

<h1 style="margin: 0;">CS WEB</h1>	<h1 style="margin: 0;">Ch 5 – JavaScript</h1>
--	---

I. INTRODUCTION.....	1
A. PRODUCTION DYNAMIQUE DE HTML	1
B. JAVASCRIPT, « SCRIPTING » COTE CLIENT, SCRIPTS EMBARQUES	1
C. JAVASCRIPT, LE LANGAGE	2
II. JAVASCRIPT EMBARQUE DANS LE HTML.....	3
A. ELEMENT HTML : SCRIPT.....	3
B. PROPRIETE DE GESTION DES EVENEMENTS	4
III. JAVASCRIPT – EXEMPLES D’UTILISATION	5
A. CONTROLER LES CHAMPS DE SAISIE : EVENEMENT ONKEYPRESS SUR L’ELEMENT INPUT	5
B. VERIFIER LES CHAMPS D’UN FORMULAIRE AVANT ENVOI : EVENEMENT ONSUBMIT SUR L’ELEMENT FORM.....	6
C. APPELEZ UN LIEN A PARTIR D’UNE LISTE DEROULANTE : EVENEMENT ONCHANGE SUR L’ELEMENT SELECT.....	6
D. AFFICHER UNE FENETRE D’AIDE	7
E. MODIFICATION DE L’ETAT D’UN FORMULAIRE	7
F. ECRITURE DE COOKIES	8
G. FAIRE COMMUNIQUER PLUSIEURS FENETRES : CALENDRIER PHP	9
H. ALLER PLUS LOIN...LIMITES DE JAVASCRIPT.....	11
IV. JAVASCRIPT – COMPLEMENTS	11

I. Introduction

A. Production dynamique de HTML

Nous avons vu l’intérêt du langage PHP pour construire des pages au contenu dynamique : l’accès aux SGBD permet effectivement d’afficher des catalogues de produits, effectuer des recherches, des classements, etc.

Le langage PHP permet également la vérification des champs de saisie des formulaires, et le renvoi de messages en fonction des erreurs rencontrées.

L’interaction nécessite l’envoi d’informations au serveur web, qui transmet à l’interpréteur PHP, ce dernier constituant la réponse en retour pour le navigateur.

Cette interaction est relativement « lourde » : d’où l’évènement du Scripting côté client.

B. JavaScript, « Scripting » côté client, scripts embarqués

Les langages de « Scripting » côté client permettent de rendre les interactions utilisateur-navigateur plus dynamiques. En effet le source du script est incorporé (on parle d’ « embarqué ») au sein de la page HTML.

Le moteur d’exécution du script (l’interpréteur du langage JavaScript) est incorporé au navigateur.

Alors que le langage PHP permet la production de page HTML dynamique, le langage JavaScript permet de dynamiser l'utilisation des pages du côté de l'utilisateur et évite ainsi les allers-retours avec le serveur .

Exemple de script embarquant du Javascript et intégrant du PHP :

```
<html>
  <head><title>PHP et JavaScript</title></head>
  <body>
<!-- ce script sera exécuté sur le poste client -->
<script langage="javascript">
var heure=new Date();
document.write ("<b>Poste client : il est " , heure.getHours() , ":" ,
heure.getMinutes(),"</b>") ;
</script>

<!-- ce script sera exécuté sur le serveur -->
<?php
echo "<br /><b>Serveur : il est " . date("H") . ":" . date("i") . "</b>";
?>
  </body>
</html>
```

L'exécution du script PHP renvoie, d'une part des balises HTML directement (y compris les balises relatives au script), d'autre part des balises HTML donnant l'heure côté serveur construites grâce à l'instruction PHP echo.

Au moment de l'affichage par le navigateur, ce dernier exécute les portions de script Javascript embarqué : c'est ainsi que l'heure côté client est affichée.



C. JavaScript, le langage

La syntaxe du langage JavaScript est très proche de Java ou C++. On peut noter cependant quelques différences fondamentales :

- Pas de typage des données : on ne spécifie pas de type à la déclaration d'une variable ; on peut donc stocker n'importe quoi dans une variable (les déclaration est introduite par le mot-clef **var** ou bien sans mot-clef, lors d'une affectation)
- Les fonctions sont introduites par le mot clef **function**, le type de retour n'est pas précisé (l'instruction **return** retournera ce qu'il faut...)
- **JavaScript n'a pas accès aux ressources locales du poste client** : il n'est pas possible d'écrire des fichiers, par exemple

- **Le code JavaScript a accès au contenu du document HTML qu'il peut donc modifier :** modification de contenu, modification de valeurs de propriétés (par exemple le style)

Il existe une spécification relative au Javascript : ECMAScript (ECMA pour European Computer Manufacturers Association, une association de normalisation européenne).

Cela constitue un ensemble de recommandations auxquelles il est utile de se conformer. Certaines implémentations de JavaScript offrent des fonctionnalités limitées à certains navigateurs, ce qui rend les scripts non portables.

II. Javascript embarqué dans le HTML

A. Élément HTML : script

La balise **script** permet

- La déclaration de variables globales
- La déclaration de fonctions qui seront appelées ultérieurement
- L'exécution de script à l'affichage de la page

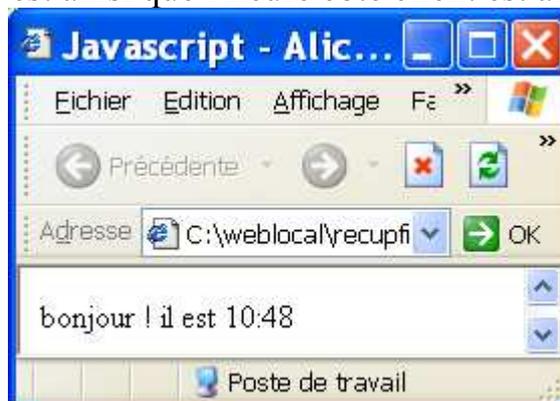
Exemple de script embarquant du Javascript:

```
<html>
  <head><title>Javascript</title>
<script langage="javascript">
var heure=new Date(); // variable globale
function afficherHeure() { // définition d'une fonction
  return (heure.getHours() + ":" + heure.getMinutes()); ;
} // fin afficherHeure
</script>
  </head>
  <body>

<script langage="javascript">
  document.write("bonjour ! il est ", afficherHeure() ) ;
</script>

  </body>
</html>
```

Au moment de l'affichage par le navigateur, ce dernier exécute les portions de script Javascript embarqué : c'est ainsi que l'heure côté client est affichée.



Le fichier source du script peut être stocké dans un fichier. On y fait référence dans la balise script :

```
<script langage="text/javascript" src="monscript.js">
</script>
```

B. Propriété de gestion des événements

Javascript permet de contrôler les événements se produisant dans le contexte d'un document. Ces événements sont :

<i>Propriété de gestion de l'évènement</i>	<i>description</i>
onabort	s'il y a une interruption dans le chargement
onbeforeunload	lorsque l'on ferme la fenêtre du navigateur web
onblur	à la perte du focus
onchange	à la perte du focus si la valeur a changée
onclick	sur un simple clic
ondblclick	sur un double clic
onerror	en cas d'erreur pendant le chargement de la page
onfocus	l'élément prend le focus (ou devient actif)
onkeydown	lorsque l'on appuie sur une touche
onkeypress	lorsque l'on reste appuyé sur une touche
onkeyup	lorsque l'on relâche une touche
onload	après la fin du chargement de la page
onmousedown	lorsque que l'on clique
onmousemove	lorsque l'on déplace la souris
onmouseout	lorsque la souris sort de l'élément
onmouseover	lorsque la souris est sur l'élément
onmouseup	lorsque l'on relache le bouton de la souris
onreset	quand on clique sur un bouton de type "reset" d'un formulaire
onresize	quand on redimensionne la fenetre
onselect	quand on sélectionne du texte
onsubmit	quand on clique sur un bouton de type "submit" d'un formulaire
onunload	quand on quitte la page en cours

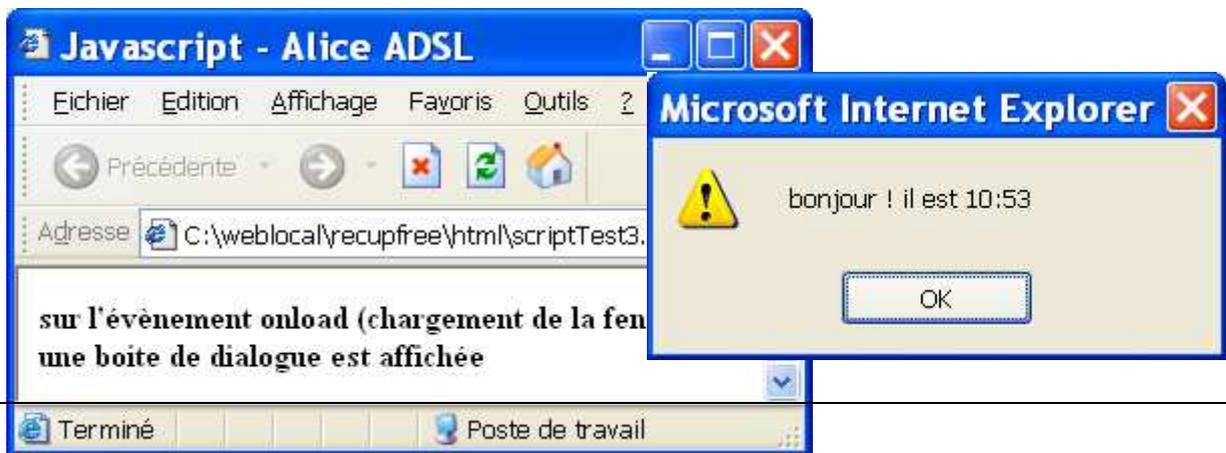
On peut associer l'appel de code Javascript au déclenchement de ces événements :

Exemple d'évènement onload:

```
<html>
  <head><title>Javascript</title>
  <script langage="javascript">
    var heure=new Date(); // variable globale
    function donnerLHeure() { // définition d'une fonction
      alert("bonjour ! il est " + heure.getHours() + ":" +
      heure.getMinutes()); ;
    } // fin afficherHeure
  </script>
  </head>
  <body onload="javascript: donnerLHeure();">
  <b>sur l'évènement onload (chargement de la fenêtre), une boîte de
  dialogue est affichée (alert)</b>
```

```
</body>
</html>
```

Au moment du chargement de la page dans le navigateur, le code associé à l'évènement est exécuté : c'est ainsi que la boîte de dialogue est affichée.



III. Javascript – exemples d'utilisation

A. Contrôler les champs de saisie : évènement onKeyPress sur l'élément input

Les champs de saisie peuvent devoir être contrôlés directement au moment de la frappe clavier.

Exemple de vérification de la saisie d'un champ obligatoire :

```
<html>
  <head><title>Javascript - tester champs numerique</title>
<script langage="javascript">

function verifierNum(e) {
  caractere = carLu(e) ;
  if ((caractere<"0" ) || (caractere>"9")) return false ;
  else return true ;
} // fin verifierNum

function carLu(e) {
  if (window.event) // pour IE
    return String.fromCharCode(window.event.keyCode) ;
  else // autres
    return String.fromCharCode(e.which) ;
} // fin carLu

</script>
</head>
<body >
  <form name="form1" action="trt.php" method="post">
    <input type="text" name="champsNum" onKeyPress="javascript: return
verifierNum(event);" />
    <input type="submit" name="btnEnvoyer" />
  </form>
</body>
```

```
</html>
```

B. Vérifier les champs d'un formulaire avant envoi : évènement onSubmit sur l'élément form

Afin d'éviter un aller-retour inutile, il est important de vérifier les formulaires avant leur soumission au serveur (il faudra absolument conserver des contrôles dans les scripts PHP : un utilisateur malintentionné pouvant très bien contourner les contrôles sur le poste client).

Exemple de vérification de la saisie d'un champ obligatoire :

```
<html>
  <head><title>Javascript - tester formulaire</title>
<script langage="javascript">
function verifierFormulaire() {

if (document.forms["form1"].elements["champs1"].value == "") {
    alert("zone obligatoire") ;
    return false ;
}
return true ;
} // fin verifierFormulaire
</script>
  </head>
  <body >
    <form name="form1"  action="trt.php" method="post "
onSubmit="javascript: return verifierFormulaire();">
      <input type="text"  name="champs1" />
      <input type="submit" name="btnEnvoyer" />
    </form>
  </body>
</html>
```

C. Appelez un lien à partir d'une liste déroulante : évènement onChange sur l'élément select

Une liste déroulante peut permettre la sélection d'un lien parmi une liste proposée et j'évènement onChange peut être utilisé pour appeler directement ce lien .

Exemple de vérification de la saisie d'un champ obligatoire :

```
<html>
  <head><title>Javascript - appeler un lien d'une liste</title>
  </head>
  <body >
    <form name="form1"  >
    <select name="langue" size="1"
onChange="window.location=eval("&quot;this.options[this.selectedIndex].value&quot;);"
    <option value="page_fr.htm"  selected="selected">français</option>
    <option value="page_en.htm">anglais</option>
    <option value="page_de.htm">allemand</option>
    <option value="page_es.htm">espagnol</option>
    </form>
  </body>
```

```
</html>
```

D. Afficher une fenêtre d'aide

Un script PHP est un fichier texte dont l'extension est généralement .PHP. Au sein de ce fichier source, on va trouver des balises HTML et du code PHP..

Les balises HTML sont directement réécrites dans le flux de sortie telles quelles.

Exemple de vérification de la saisie d'un champ obligatoire :

```
<html>
  <head><title>Javascript - tester fenetre aide</title>
  <script langage="javascript">
  var fenAide ;

  function fenetreAide() {
  fenAide = window.open("aide.html", "fenAide", "width=300, height=300,
  location=no, toolbar=no, scrollbar=yes, resizable=no, directories=yes,
  status=no") ;
  } // fin fenetreAide

  function fermerAide() {
  if ((fenAide=="undefined")||(fenAide.closed==true))
    { ; } // on pourrait ici la créer
  else
    fenAide.close();
  } // fin fermerAide
  </script>
  </head>
  <body >
    <a href="javascript: fenetreAide();">aide</a>
    <a href="javascript: fermerAide();">fermer l'aide</a>
    <form name="formAide">
      <input type="button" onClick="javascript: fenetreAide();" />
    </form>
  </body>
</html>
```

E. Modification de l'état d'un formulaire

JavaScript donne accès aux différents objets d'un formulaire

Exemple :

```
<html>
... une division dont le contenu est caché (style)
  <div id='aidechap' class="aid" style='display:none;'>...</div>
... un bouton pour changer le style et afficher le contenu
<input type='button' name='btnAid' size='30' value='aidez moi'
  onclick="javascript:document.all.aidechap.style.display=''">
```

Exemple :

```
<html>
... // des boutons radio
<input type='radio' name='fOk' value='1'>mauvaise reponse
```

```
<input type='radio' name='fOk' value='2'>mauvaise réponse  
<input type='radio' name='fOk' value='3'>bonne réponse  
<input type='radio' name='fOk' value='4'>mauvaise réponse  
...// un bouton pour obtenir la bonne réponse  
<input type='button' name='btnRep' size='30' value='obtenir la réponse'  
onclick="javascript: document.questions.fOk[2].checked=true;">
```

F. Ecriture de cookies

Les cookies sont un mécanisme permettant la mémorisation de données relatives à la navigation sur un site Web. Ces données pourront être utilisées lors d'une prochaine navigation, par exemple. Ce mécanisme utilise le stockage des données dans un fichier texte.

Exemple de création et de lecture de cookie :

```
<html>  
  <head>  
    <title>cookies</title>  
    <script type="text/javascript" >  
function setcookie(name,value) {  
  var days = 365;  
  var date = new Date();  
  date.setTime(date.getTime()+(days*24*60*60*1000));  
  var expires = "; expires="+date.toGMTString();  
  document.cookie = name+"="+value+expires+";";  
}  
  
function ecrireCookieVisite() {  
  setcookie('visite','oui') ;  
  return true;  
}  
  
function lireCookieVisite() {  
  deb = document.cookie.indexOf('visite' + "=")  
  if (deb >= 0) {  
    return true;  
  }  
  return false;  
}  
  
</script>  
</head>  
<body>  
  <div>  
    <script>  
      if (lireCookieVisite()==true) {  
        document.write("<h2>content de vous revoir</h2>");  
      }  
      else {  
        document.write("<h2>Bonjour et bienvenu</h2>");  
      }  
      ecrireCookieVisite();  
    </script>  
  </div>  
</body>  
</html>
```

G. Faire communiquer plusieurs fenêtres : calendrier PHP

Les fenêtres créées en Javascript sont connues du navigateur : il est ainsi possible de « faire communiquer » ces fenêtres et mettre à jour des champs d'une fenêtre à partir d'une autre fenêtre. Dans l'exemple qui suit, la « saisie » d'une date sera commandée par une fenêtre affichant un calendrier. La fenêtre affichant le calendrier est construite à l'aide d'un script PHP (le calendrier pourrait tout aussi bien être construit en Javascript)..

Exemple de page nécessitant la saisie d'une date :

```
<html>
  <head><title>Javascript - tester calendrier</title>
<script langage="javascript">
var fenCal ;

function ouvrirCalend() {
fenCal = window.open("calendrier.php", "fenCal", "width=200, height=300,
location=no, toolbar=no, scrollbar=yes, resizable=no, directories=no,
status=no") ;
} // fin ouvrirCalend

function fermerCalend() {
if ((fenCal=="undefined")||(fenAide.closed==true))
  { ; }
else
  fenCal.close();
} // fin fermerCalend

</script>
</head>
<body onUnLoad="javascript: fermerCalend();">
  <form name="formDate">
    <input type="text" name="jour" size=2 disabled />
    <input type="text" name="mois" size=2 disabled />
    <input type="text" name="annee" size=4 disabled />
    <input type="button" value="ouvrir le calendrier"
onClick="javascript: ouvrirCalend();"
    <input type="button" value="fermer de calendrier"
onClick="javascript: fermerCalend();"
  </form>
</body>
</html>
```

Exemple (à améliorer) de page PHP permettant d'afficher le calendrier et JavaScript permettant de récupérer la date sélectionnée dans un champ de formulaire :

```
<html>
  <head><title>Calendrier PHP</title>
<script langage="javascript">

function recopierDates(jour, mois, annee) {
  opener.document.formDate.jour.value=jour;
  opener.document.formDate.mois.value=mois;
  opener.document.formDate.annee.value=annee;
} // fin recopierDates
```

```

</script>
</head>
<body>
<?php
$lesMois=array("janvier","fevrier","mars","avril","mai","juin","juillet",
"aout","septembre","octobre","novembre","decembre");

$date=getdate() ;

$mois=$date["mon"] ;
$annee=$date["year"] ;

if ((isset($_POST["mois"]))&&(isset($_POST["annee"]))) {
    $mois=$_POST["mois"] ;
    $annee=$_POST["annee"] ;
}
else { ;
}
$moisSuiv=$mois+1 ;
$anneeSuiv=$annee ;
if ($moisSuiv>12) {
    $moisSuiv=1 ;
    $anneeSuiv++ ;
}

echo $lesMois[$mois-1] . "<br />" . $annee;
?>
<form method="post">
    <input type="hidden" name="mois" value="<? echo $moisSuiv ; ?>" />
    <input type="hidden" name="annee" value="<? echo $anneeSuiv; ?>" />
    <input type="submit" value="mois suivant"
</form>
<?php
genCalendrier($mois, $annee) ;
?>

<?php
function genCalendrier($mois, $annee) {

$premJourMois=getdate(mktime(0,0,0,$mois,1,$annee)) ;
$jourSemPremier=$premJourMois["wday"] ;

$maxSem=5 ;

echo <<< EOS
<table>
<tr><th>L</th><th>M</th><th>M</th><th>J</th><th>V</th><th>S</th><th>D</th></tr>
EOS ;
$test=0;
for ($numSem=0 ;$numSem<=$maxSem ;$numSem++) {
    echo "<tr>";
    for ($jour=1 ;$jour<=7 ;$jour++) {
        $jourDuMois=1-$jourSemPremier+$numSem*7+$jour;
        $jj=getdate(mktime(0,0,0,$mois,$jourDuMois,$annee));
        $leJour=$jj["mday"];
        $leMois=$jj["mon"];
        $lAnnee=$jj["year"];
    }
}
}

```

```
    if ($leJour==1) { // si le jour calculé est 1, on est au premier jour
        $test++;
        if ($test>=2) $maxSem--; // on arrive à la dernière semaine du mois
    }
    if ($test==1) { // à partir du premier jour du mois
        echo "<td>";
        echo "<a href=\"javascript: recopierDates('$leJour', '$leMois', '$lAnnee')
;\">$leJour</a>";
        echo "</td>";
    }
    else {
        echo "<td></td>";
    }
}
echo "</tr>";
}
echo "</table>";
} // fin fonction genCalendrier
?>
</body>
</html>
```

H. Aller plus loin...limites de Javascript

JavaScript est un langage à part entière et on peut construire des applications riches côté client : cependant il est nécessaire de prendre en compte :

- La lourdeur des applications ainsi produites : l'objectif des applications web est de fournir une interface simple (client léger), le traitement étant pris en compte par un code coté serveur.
- Les implémentations non standard de JavaScript amènent une difficulté de rendre ces applications portables d'un navigateur à un autre.
- **L'exécution de JavaScript n'est possible que si cette fonction est activée dans les paramètres du navigateur**

IV. JavaScript – compléments

Les technologies Web 2.0 font usage du JavaScript pour construire des applications Web très interactives.

L'ensemble des technologies qui constitue Ajax (Asynchronous JavaScript and XML) semble avoir un bel avenir : il permet l'appel de code PHP pour recevoir en réponse des données XML qui vont servir à modifier l'interface de l'utilisateur sans pour autant changer de page Web.