavaScript">...</balise>`

```

<?xml version="1.0" encoding="utf8"?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<title>Exemple de page HTML contenant du JavaScript</title>
<script type="text/javascript">
    <!--
        function texte() { document.write("Texte généré."); }
    // -->
</script>
</head>
<body>
<script type="text/javascript">
    <!--
        document.write("Vous pouvez mettre du code javascript dans le corps du document
    // -->
</script>
<p>
    Ou bien dans une fonction appelée en cliquant
    <a href="Javascript:texte()">ici</a>,
<p>
    ou en passant la souris au-dessus de
    <a href="" onMouseOver="texte()">cela</a>...
</body>
</html>

```

Exemple de JavaScript inséré dans du code XHTML 1.0

```
<?xml version="1.0" encoding="utf8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Exemple de page XHTML contenant du Javascript</title>
    <script type="text/javascript">
      <![CDATA[
        function fenetre() { alert('Message d'alerte dans une fonction.')}
      ]]>
    </script>
  </head>
  <body onload="alert('Message d'alerte généré à la fin du chargement.')">
    <script type="text/javascript">
      <![CDATA[
        alert('Message d'alerte dans le corps du document. ');
      ]]>
    </script>
    <p>
      Ceci est le corps du document.
      <a href="javascript:fenetre()">Message d'alerte</a>.
    </p>
  </body>
</html>
```

Plan

- 1 Généralités sur Javascript
- 2 Les bases**
- 3 Les objets prédéfinis
- 4 Les évènements

4 types de base

entier : 127 (base 10), 0755 (base 8), 0xFA15 (base 16)

flottant : 0.123, -0.4e5, .67E-89

booléen : true, false

chaîne de caractères : "chaine" ou 'chaine'

Typage et Variable

Pas de déclaration des variables

```
nbr = 10;  
fl = 3.141;  
str1 = "L'étoile";  
str2 = 'brille';  
lien = '<a href="index.htm">Home</a>';
```

Portée des variables

- locale (uniquement dans le script ou la fonction)

```
var vloc = 0 ;
```

- globale (en tout point du document)

```
vglob = 0 ;
```

Expressions

- arithmétique

`(3+4) * (56.7 / 89)`

- chaîne de caractères

`"L'étoile" + " " + "filante"`

- logique

`temp == 37`

`h2o = (temp<100) ? "eau" : "vapeur";`

`h2o = (temp>0) ? ((temp<100) ? "eau" : "vapeur") : "glace";`

Opérateurs

- affectation

`+=, -=, *=, /=, %=, &=, |=, <<=, >>=`

- comparaison

`==, !=, <, <=, >, >=`

- arithmétique

`%, ++, --`

- logique (évaluation paresseuse)

`&&, ||, !`

- bit

`&, |, ^ (XOR), <<, >>, >>>`

Structures de contrôle

if else, switch case, for, while, break, continue, do while

Fonctions

Définition

```
function nomfonction(param1, ..., paramN) {  
    // code JavaScript  
    return expression ;  
}
```

Appel

```
nomVariable = nomfonction(exp1, ..., expN);
```

passage des paramètres par valeur

Arguments

```
function somme() {  
    var argv = somme.arguments;  
    var argc = somme.arguments.length;  
    var result = 0;  
    for (var i = 0 ; i < argc ; i++) {  
        result += argv[i];  
    }  
    return result;  
}
```

somme(1,2,3) retourne 6 et somme(2) retourne 2

`prompt()`

Ouvre une boîte de dialogue avec une zone saisie et 2 boutons OK et Annuler, retourne l'information lue

`confirm()`

Ouvre une boîte de dialogue avec 2 boutons OK et Annuler, retourne un booléen

`alert()`

Permet d'écrire un message dans une fenêtre

Exemple

```
<html>
  <head>
    <title>Utilisation de prompt() et d>alert()</title>
  </head>
  <body>
    <script type = "text/javascript">
      //
        annee = prompt('En quelle année sommes-nous ? ', 2000);
        alert('Vous avez répondu : ' + annee);
      //]]&gt;
    &lt;/script&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="877 877 988 921" data-label="Page-Footer"><img alt="University of Rouen logo" data-bbox="877 877 988 921"/><p>UNIVERSITÉ DE ROUEN</p></div><div data-bbox="638 933 988 958" data-label="Page-Footer"><img alt="Navigation icons" data-bbox="638 933 988 958"/></div><div data-bbox="48 966 279 993" data-label="Page-Footer"><p>Thierry Lecroq (Univ. Rouen)</p></div><div data-bbox="458 966 537 992" data-label="Page-Footer"><p>Javascript</p></div><div data-bbox="924 966 987 992" data-label="Page-Footer"><p>21 / 36</p></div>
```

Écrire, afficher des informations dans la fenêtre HTML

Exemple

```
<html>
  <head>
    <title>Utilisation de document.write</title>
  </head>
  <body>
    <script type = "text/javascript">
      //
        document.write('Hello word &lt;br/&gt;');
      //]]&gt;
    &lt;/script&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="877 868 988 922" data-label="Page-Footer"><img alt="Logo of the University of Rouen" data-bbox="877 868 988 922"/><p>UNIVERSITÉ DE ROUEN</p></div><div data-bbox="638 933 988 958" data-label="Page-Footer"><img alt="Navigation icons" data-bbox="638 933 988 958"/></div><div data-bbox="48 966 279 993" data-label="Page-Footer"><p>Thierry Lecroq (Univ. Rouen)</p></div><div data-bbox="458 966 537 992" data-label="Page-Footer"><p>Javascript</p></div><div data-bbox="924 966 987 992" data-label="Page-Footer"><p>22 / 36</p></div>
```

Ouverture d'une fenêtre

`open(url, name, options)`

Permet d'ouvrir une fenêtre et d'écrire dedans

Exemple

```
<script type = "text/javascript">
  //<![CDATA[
    fenetre = open('', '', 'height=50, width=300, status=yes');
    fenetre.document.write('<html>');
    fenetre.document.write('<head>');
    fenetre.document.write('<title>' + 'Titre fenetre' + '</title>');
    fenetre.document.write('</head>');
    fenetre.document.write('<body>');
    fenetre.document.write('Texte dans la fenetre');
    fenetre.document.write('</body>');
    fenetre.document.write('</html>');
  //]]>
</script>
```

Ouverture d'une fenêtre

Les options

`directories` : barre de liens

`menubar` : barre de menu

`status` : barre de statut

`location` : barre d'adresse

`scrollbars` : ascenseurs

`resizable` : redimensionnement par l'utilisateur

`height` : hauteur

`width` : largeur

`left` : position gauche

`top` : position haute

`fullscreen` : plein écran

Plan

- 1 Généralités sur Javascript
- 2 Les bases
- 3 Les objets prédéfinis**
- 4 Les évènements

Les évènements

- `onclick` : un clic du bouton gauche de la souris sur une cible
- `onmouseover` : passage du pointeur de la souris sur une cible
- `onblur` : une perte de focus d'une cible
- `onfocus` : une activation d'une cible
- `onselect` : une selection d'une cible
- `onchange` : une modification du contenu d'une cible
- `onsubmit` : une soumission d'un formulaire
- `onload` : un chargement d'une page
- `onunload` : la fermeture d'une fenetre ou le chargement d'une page autre que la courante

Les évènements

Exemple

```
<a href="" onclick="alert('Bonjour')">Cliquez</a>
```

Plan

- 1 Généralités sur Javascript
- 2 Les bases
- 3 Les objets prédéfinis
- 4 Les évènements

L'objet document

Propriété

title

Méthode

write

L'objet Date

```
var maDate = new Date()
```

- `getFullYear()` : 2 chiffres
- `getFullYear()` : 4 chiffres
- `getMonth()` : 0 – 11
- `getDate()` : 1 – 31
- `getDay()` : 0 – 6 (dimanche – samedi)
- `getHours()` : 0 – 23
- `getMinutes` : 0 – 59
- `getSeconds()` : 0 – 59

L'objet String

Lorsqu'on définit une constante ou une variable chaîne de caractères, JavaScript crée d'une façon transparente une instance String

1 propriété : `length`

les balises HTML/XHTML ont leur équivalent en méthode

Liste (non exhaustive) des méthodes :

`bold()`, `italics()`, `fontcolor()`, `fontsize()`,
`small()`, `big()`, `toUpperCase()`, `toLowerCase()`,
`sub()`, `sup()`, `substring()`, `eval()`, `split()`, `replace()`

L'objet RegExp

```
var re = new RegExp(motif, option)
```

- règles classiques pour la formation des motifs
- options possibles : "g", "i", "gi", ""

Les méthodes

- `test()` : booléen
- `exec()` : retourne la première occurrence
- `match()` : s'applique à un objet de type `String`

L'objet Math

Propriétés : `Math.PI` et `Math.E`

Méthodes :

```
atan(), acos(), asin(), tan(), cos(), sin(),  
abs(), exp(), log(), max(), min(), pow(),  
round(), sqrt(), floor(), random()
```

Les tableaux

```
var T = new Array()
```

Les tableaux classiques

```
var jours = new Array();
```

```
var jours = new Array("Lundi", "Mardi", "Mercredi", "Jeudi",  
                      "Vendredi", "Samedi", "Dimanche");
```

```
jours[0]
```

```
jours.length
```

Les tableaux associatifs

```
var tableau = new Array();  
  
tableau["un"] = "La première chaîne";  
tableau["deux"] = "La deuxième chaîne";  
tableau["tnt"] = "Plein d'autres chaînes";  
  
tableau["un"]  
  
tableau.length
```

Les méthodes de l'objet Array

```
var tableau3=tableau1.concat(tableau2);  
  
var chaine=tableau.join(séparateur);  
  
tableau.pop();  
  
tableau.push(liste d'éléments);  
  
tableau.reverse();  
  
tableau.shift();  
tableau.unshift(liste d'éléments);  
  
tableau2=tableau1.slice(début, fin);  
  
tableau.sort();  
  
tableau.splice(début, longueur);  
tableau.splice(début, longueur, liste d'éléments);
```

L'objet form

Attributs

- `name` : nom
- `action` : fichier
- `method` : get ou post
- `enctype` : encodage
- `target` : cadre cible (`_blank` pour une nouvelle page)

Méthodes

- `submit()` : soumission
- `reset()` : remise à zéro

Évènements

- `onSubmit()` : lors de la soumission
- `onReset()` : lors de la remise à zéro

Les éléments de formulaires

input text

```
<input type="text" id="motclef" value="Mot clef">
```

```
document.forms["monFormulaire"].elements["motcle"]
```

Les propriétés :

- value : valeur
- defaultValue : valeur par défaut
- form : objet formulaire
- maxLength : longueur maximale

Les méthodes :

- blur() : perte de focus
- focus() : prise de focus
- select() : donne le focus et sélectionne la zone de saisie

Les événements :

- onBlur : lors de la perte de focus
- onChange : lors d'un changement
- onFocus : lors de la prise de focus

Les éléments de formulaires

input button

Les propriétés :

- value : libellé

Les méthodes :

- click() : clic

Les événements :

- onClick : lors d'un clic

Les éléments de formulaires

select

Les propriétés :

- size : nombre de lignes
- options : tableau
 - ▶ value : valeur
 - ▶ text : libellé
 - ▶ defaultSelected : true of false
 - ▶ selected : true of false
- selectedIndex : indice de la ligne sélectionnée

Example

```
<script type="text/javascript">
<!--
function basculer(orig, dest) {
    if (orig.options.selectedIndex >= 0) {
        var o = new Option(orig.options[orig.selectedIndex].text,
                           orig.options[orig.selectedIndex].value);
        dest.options[dest.options.length]=o;
        orig.options[orig.selectedIndex]=null;
    }
    else {
        alert("Aucune ligne sélectionnée");
    }
}

function monter(f) {
    var no=f.choix.selectedIndex;
    if (no<0) { return true; }
    if (no>=1) { inverser(f.choix, no, no-1); }
}

function baisser(f) {
    var no=f.choix.selectedIndex;
    if (no<0) { return true; }
    if (no<f.choix.options.length-1) { inverser(f.choix, no, no+1); }
}
```

Exemple

```
function inverser(liste, ind1, ind2) {
    var valeur=liste.options[ind1].value;
    var texte=liste.options[ind1].text;
    liste.options[ind1].value=liste.options[ind2].value;
    liste.options[ind1].text=liste.options[ind2].text;
    liste.options[ind2].value=valeur;
    liste.options[ind2].text=texte;
    liste.selectedIndex=ind2;
}

function verifierListe(f) {
    if (f.choix.options.length == 0) {
        alert("Indiquez les raisons vous ayant poussé à créer votre site");
        return false;
    }
    for (var i=0; i<f.choix.options.length; i++) {
        f.choix.options[i].selected=true;
    }
    f.submit();
}
// -->
</script>
```

Exemple

```
<form id="nomFormulaire" method="get" action="formu.html">
```

Double cliquez sur les raisons qui vous ont poussé ; à ; créer vot.

Internet, puis ordonnez les par ordre d'importance :

```
<table>
```

```
<tr>
```

```
<td>
```

```
Possibilit&eacute;s :</br>
```

```
<select id="raison" size="5" multiple="multiple"
```

```
style="width:250px;" onDblClick="basculer(this, this.form.choix)">
```

```
<option value="1">Apprendre le langage</option>
```

```
<option value="2">Parler de ma passion</option>
```

```
<option value="3">Cr&eacute;er une activit&eacute; commerciale</option>
```

```
<option value="4">&Eacute;changer avec le monde entier</option>
```

```
<option value="5">Faire mon m&eacute;tier</option>
```

```
</select>
```

```
</td>
```

Exemple

```
<td>
Vos choix :<br/>
<select id="choix" size="5" multiple="multiple"
        style="width:250px;" onDblClick="basculer(this, this.form.raison)">
</select>
</td>
<td>
<input type="button" value="Monter" onclick="monter(this.form)"><br/>
</td>
<td>
<input type="button" value="Baisser" onclick="baisser(this.form)"><br/>
</td>
</tr>
</table>
<p>
<input type="button" value="Valider" onclick="verifierListe(this.form)">
</p>
</form>
```

Plan

- 1 Généralités sur Javascript
- 2 Les bases
- 3 Les objets prédéfinis
- 4 Les évènements

Définition d'objets simples

Exemple

```
function Chien(nom, race, maitre) {  
    this.nom = nom;  
    this.race = race;  
    this.maitre = maitre;  
    this.print = printChien;  
}  
  
function printChien() {  
    document.write("Chien ", this.nom, " de race ", this.race,  
        " appartenant &agrave; ", this.maitre, "<br>");  
}
```

Définition d'objets simples

Exemple

```
function printChien() {  
  with (this) {  
    document.write("Chien ", nom, " de race ", race,  
                  " appartenant &agrave; ", maitre, "<br/>");  
  }  
}  
  
rantanplan = new Chien("Rantanplan", "indéfinie", "Lucky Luke");  
rantanplan.print();
```