Introduction à Javascript

Code: js-intro

Originaux

url: **http://tecfa.unige.ch/guides/tie/html/js-intro/js-intro.html**
url: **http://tecfa.unige.ch/guides/tie/pdf/files/js-intro.pdf**

Auteurs et version

- Daniel K. Schneider - Patrick Jermann - Vivian Synteta - Olivier Clavel
- Version: 1.1 (modifié le 15/8/01 par DKS)

Prérequis:

- Connaître (savoir lire) le langage HTML

Module technique précédent: html-intro
Module technique précédent: html-forms

Suites possibles: PHP ou JSP/JHTML

Module technique suppl.: php-intro et ensuite php-html

Module technique suppl.: java-intro et ensuite java-jsp

Objectifs:

1. Connaître les origines et les principes de JavaScript
2. Appliquer quelques exemples simples d'utilisation:
   - Obtenir des informations sur le browser
   - Vérification de formulaires
   - Quiz on-line
3. (pas de DHTML)

---

1. Table de matières détaillée

1. Table de matières détaillée 3
2. JavaScript (client-side), introduction 4
   - **2.1 Origine 4**
   - **2.2 Utilisation principale de JavaScript 4**
   - **2.3 Versions 5**
   - **2.4 Ressources et outils de développement 5**
   - **2.5 Caractéristiques du langage JavaScript 7**
   - **2.6 Insertion de code JavaScript dans une page HTML 14**
   - **2.7 Utilisation de code JavaScript 17**
3. Manipulation du browser et de fenêtres 19
   - **3.1 Informations sur le browser 19**
   - **3.2 Ouvrir une nouvelle fenêtre 21**
4. Traitement de formulaires avec Javascript 24
   - **4.1 Un simple quiz 24**
   - **4.2 Vérification d'un formulaire I 28**
   - **4.3 Vérification d'un formulaire (II) 29**
5. HTML Dynamique (ou presque) 33
   - **5.1 Les boutons JavaScript 33**
   - **5.2 Menus déroulants 34**
6. Javascript et DOM 35

---

2. JavaScript (client-side), introduction

2.1 Origine

- JavaScript a été développé par Netscape (d'abord sous le nom LiveScript)
- Il existe une version Microsoft (appelée parfois JScript)
- Le nom "JavaScript" reflète un certain voisinage syntactique avec JAVA (et il a été choisi pour des raisons de marketing)

2.2 Utilisation principale de JavaScript

- Applications interactives (par ex. tests et quiz)
- Pages HTML plus riches (par ex. "highlighting"), DHTML
- Génération de pages HTML selon le profil de l'utilisateur
- Gestion de plugins, versions de Java etc.
- Simples animations
- Vérification de formulaires cgi-bin

2.3 Versions

- JS 1.0: Netscape 2 et plus
- JS 1.1: Netscape 3 et plus
- JS 1.2: Netscape 4 et plus, IE 5 (avec qq déficiences)
- JS 1.3: Netscape 4.06 et 4.5 et plus
- JS 1.5: Netscape 6
- ECMA Script: JS 1.1 standardisé (et sans objets propriétaires)

2.4 Ressources et outils de développement

A. Outils de développement

- on conseille l'utilisation d'un éditeur simple
  - sinon Netscape vend un outil "GUI" et certains outils commerciaux peuvent générer du JavaScript
- pour tester des fonctions JavaScript simples:
  - utiliser la console Javascript de Netscape 4.x, taper l'URL:

    ```
    JavaScript:
    ```

  - ou l'URL "JavaScript:" suivi par la fonction, par ex:

    ```
    JavaScript:alert("salut")
    ```

- pour debugger, ouvrir la console Javascript (NS 6: dans un menu)

B. Tutoriels on-line

- Voir les guides de Netscape

  url: **http://tecfa.unige.ch/guides/js/pointers.html#tuto**
- Par imitation: sur le réseau il existe des programmes JavaScript
  - certains peuvent être utilisés "tel quel"
  - d'autres doivent être adaptés à vos besoins

C. Livres

- Danny Goodman's Javascript Bible
- Le livre Javascript chez O'Reilly
- Cf. aussi les "Roadmaps" de Danny Goodman

  url: **http://www.dannyg.com/javascript/**

  url: **(copies NS 4.x dans: http://tecfa.unige.ch/guides/js/dgmaps/)**

D. Pointeurs

url: http://tecfa.unige.ch/guides/js/pointers.html

2.5 Caractéristiques du langage JavaScript

A. Caractéristiques clefs:

- Syntaxe: ressemble à Java, C, Php etc.
- Langage basé objets (pas orienté objets)
- Des objets "browser", "HTML", etc. sont "built-in" et permettent la manipulation de contenus "Web" (notamment formulaires et liens/images). Il existe 3 versions principales de ces objets DOM (Document Object Model):
  - IE 3,4 + NS 3
  - NS 4.x
  - NS 6 + IE 5.5
- Intégration avec HTML

☞ Les éléments dans cette section ne sont qu'indicatifs !

- Il est vivement conseillé d'aller consulter la documentation Netscape/Microsoft et/ou un tutoriel sur le WWW,

  url: http://tecfa.unige.ch/guides/js/pointers.html#man

B. Les objets

- JavaScript est un langage "basé objets":
  - le programmeur peut "utiliser" des objets "JavaScript" pré-définis
  - il peut définir des objets simples (sans héritage)
  - tout élément conceptuel utilisé, affiché etc. dans le Navigateur est représenté par un objet qui s'insère dans une structure DOM
- un objet est une structure informatique qui:
  - possède des propriétés (variables contenant de l'information);
  - possède des méthodes (procédures permettant de manipuler les informations contenues dans l'objet);
  - (parfois) possède des "event handlers" associés.
- Les objets "builtin" sont organisés hiérarchiquement, le fameux DOM (Document Object model). Le DOM represente la structure informatique d'une page telle qu'elle est accessible pour le programmeur.

C. Instructions liées aux objets

- une propriété ou une méthode d'un objet est invoquée par:

```
objectName.propertyName
objectName.methodName(arguments...)
```

Méthodes (fonctions) ou propriétés (variables attachées aux objets) vous permettent de traiter une page HTML (essentiellement des formulaires), créer des boites de dialogues, de nouvelles pages, faire du DHTML etc.

- Création d'objets: new

```
objectName = new objectType ( param1 [,param2] ...[,paramN] )
```

- this

```
"this" réfère à l'objet courrant.
```

- with

```
with (object){
        statements
}
```

Exemple 2-1: l'objet "document":

- L'objet "document" représente toute information attachée au contenu d'une page HTML, par exemple:
  - le contenu des formulaires
  - le "title" du document
- Il contient (selon la version de JavaScript):
  - environ 7 "event handlers" pour gérér les gestes de l'utilisateur
  - envrion 20 propriétés
  - environ 9 méthodes
- Il possède 8 objets enfants (en JS 1.2)

Grace à ces structures et méthodes informatiques vous pouvez créer dynamiquement des documents ou encore obtenir qq informations sur le contenu d'un document.

Exercice 1: Trouvez la documentation de l'objet "document"

Etape 1.a: voir: http://tecfa.unige.ch/guides/js/pointers.html#man

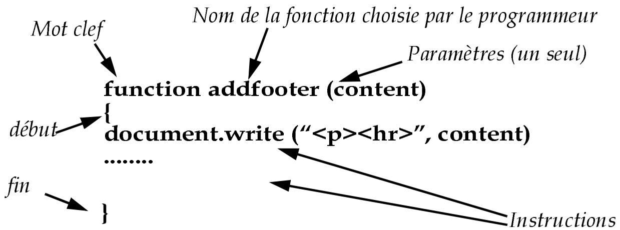
Etape 1.b: sélectionner un manuel de référence Javascript

Etape 1.c: chercher ....

D. La fonction

- Il s'agit d'une procédure (méthode attachée à votre page) qui:
  - calcule quelque chose,
  - retourne une valeur (ou non),
  - appelle d'autres fonctions (ou non),
  - fait appel à des méthodes JavaScript (ou non).
- Voici la syntaxe et un exemple abstrait

```
function <nom> (<paramètre, ......) {
   <instructions >
}
```



Mot clef

Nom de la fonction choisie par le programmeur

Paramètres (un seul)

function addfooter (content)

début **{**

document.write ("<p><hr>", content)

....

fin **}**

Instructions

E. Variables

- peuvent mais ne doivent pas être initialisées

`var number = 10;`

- JS supporte plusieurs types

F. Opérateurs arithmétiques et logiques

- à peu près comme dans C, Java, Php

G. Instructions de contrôle

- if .... else: différent de PHP !

```
if (condition) {
          statements1
[ } else {
          statements2 ]
}
```

- for

```
for ([initial-expression]; [condition]; [increment-expression]) {
   statements
}
```

- while

```
while (condition) {
          statements
}
```

- break (dans un while ou for loop)
- switch (JS 1.1 et +?)

```
switch (expression){
   case label :
      statement;
      break;
   case label :
      statement;
      break;
   ...
   default : statement;
}
```

- ....

2.6 Insertion de code JavaScript dans une page HTML

Il existe plusieurs possibilités:

A. Insertion dans un fichier *.html

- Utilisation de la balise “script”
  `<script language=JavaScript>`
- on l’insère normalement toutes les fonctions et initialisations dans le “head” de la page HTML
  - cela assure que les procédures soient connues, avant d’être appelées

emp e Bon our a ec a a cript

url: **Pour voir:** `http://tecfa.unige.ch/guides/js/ex-intro/hello-world1.html`

- programme sayhello() qui permettra d’afficher une petite fenêtre avec le texte: "Bonjour cher lecteur !!!"

```
<HEAD>
<TITLE>Hello World avec JavaScript (18-Nov-1997)</TITLE>

<script language=JavaScript>
  // ICI on definit une fonction JavaScript
  function sayhello() {
    alert("Bonjour cher lecteur !!! ")
  }
</script>
</HEAD>
<BODY>
```

B. Autre possibilités:

- Le code réside dans fichier externe:

```
<SCRIPT SRC=”mon_programme.js”>
...
</SCRIPT>
```

- Certains tags HTML peuvent contenir des paramètres avec des valeurs en JavaScript (à suivre)

C. Gestion de browsers qui ne comprennent pas JavaScript

Cacher un script:

```
<SCRIPT>
<!-- Begin to hide script contents from old browsers.

..... expressions JavaScript...

// End the hiding here. -->
</SCRIPT>
```

Afficher un contenu alternatif pour ces browsers: <noscript>

```
<NOSCRIPT>

<B>This page uses JavaScript, so you need to get Netscape Navigator 2.0
or later!
<BR>
<A HREF=”http://home.netscape.com/comprod/mirror/index.html”>
<IMG SRC=”NSNow.gif”></A>
If you are using Navigator 2.0 or later, and you see this message,
```

```
you should enable JavaScript by on the Advanced page of the
Preferences dialog box.

</NOSCRIPT>
```

Gérer différentes versions de JavaScript:

- tous les sites "Webmaster" et/ou Javascript
- ou directement le JavaScript guide, par exemple:
  url: **http://tecfa.unige.ch/guides/js/jsguide12/index.htm**

D. Commentaires:

- Toute ligne qui commence par un // est un commentaire:

```
<script language=JavaScript>
  // CECI EST UN COMMENTAIRE
  // faites des commentaires pour documenter votre code !
```

2.7 Utilisation de code JavaScript

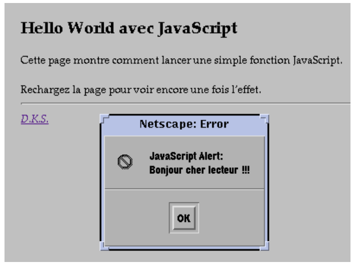
☞ une fonction JavaScript est déclenchée par 2 moyens:

A. Appel de fonction dans le <body> d'une page HTML

- Pour déclencher la fonction définie à la page 14,
  insérez le code suivant quelque part dans votre page:

```
<script>
 // ICI on apelle la fonction
sayhello()
</script>
```

url: **Pour voir:** `http://tecfa.unige.ch/guides/js/ex-intro/hello-world1.html`



Hello World avec JavaScript

Cette page montre comment lancer une simple fonction JavaScript.

Rechargez la page pour voir encore une fois l'effet.

D.K.S.

JavaScript Alert:
Bonjour cher lecteur !!!

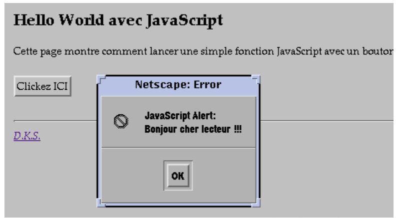
B. Appel de fonction par un "événement"

la notion d'événement en bref:

- un événement est produit par un "geste" de l'utilisateur
- par exemple: bouger la souris, cliquer sur un bouton, remplir un champs etc.
- JavaScript pré-définit un certain nombre d'événements que l'on peut exploiter

Exemple 3: Hello avec un événement JavaScript

```
<FORM METHOD="post">
<INPUT TYPE="button" VALUE="Clickez ICI" onClick="sayhello()">
</form>
```



Hello World avec JavaScript

Cette page montre comment lancer une simple fonction JavaScript avec un bouto

D.K.S.

JavaScript Alert:
Bonjour cher lecteur !!!

- **"onClick"** est un <u>événement</u> qui est déclenché quand l'utilisateur clique sur le bouton.
- Une fois l'événement déclenché, la fonction "sayhello" est exécutée.

url: **Pour voir:** `http://tecfa.unige.ch/guides/js/ex-intro/hello-world2.html`

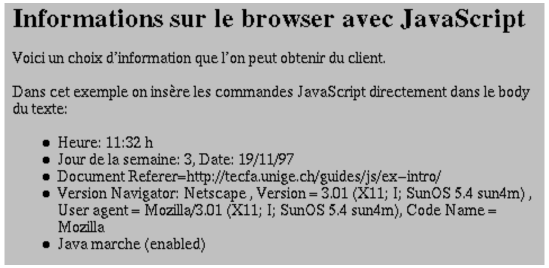
3. Manipulation du browser et de fenêtres

3.1 Informations sur le browser

- Avec JavaScript on peut obtenir de l'information sur le client que vous utilisez
- Cette exemple montre aussi comment générer du HTML avec Javascript
  - Attention: Une fois la page chargée dans le browser, on ne peut plus rien écrire (sauf dans input widgets, layers, etc.)

Exemple 3: Informations sur le browser avec JavaScript

url: **Pour voir: http://tecfa.unige.ch/guides/js/ex-intro/info.html**



Informations sur le browser avec JavaScript

Voici un choix d'information que l'on peut obtenir du client.

Dans cet exemple on insère les commandes JavaScript directement dans le body du texte:

- Heure: 11:32 h
- Jour de la semaine: 3, Date: 19/11/97
- Document Referer=http://tecfa.unige.ch/guides/js/ex–intro/
- Version Navigator: Netscape , Version = 3.01 (X11; I; SunOS 5.4 sun4m) , User agent = Mozilla/3.01 (X11; I; SunOS 5.4 sun4m), Code Name = Mozilla
- Java marche (enabled)

Voici le code:

```
<script language="JavaScript">

// pour connaître l'heure et la date il faut d'abord créer un objet Date.
today = new Date()

document.write("<li>Heure: ",today.getHours(),":",today.getMinutes(), " h ")

document.write("<li>Jour de la semaine: ", today.getDay(), ", Date: ", today.getDate(),"/",
today.getMonth()+1,"/",today.getYear());

document.write("<li>Document Referer=", document.referrer);

document.write("<li>Version Navigator: ", navigator.appName, " , Version = ",
navigator.appVersion, " , User agent = ", navigator.userAgent, ", Code Name = ",
navigator.appCodeName);

if (navigator.javaEnabled()) {
   document.write("<li>Java marche (enabled)");
   }
   else
   document.write("<li>Java ne marche pas (faites quelque chose!)");

</script>
```

- document.write() est une méthode pour "écrire" dans le document courant.
- Important: Il n'est pas possible de modifier ce qui a été écrit sans recharger la page.

3.2 Ouvrir une nouvelle fenêtre

Exemple 3 Création d'une nouvelle fenêtre I

url: **http://tecfa.unige.ch/guides/js/ex-intro/fenetre.html**

- Notez comment on appelle les méthodes: `nom_objet.méthode()`
- la méthode `window.open()` permet d'ouvrir une nouvelle fenêtre du browser (voir la documentation pour les paramètres)
- Lorsqu'on ouvre une nouvelle fenêtre, ca crée un objet qu'il faut stocker dans une variable (ici: `win`) afin de pouvoir l'utiliser

```
win = window.open("", "Resultats", .....);
```

- la méthode `xxx.document.open` initialise l'objet document de la fenêtre (dans lequel on va "écrire")

```
win.document.open();
```

- "document" est une propriété de l'objet xxx qui contient l'objet document associé avec le window xxx.
- xxx.document.writeln () permet d"écrire" du code HTML
- xxx.document.close va finaliser la page (afficher le contenu pour l'utilisateur).

```
win.document.close();
```

- Note: document.writeln ("bla") (sans le xxx) aurait été utilisé pour écrire quelque chose dans la page HTML courante, mais comme on écrit depuis la page courante vers une <u>autre</u> fenêtre, il faut utiliser:

```
win.document.writeln ("Salut");
```

Voici le code:

```
function ouvrir () {
  // on crée un nouvelle fenêtre
  win = window.open("", "Resultats",        "width=250,height=150,status=1,resizable=1");
  // on ouvre l'acces au contenu de la fenêtre
  win.document.open();
  // Ici on devrait d'abord écrire une jolie entête HTML <head> ....
  // Mais on s'en passe et on affiche juste un message
  win.document.writeln ("<h1>Message secret</h1>");
  win.document.writeln ("Bonjour cher ami!");
  // On rajoute un bouton pour fermer cette fenetre
  win.document.writeln ("<hr><center><form><input type='button' value='FERMER'
onClick='window.close()'>");
  // Et on finalise, rien n'est affiché avant ce document.close () !!
  win.document.close();
}
....

<hr><form>
<input type="button" name="ProcessButton" value="Voir un truc"
       onClick="ouvrir()">
</form><hr>
```

Exemple 3 3 Création d'une nouvelle fenêtre II

Cet exemple est plus compliqué, il montre

- comment on peut faire un bouton permettant de fermer la fenêtre
- comment écrire une fonction un peu générale
- comment utiliser une variable pour "construire" le contenu (c'est idiot d'écrire pleins de xx.document.writeln)

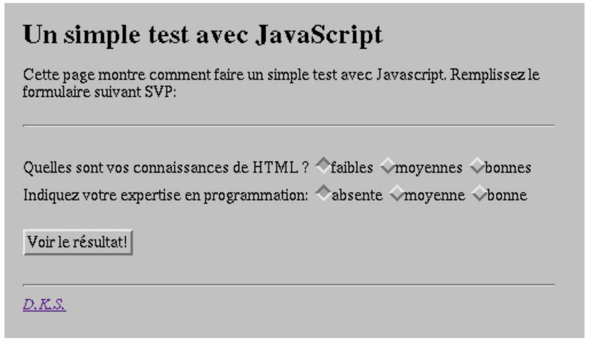
url: **Voir: http://tecfa.unige.ch/guides/js/exemples/createw.html**

4. Traitement de formulaires avec Javascript

4.1 Un simple quiz

Exemple 4 Simple quiz avec JavaScript

url: **cf. http://tecfa.unige.ch/guides/js/ex-intro/test1.html**



Un simple test avec JavaScript

Cette page montre comment faire un simple test avec Javascript. Remplissez le formulaire suivant SVP:

Quelles sont vos connaissances de HTML ? ○ faibles ○ moyennes ○ bonnes

Indiquez votre expertise en programmation: ○ absente ○ moyenne ○ bonne

D.K.S.

Voici le code:

```
<HTML>
  <HEAD>
    <TITLE>Un simple test avec JavaScript (18-Nov-1997)</TITLE>
    <!-- Created by: D.K.S., 18-Nov-1997 -->
<script language=JavaScript>
     // Initialisation de variables
      var q1 = 1;
      var q2 = 1;
    // calcul
      function calculer () {
      score = q1 + q2;
      alert("Sur une échelle qui va de 2 à 6 vous avez " + score)
      }
</script>
</HEAD>
  <BODY>
    <H1>Un simple test avec JavaScript</H1>
Cette page montre comment faire un simple test avec Javascript.
Remplissez le formulaire suivant SVP: <P>
<hr><form>
Quelles sont vos connaissances de HTML ?
<input type="radio" name="choice" value="1" onClick="q1=1" checked>faibles
<input type="radio" name="choice" value="2" onClick="q1=2">moyennes
<input type="radio" name="choice" value="3" onClick="q1=3">bonnes
<br>
Indiquez votre expertise en programmation:
<input type="radio" name="choice2" value="1" onClick="q2=1" checked>absente
<input type="radio" name="choice2" value="2" onClick="q2=2">moyenne
<input type="radio" name="choice2" value="3" onClick="q2=3">bonne
<P>
<input type="button" name="ProcessButton" value="Voir le résultat!"
       onClick="calculer()">
</form><hr>
</HTML>
```

A retenir:

- L’utilisation de "onClick":
  - Chaque fois que l’utilisateur clique sur un bouton radio, on enregistre la valeur du bouton dans une variable:

```
<input type="radio" name="choice" value="1" onClick="q1=1" checked>faibles
<input type="radio" name="choice" value="2" onClick="q1=2">moyennes
<input type="radio" name="choice" value="3" onClick="q1=3">bonnes
```

  - Notez que la variable "q1" a été définie préalablement dans le code
- L’appel à la fonction du calcul lors du submit:

```
<input type="button" name="ProcessButton" value="Voir le résultat!"
       onClick="calculer()">
```

- Le calcul:

```
 function calculer () {
      score = q1 + q2;
      alert("Sur une échelle qui va de 2 à 6 vous avez " + score)
      }
```

- Affichage des résultats
  - on utilise un simple "alert", ce qui n’est pas très beau.
  - Voir **http://tecfa.unige.ch/guides/js/ex-intro/test2.html !**
  - ou s’inspirer de l’exemple 3-2 “Création d’une nouvelle fenêtre I” [21]

A. Une méthode alternative pour traiter un formulaire

- Il n’est pas nécessaire d’enregistrer l’effet de chaque "click", mais on peut aussi avec JS interroger l’état d’un formulaire.
- C’est une solution plus difficile à comprendre pour les débutants, mais plus efficace au niveau écriture de code.
- Attention: testé avec NS 3/4 uniquement

Exemple 4 Interrogation d’un formulaire

url: **Voir: http://tecfa.unige.ch/guides/js/exemples/interactif2.html**

- à étudier soi-même :)

4.2 Vérification d’un formulaire I

- Voir: `http://tecfa.unige.ch/guides/js/ex-intro/test1-verif.html`

A retenir:

- Il y a pleins de façons de faire, voir les sites "WebMaster"
- Ici on fait simple, car on a peu de questions:
  - on initialise au départ les variables q1 et q2 à -1

```
var q1 = -1;
var q2 = -1;
```

  - on écrit une fonction qui teste si une de ces variables est encore sous 0

```
  function verif () {
    if ( (q1 < 0) || (q2 < 0) ) {
      alert ("Il faudrait remplir tous les champs SVP !");
      return false;
    }
    else return true;
}
```

  - notez que cette fonction retourne **`false`** si c’est le cas et **`true`** si tout va bien.
  - Ensuite, la fonction calculer appelle tout d’abord cette fonction verif. Si verif retourn false on sort, sinon on continue .....

```
  function calculer () {
    // Si verif retourn false on quitte le navire
    if (!verif()) return ;
    // sinon on continue
    score = q1 + q2;
    alert("Sur une échelle qui va de 2 à 6 vous avez " + score) ;}
```

- ... le reste est pareil

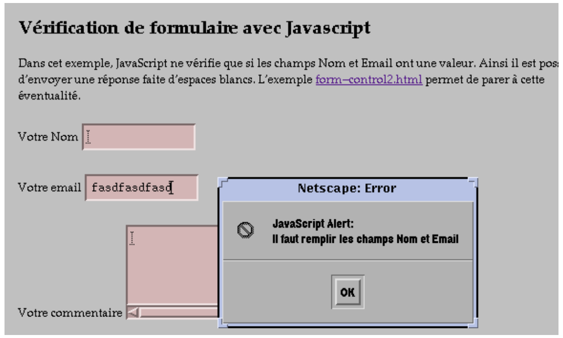
4.3 Vérification d'un formulaire (II)

Exemple 4-3 Vérification d'un formulaire

url: `http://tecfa.unige.ch/guides/js/ex-intro/form-control.html`

Cet exemple reprend le formulaire discuté dans le module "Les formulaires HTML" et y ajoute deux fonctions Javascript permettant de vérifier que l'utilisateur n'a pas laissé de champs vides.

- Beaucoup de sites utilisent JS pour vérifier un formulaire avant de l'envoyer à un CGI pour traitement.



Vérification de formulaire avec Javascript

Dans cet exemple, JavaScript ne vérifie que si les champs Nom et Email ont une valeur. Ainsi il est pos
d'envoyer une réponse faite d'espaces blancs. L'exemple form–control2.html permet de parer à cette éventualité.

Votre Nom

Votre email fasdfasdfasd

Votre commentaire

Netscape: Error

JavaScript Alert:
Il faut remplir les champs Nom et Email

A remarquer:

- Le bouton qui permet d'envoyer le contenu du formulaire est différent de l'exemple «Envoyer le contenu d'un formulaire avec email» [p. 21]:
  - Lorsque l'utilisateur clique sur le bouton "envoi" la fonction 'checkForm' est appelée avec le contenu du formulaire en argument: checkForm(this.form)

AVANT

```
<input type="submit" value="Envoi">
```

APRES (avec vérification):

```
<input type="button" value="Envoi"
        onClick="checkForm(this.form)">
```

- La fonction "checkForm" appelle la fonction "checkBlank", qui vérifie:
  - (1) est-ce que les champs en question ont une valeur (est-ce que l'utilisateur a tapé quelque chose).
  - Si la question (1) reçoit une reponse positive, alors la fonction checkBlank retourne la valeur true qui signifie dans notre cas, 'tout est en ordre'.
  - Dès lors le test dans la fonction checkForm est satisfait et l'appel form.submit() envoie le contenu du formulaire.
- A faire mieux:
  - La fonction checkBlank ne détecte pas si l'utilisateur a utilise un espace blanc dans sa réponse.
  - L'exemple 4-3 "Vérification d'un formulaire" [29] permet de parer à cette éventualité.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function checkBlank(input,msg)
{
  if (input.value == null || input.value.length == 0) {
    alert ("Il faut remplir les champs Nom et Email");
    return false;
  }
  return true;
}
function checkForm(form)
{
  if (
      !checkBlank(form.nom) ||
      !checkBlank(form.email)) {
    return false;
  }
  form.submit();
    alert ("Merci pour votre reponse ...");
  return true;
}
</SCRIPT>
</HEAD>
<BODY>
<h1>Vérification de formulaire avec Javascript</h1>
<form enctype="application/x-www-form-urlencoded"
  action="mailto:Patrick.Jermann@tecfa.unige.ch" method=post>
Votre Nom <input type="text" name="nom" size=15> <p>
Votre email <input type="text" name="email" size=15> <p>
Votre commentaire <textarea name="comment" rows=5 cols=30></textarea><p>
<input type="button" value="Envoi" onClick="checkForm(this.form)">
<input type="reset" value="Effacer">
</form>
</BODY></HTML>
```

A. Vérification d'un formulaire (III)

url: **Cf.** `http://tecfa.unige.ch/guides/js/ex-intro/form-control2.html`

- La fonction checkForm appelle la fonction checkBlank qui vérifie deux choses:

1) est ce que les champs en question ont une valeur (est-ce que l'utilisateur a tapé quelque chose).

2) Est-ce que les champs sont remplis par autre chose que des espaces ?

Si les questions 1) et 2) ont une reponse positive alors la fonction "checkBlank" retourne la valeur true qui signifie dans notre cas, 'tout est en ordre'. Dès lors le test dans la fonction chekForm est satisfait et l'appel form.submit() envoie le contenu du formulaire.

Voici la fonction checkBlank modifiée:

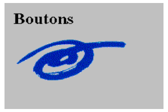
(voir la source de la page HTML ci-dessus pour l'ensemble)

```
function checkBlank(input,msg)
{
   if (input.value == null || input.value.length == 0) {
    alert ("Il faut remplir les champs Nom et Email");
    return false;
  }
  var str = input.value;
  for (var i = 0; i < str.length ;i++){
    var ch = str.substring(i,i+1);
    if ( ch == " ") {
      alert (msg );
      return false;
    }
  }
  return true;
}
```

5. HTML Dynamique (ou presque)

5.1 Les boutons JavaScript

url: **Cf. http://tecfa.unige.ch/guides/js/ex-intro/switch-buttons.html**



```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
off_logo = new Image(120,25);
off_logo.src = "logo.gif";
on_logo = new Image(120,25);
on_logo.src = "logo_on.gif";

function hiLite(imgDocID,imgObjName) {
      document.images[imgDocID].src = eval(imgObjName + ".src")
}
</SCRIPT>
</HEAD>
<BODY>
<A HREF="http://agora.unige.ch/"
     onMouseOver="hiLite(’logo’,’on_logo’)"
     onMouseOut="hiLite(’logo’,’off_logo’)">
<IMG SRC="newimages/logo.gif" BORDER=0  NAME="logo"></A>
</BODY></HTML>
```

5.2 Menus déroulants

- Faire un menu déroulant est assez difficile
- Il faut tenir compte des différents browsers
- Il existe pleins de sites qui donnent des scripts
  voir les Webmaster’s sites: http://tecfa.unige.ch/guides/toolbox.html

Exemple 5-1: Les HierMenus de webreference.com

- Un menu DHTML hiérarchique qui marche avec la plupart des clients WWW
  url: http://webreference.com/dhtml/hiermenus/
- Il faut bien lire les instructions (et surtout respecter les nom des array)
  url: http://webreference.com/dhtml/hiermenus/instructions/

Demonstration à TECFA

url: http://tecfa.unige.ch/navi/menu/about.html **(explications en Anglais)**
url: http://tecfa.unige.ch/navi/menu/toolmenu.html **(exemple)**

---

6. Javascript et DOM

..... à développer

Exemple 6-1: La visualisation des éléments de l’objet "document"

url: http://tecfa.unige.ch/guides/js/ex-intro/dombrowse.html
