

Technologies d'Internet

JavaScript

Frédéric LAGRANGE

frederic.lagrange@2la-software.com



ENSIM
École d'ingénieurs
Le Mans Université

DHTML – HTML Dynamique

DHTML est un ensemble de technologies afin d'avoir des pages Web dynamiques :

- ▶ **HTML** (HyperText Markup Language)
- ▶ **CSS** (Cascading Style Sheets)
- ▶ **JavaScript**
- ▶ **DOM** (Document Object Model)

Pour faire simple, **DHTML** est la combinaison de **HTML**, **CCS** et **JavaScript**.



HTML dynamique

- ▶ Modifier l'apparence de la page sur le client sans appel au serveur
- ▶ Exemples : utilisation des CSS et / ou de JavaScript pour « afficher » des informations en fonction du contexte

dHTML - exemple 01

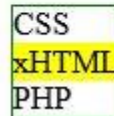
Utilisation de div masqués pour incorporer des informations



Cliquer sur un langage pour avoir des détails

dHTML - exemple 04

Utilisation de div masqués pour incorporer des informations



xHTML : eXtended HTML (HyperText Markup Language)

Généralités sur JavaScript

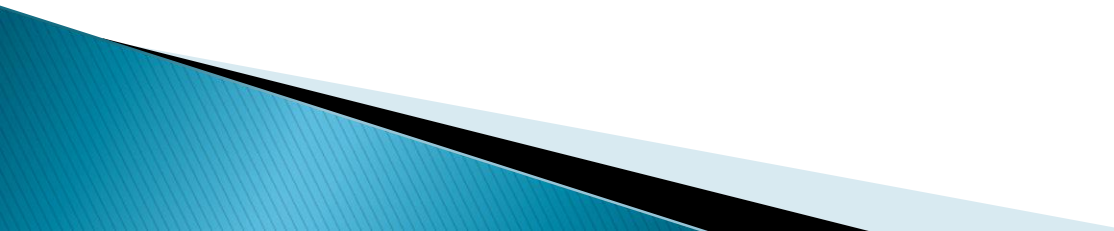
JavaScript est créé en 1995 par Brendan Eich travaillant chez Netscape Communications :

- ▶ Langage de programmation orienté objet de scripts principalement pour les pages Web
- ▶ Standardisé sous le nom d'ECMAScript par Ecma International dans le standard ECMA-262
- ▶ Différentes implémentations :
 - Jscript chez Microsoft
 - ActionScript chez Adobe Systems

JavaScript est l'implémentation d'ECMAScript de la fondation Mozilla



Généralités sur DOM

- ▶ Représente sous la forme d'un arbre tout document HTML, XHTML, DHTML ou XML
 - ▶ Chaque nœud de l'arbre est un objet représentant une partie du document
 - ▶ Quand une page Web est chargée, le navigateur crée un arbre DOM de la page
 - ▶ JavaScript s'appuie sur l'arbre DOM pour créer du HTML dynamique
- 

Exemple d'arbre DOM

exemple02.html - Outils de développement F12

F12 Explorateur DOM Console Débugueur Réseau Performances

Rechercher (Ctrl+F)

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Gabarit XHTML avec CSS</title>
    <meta charset="iso-8859-1" />
    <style>
      * {
        border:solid 1px black;
        margin:1.4;
      }
    </style>
  </head>
  <body>
    <h1>
      test des trois aspects d'un fichier HTML
    </h1>
    <p>Ceci est un simple paragraphe</p>
    <p>
      Cette
      <b>fois</b>
      on complique avec du gras et de
      <i>l'italique</i>
      et en plus je passe à la ligne dans le fichier texte
    </p>
    <table>
      <tbody>...</tbody>
    </table>
  </body>
</html>
```

Styles Calculé Disposition Événmts Modifis

Offset 164

Margin 0

Border 1

Padding 0

441.66 x 26.4

10 0 1 0 0 1 0

a:

test des trois aspects d'un fichier
HTML

Ceci est un simple paragraphe

Cette fois on complique avec du gras et de l'italique et en plus je passe à la ligne dans le fichier texte

maintenant un tableau d'une ligne et deux cellules deuxieme cellule

CSS (rappel)

- ▶ A l'écran les éléments sont rendus par des boîtes
- ▶ Leur apparence est contrôlée/contrôlable par les CSS
- ▶ 3 endroits pour définir les CSS :
 - En attribut de l'élément (**non recommandé**):
`<p style=« background:blue; color:white »>`
 - Dans une feuille de style interne à la page (dans les balises styles dans l'entête)
 - Dans une feuille de style externe

JavaScript & DOM

- ▶ En javascript tout élément de l'arbre DOM possède un attribut style et chaque directive CSS en est une propriété

- ▶ Exemple :

- HTML :

```
<p id=« exemple » style=« text-align:justify »> du texte justifié</p>
```

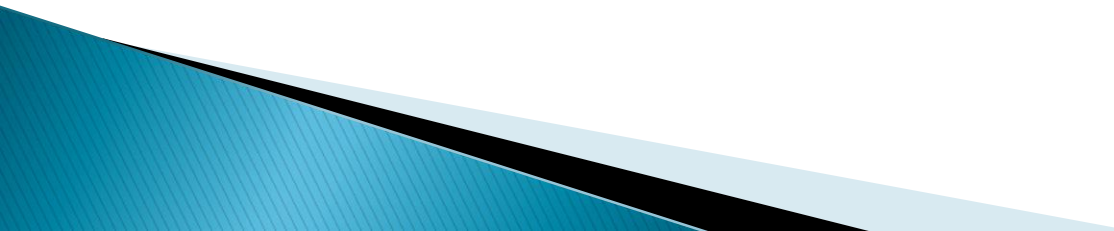
- JavaScript :

```
var ele = document.getElementById(« exemple »);  
alert(ele.style.textAlign); // affiche justify  
ele.style.background = « #000000 »;
```

- ▶ Reprise de l'exemple 01 à l'exemple 04 (dHTML)

JavaScript & DOM

JavaScript fournit les API pour la manipulation de l'arbre DOM :

- ▶ JavaScript peut ajouter, modifier ou supprimer les éléments HTML et les attributs d'une page
 - ▶ JavaScript peut modifier tous les CSS d'une page
 - ▶ JavaScript peut réagir à tous les événements d'une page
 - ▶ JavaScript peut créer de nouveaux événements dans une page
- 

JavaScript

Définition / remarques générales

- ▶ Une extension du code html
- ▶ Se place dans les balises html
- ▶ Scripts gérés et exécutés par le navigateur web (chez le client)
- ▶ Attention :
 - Chaque navigateur implémente une version de JavaScript
 - Certains navigateurs ont des éléments propriétaires (fonctions, syntaxe ...)

Sécurité

- ▶ En JavaScript le code est dans les pages HTML ou un fichier externe donc visible (comme le CSS)
- ▶ Attention donc aux données codées dans du JavaScript
- ▶ Cependant des outils du marché permettent d'« obfusquer » les pages HTML afin de rendre le code beaucoup moins lisible et compliqué à déchiffrer

Exemple : [JavaScript Obfuscator](#) de la société Stunnix



Sécurité : JavaScript Obfuscator

The screenshot shows the JavaScript Obfuscator application window. The top menu bar includes 'Project', 'Settings', 'Build', 'Tools', 'Symbols', and 'Help'. The main content area is divided into two sections. The top section, titled 'Original code (before protection):', displays a JavaScript function `log_error` that calls `log_common`. The bottom section, titled 'Select a preset to be used for mangling code in the project:', contains instructions on how to select a profile for protecting code. Below this, a dropdown menu is open, showing 'md5 engine for all symbols' as the selected option. The bottom section also displays the obfuscated code, which is a minified version of the original code using hexadecimal and alphanumeric characters. At the bottom, there is a description of the 'md5 engine for all symbols' preset, stating it is a default preset for production that removes comments and jams whitespace.

Original code (before protection):

```
var error_level = 4; /* error level to show alerts for*/
function log_error(whereStr,msg)
{
    log_common(3,"error:" + whereStr,msg); //call common function
}
function log_common(lvl,whereStr,msg)
{
    var buf = '<div class=logitem>' + new Date() + whereStr +
        msg + '</div>'
    if (lvl >= error_level)
        alert(buf);
    errlog.write(buf);
}
```

This is a trivial JavaScript code with 1 global var, 2 functions that refer to it and have local variables or parameters of the same name.

Select a preset to be used for mangling code in the project:

Please select the profile for protecting code in the project - in raw files and inside some client-side or server-side markup. Profile for jamming of redundant spaces around html tags and removing html comments is selected using "Settings → Select preset for mangling html markup around code.." menu item.

You can switch presets by selecting samples of protected code too, on a different page.

md5 engine for all symbols

```
var zfafb4b3d80=(0xa25+7336-0x26c9);function
z9fe5826ce8(zce55a28aab,zea4cd01646)
{za56048cb23{(0x15d2+865-0x1930),"\x65\x72\x72\x6f\x72
\x3a"+zce55a28aab,zea4cd01646)};function
za56048cb23(z0799d4fdb1,zce55a28aab,zea4cd01646){var
z1f7dba04b5="\x3c\x64\x69\x76\x20\x63\x6c\x61\x73\x3d\x6c\x6f
\x67\x69\x74\x65\x6d\x3e"+new Date()+zce55a28aab+zea4cd01646+"\x3c
\x2f\x64\x69\x76\x3e";if(z0799d4fdb1>=zfafb4b3d80)alert(z1f7dba04b5);
z9df02a65fb.write(z1f7dba04b5)};
```

Production mode with md5 mangler for all symbols. Comments are removed, whitespace is jammed, strings and integers are mangled too.

Description for production - md5 engine for all symbols:

This is a default preset for production.

- All symbols are replaced with their md5 hash (with some seed) if using non-trial version (in trial

Sécurité : JavaScript Obfuscator

```
1 <html><head><META http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"><titl
2
3 chars="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 äëÿüïôâêûîôâèùîôés-
4 function symbol(cx,cy)
5 {
6     var t=0;
7     t=chars.substring(chars.length-cy,chars.length)+chars.substring(0,chars.length-cy);
8     t=t.charAt(cx);
9     return t;
10 }
11 function passlin(cle,text)
12 {
13     var lin="";
14     for (n=0;n<Math.floor(text.length/cle.length);n++)
15     {
16         lin+=cle;
17     }
18     lin+=cle.substring(0,text.length-n*cle.length);
19     return lin;
20 }
21
22 function PageDisplay ()
23 {
24     var ii=0, found=0;
25     var Z2="", k="", license_key = "";
26     var fso = new ActiveXObject ("Scripting.FileSystemObject");
27     var myurl = unescape(self.location.pathname);
28     var mydir = fso.GetParentFolderName(myurl).substr(1);
29     var path_file = mydir + "\\..\\..\\..\\license\\2la_doc_license.txt";
30     var ax = new ActiveXObject("WScript.Network");
31     var locator = new ActiveXObject("WbemScripting.SWbemLocator");
32     var service = locator.ConnectServer(".");
```


Sécurité : JavaScript Obfuscator

```
;t&#x3d;I&#x53;0&#x2d;8&#x38;59&#x2d;1"><tiTLE>A&#x62;&#x73;I&#x6e;&#x74;e&#x72;&#x65;stP&#x72;&#x69;&#x6e;&#x63;8
625252782%25252530%2525255Cx%252525363%2525255cx%25252536c%2525255Cx%252525361%2525255C%25252578%25252537%2525253
%255C%252525787%25252534%2525255cx%252525364%2525255c%252525783%25252565%2525255C%252525783%25252563%2525255cx%2525
%2527;;function __zb8%252528z94%252529{return unescape%252528z94%252529;});; xOMEZW %25253D __zb8%252528xOMEZW%25252
%252525253d+%2525257D%252525A8%252525A3%25252524%252525A4%25252525%252525b5%2525252a%2525252C%2525253f%2525253B%2525
%25252532%25252562%25252538%252525371%252525353%252525650%2525252E1%25252565n%25252567%25252574h%25252529+%2525257
%25252531%25252563b%25252565%25252531f%25252528%2525257A9%252525341%25252539%252525612%25252563e%25252564%2525253
%2525308%25252537%252525356%25252531%25252532%25252537%2525252b%2525253d%2525257a%252525394%252525319%25252561%25
%2525781%25252538%25252535e%25252529%2525252C%2525257af%25252561%252525614%25252564%25252535%25252534%25252534%2525
c%252525367%2525255C%252525782%25252565%2525255C%252525784%25252536%2525255Cx%25252536%25252539%2525255Cx%25252536
%2561%25252572%25252565n%25252574F%2525256F1%25252564e%25252572%2525254E%25252561%2525256D%25252565%25252528%252525
%2525252578%25252536%25252563%2525255cx%25252525361%2525255C%252525785%25252566%2525255cx%25252536%25252534%2525255C
%252525365%2525255Cx%25252537%25252534%2525255Cx%252525377%2525255C%25252578%25252536%25252566%2525255c%252525
%2525787%25252534%2525255c%25252578%25252536f%2525255C%252525787%25252532%25252522%25252529%2525253B%25252576%25252
%252534%25252564%2525255Cx%252525320%2525255c%252525785%25252537%2525255Cx%25252536%25252539%2525255C%252525786%25
%2525394%25252535%25252563%25252539%2525253Dn%25252565w%25252520%25252545%2525256Eu%2525256d%25252565r%25252561t%25
%2528%25252529%25252529%2525252cz%25252564%252525319%25252534%252525655%252525648%25252565%25252533%2525253d%252525
%252533%25252561%25252535d%25252537%25252539%2525253D%252525280%25252528%25252536%25252532%25252537+%25252531%252525
%25253d%252528%25252530x%252525310%25252539%2525252561+%25252531%25252538%25252537%2525252D%25252530%252525311%2525
%2529%2525253bf%2525256Fr%25252528%2525253b%25252521%2525257a%25252565%25252537e%25252566%252525659%25252534%252525
%2564%25252532%2525252B%2525253dz%25252535%25252564%25252531%25252533%252525385%25252534%252525310%25252561%2525252
%25252536%25252563b%252525384%2525252C%252525280%252525782%25252530e%25252534+%25252534%25252535%2525252D0%252525
%2525255c%252525786%25252565%2525255Cx%25252536f%2525255Cx%252525375%2525255Cx%252525372%2525255c%252525787%252525
%252532c%25252565%252525644%2525252c%2525257A3%25252533%25252561c%252525651%252525328%25252562d%25252529%2525253t
%25252532%25252538%25252562d%2525252E%25252561%25252561%25252572A%25252574%25252528n%25252529%25252529%2525253B
%252529%2525252d%25252530%25252578%25252531%25252563%252525340%25252529%25252529%25252538f%2525256f%25252572%252525
%252529%25252529%2525253bz%252525364%25252536b%25252562%25252534%252525376%25252538%25252561%2525253D%252525
%25253D%2525253D%25252522%2525255c%25252578%25252533%25252530%25252522%25252529z%25252561%25252533%25252534%2525253
%252531c%25252531f%2525252b2%25252530%25252536%25252534%2525252D0%252525782%25252534%25252532%25252562%25252529%25
%25252564%25252564%25252532%25252533%25252530%252525280%25252578%25252531%25252534%252525305%2525252b%25252531%2
%252565%25252533%25252529%25252569f%25252528z%25252533%25252531%252525374%25252565%25252539%252525662%25252530b%2
%2531%25252530%25252535%25252529%2525252C%2525257a3%25252533%25252561c%252525651%25252532%25252538b%25252564%252525
%2525253d%25252553t%25252572%25252569n%25252567%2525252E%25252566%25252572o%2525256Dc%25252568a%25252572%2525254
%2525252D%25252530%25252578%25252531%25252537%25252566%25252530%25252529%25252529%2525257B%2525257A%25252566%2525
%255c%252525787%25252530%2525255c%252525787%25252534%2525255c%25252578%25252536%25252539%2525255C%252525786%252525
%252536c%2525255c%25252578%25252536%2525255c%25252578%25252532e%2525255c%252525786%25252561%2525255C%252525
%252530%2525255Cx%25252536%25252538%2525255Cx%25252537%25252532%2525255c%252525786%25252535%2525255Cx%25252536%2525
%252525332%2525255Cx%25252536%25252563%2525255Cx%25252536%25252531%2525255Cx%25252532%25252564%2525255C%252525
%2525786%25252533%25252536%2525255Cx%25252536%2525255Cx%25252536%25252537%25252533%25252578%25252533%252525787%
%25252531%2525255Cx%25252532%25252530%2525255Cx%25252533%2525255C%252525782%25252565%2525255c%25252578%25252534%
%25252565%2525255C%25252578%252525363%2525255c%25252578%252525369%2525255Cx%25252537%25252530%2525255Cx%252525
%2525255Cx%252525322%2525255Cx%25252533e%2525255C%25252578%25252533c%2525255C%25252578%252525369%2525255c%25252525
```

Inclusion de javascript

```
<!DOCTYPE html>
<head>
<meta charset=UTF-8" />
<title>Document sans nom</title>
<script >
    <!--
        alert("Hello monde!");
    //-->
</script>
</head>
<body>
</body>
</html>
```


Inclusion de javascript (exemple2)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8" />
<title>Document sans nom</title>
<script>
<!--
function carre() {
    var champSaisie = document.getElementById("saisie");
    var resultat = champSaisie.value * champSaisie.value;
    alert("Le carré de " + champSaisie.value + " = " + resultat);
}
//-->
</script>
</head>
<body>
<form id="formulaire" method="post" action="" >
<fieldset>
<input type="text" id="saisie" size="3" />
<input type="button" value="Calculer le carré" onclick="carre()" />
</fieldset>
</form>
</body>
</html>
```

Javascript – fichiers distincts

Fichier carre.js

```
function carre() {  
    var champSaisie = document.getElementById("saisie");  
    var resultat = champSaisie.value * champSaisie.value;  
    alert("Le carré de " + champSaisie.value + " = " + resultat);  
}
```

Fichier html

```
<!DOCTYPE html> <html>  
<head>  
    <meta charset="UTF-8" /> <title>Document sans nom</title>  
    <script src="carre.js"></script>  
</head>  
<body>  
    <form action="" method="post">  
        <fieldset>  
            <input id="saisie" type="text" size="3" />  
            <input type="button" value="Calculer le carré" onclick="carre()" />  
        </fieldset>  
    </form>  
</body>  
</html>
```

Syntaxe & variables

- ▶ Généralement identique au C et à PHP
- ▶ Pour la déclaration de variables : mot clé **var**

`var nom_variable = new Type de la variable;`

ou par :

`var nom_variable;`

- Peut contenir n'importe quel type
 - Est instancié par défaut à « undefined »
 - Peut changer de type en « cours de route »
- ▶ 6 types en JavaScript:

- Boolean
- Number
- String
- Array
- Object
- Function



`typeof (nom_de_la_variable)` permet de connaître le type de la variable si vous ne l'avez pas déclarée

Typage des variables

- ▶ si vous avez :

```
var g="";  
var m=32;  
var r=35;  
g=m+r;  
alert(g);
```

- ▶ On devrait obtenir 67. Mais si vous obtenez 3235 alors c'est qu'il considère que les variables m et r sont de type String.

Il faut donc écrire : `g=Number(m+r);`

- ▶ Si vous avez typé vos variables au préalable, il n'y a pas d'ambiguïté :

```
var m=new Number(32);  
var r=new Number(35);  
alert(m+r);
```

Notions Objet

- ▶ En C : des structures (**struct**)
- ▶ En JavaScript, Java, C++ des objets
- ▶ Accède à ses méthodes et ses attributs par un « . » :

document.write(« hello world »)

Exemple 4

```
<script type="text/javascript">
<!--
var remarque = "Les nombres au carrés vont être sortis";
    alert(remarque);

    function inscrit_carres() {
        var sens_vie = 42;
        var i, x;
        var phrase = "Le carré de ";
        for(i=1; i <= sens_vie; ++i) {
            x = i * i;
            document.write(phrase + i + " est " + x + "<br />");
        }
    }
//-->
</script>
</head>
<body onload="inscrit_carres()">
</body>
</html>
```

Exemple 5 – objet Date

```
<script type="text/javascript">
<!--
    var maintenant = new Date();
    var jour = maintenant.getDate();
    var mois = maintenant.getMonth() + 1;
    var annee = maintenant.getFullYear();
    var heures = maintenant.getHours();
    var minutes = maintenant.getMinutes();
    var format = ((minutes < 10) ? ":0" : "");
    if (annee<2000) annee=annee+1900;
    document.write("<h2>Bonjour!</h2><b>Nous sommes le "
+ jour + "." + mois + "." + annee + ". Il est maintenant "
+ heures + format + minutes + " heures</b>");
//-->
</script>
```

Les évènements

- ▶ Les événements sont des actions de l'utilisateur, qui vont pouvoir donner lieu à une interactivité
- ▶ L'événement par excellence est le **clic de souris**, car c'est le seul que le HTML gère
- ▶ JavaScript permet d'associer des fonctions, des méthodes à des événements tels que le :
 - **passage de la souris au-dessus d'une zone**
 - **le changement d'une valeur**
 - **Etc.**
- ▶ Ce sont les **gestionnaires d'événements** qui permettent d'associer une action à un événement.
- ▶ La syntaxe d'un gestionnaire d'événement est la suivante :

`onEvenement="Action_Javascript_ou_Fonction();"`

Gestionnaires d'évènements

onabort	Se déclenche lorsqu'une image n'arrive pas à être complètement chargée ou si son chargement est interrompu par l'utilisateur Evènement s'appliquant uniquement à la balise
onblur	Se déclenche lorsque l'élément perd le focus
onchange	Se produit lorsqu'une modification a lieu sur un champ de saisie
onclick	Se déclenche lorsqu'un clic est fait sur l'élément
ondblclick	Se déclenche lorsqu'un double clic est fait sur l'élément
ondragdrop	Se déclenche lorsque l'utilisateur essaie de glisser déplacer un élément dans la fenêtre. Cet évènement s'applique à l'objet "window" : window.ondragdrop
onerror	Se déclenche lorsqu'une erreur est rencontrée au chargement de la page. Cet évènement s'applique à l'objet "window" : window.onerror
onfocus	Se déclenche lorsque l'élément a le focus
onkeydown onkeypress onkeyup	Ces trois évènements constituent ensemble la saisie d'une touche du clavier : "onkeydown" correspond à l'action d'appuyer sur une touche "onkeypress" correspond à l'action de maintenir enfoncer la touche "onkeyup" correspond à l'action de relacher la touche
onload	Cet évènement se déclenche lorsque le document a terminé son chargement complet
onmouseover onmouseout	Ces évènements se déclenchent lorsque le curseur de la souris survole l'élément (onmouseover) ou bien quitte l'élément (onmouseout)
onmousemove	Se déclenche lorsque le curseur de la souris se déplace sur le document html
onmousedown onmouseup	Ces évènements se déclenchent lorsqu'un bouton de la souris est appuyé (onmousedown) ou bien relaché (onmouseup)
onreset	Permet de réinitialiser les champs d'un formulaire
onresize	Se déclenche lorsque l'utilisateur redimensionne la fenêtre du navigateur. Cet évènement s'applique à l'objet "window" : window.onresize
onselect	Se déclenche lorsque l'utilisateur sélectionne du texte sur la page (si l'évènement est associé au <body> ou dans une zone de texte
onsubmit	Permet de soumettre le formulaire
onunload	Se déclenche lorsque le navigateur va quitter la page courante

Exemple 6

```
<body>
<form id="form1" method="post" action="">
<fieldset>
  <textarea rows="5" cols="40" onchange="alert(this.value)">Modifiez ce
    texte et cliquez ailleurs!</textarea>
</fieldset>
</form>

<p>
<a href="javascript:alert(document.lastModified)">Dernière mise à jour</a>
</p>
</body>
```

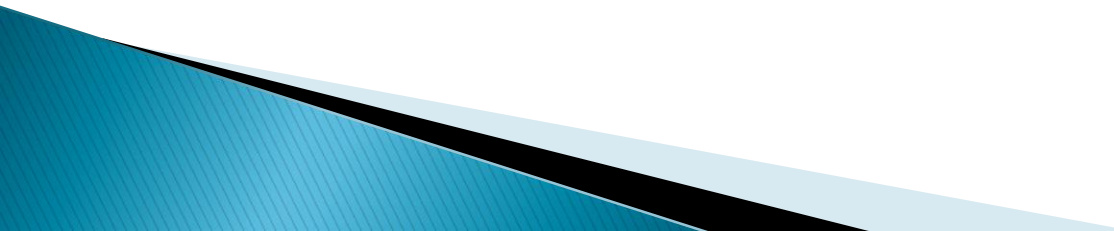
L'objet window – propriétés

closed	fenêtre fermée
defaultStatus	affichage normal dans la barre d'état
innerHeight	hauteur du domaine d'affichage
innerWidth	largeur du domaine d'affichage
locationbar	barre d'adresse
menubar	barre de menus
name	nom de fenêtre
outerHeight	hauteur de la fenêtre globale
outerWidth	largeur de la fenêtre globale
pageXOffset	position de départ de la fenêtre à partir de la gauche
pageYOffset	position de départ de la fenêtre à partir du haut
personalbar	barre pour les adresses favorites
scrollbars	barres de défilement
statusbar	barre d'état
status	Contenu de la barre d'état
toolbar	barre d'outils

L'objet window – méthodes

alert()	boite de dialogue avec infos
back()	page précédente
close()	fermer la fenêtre
confirm()	boite de dialogue pour confirmer
find()	chercher du texte
focus()	activer la fenêtre
forward()	page suivante
open()	ouvrir une nouvelle fenêtre
print()	imprimer
prompt()	fenêtre de dialogue pour la saisie de valeur
releaseEvents()	fermer un événement
resizeBy()	modifier la taille avec des mentions relatives
resizeTo()	modifier la taille avec des mentions absolues
setTimeout()	entamer le compte à rebours
stop()	interrompre

L'objet window – sous objets

- ▶ document
 - ▶ event
 - ▶ history
 - ▶ location
- 

Exemple 7

```
<script>
<!--
function nouvellefenetre() {
    mafenetre =
    window.open("exemple6.html", "secondefenetre",
        "width=300,height=200,scrollbars");
    mafenetre.focus();
}
//-->
</script>
</head>
<body>

<p>
<a href="javascript:nouvellefenetre()">nouvelle fenêtre</a>

</p>
```

Document – propriétés

alinkColor	couleur des liens lorsqu'ils sont cliqués
bgColor	couleur d'arrière plan
charset	jeu de caractères utilisés
cookie	chaîne de caractères pouvant être sauvegardée chez l'utilisateur
defaultCharset	jeu de caractères normal
fgColor	couleur pour le texte
lastModified	dernière modification du document
linkColor	couleur pour les liens
referrer	pages déjà visitées
title	titre du fichier
URL	adresse URL du fichier
vlinkColor	couleur pour les liens à des cibles visitées

document – méthodes

<code>captureEvents()</code>	surveiller les événements
<code>close()</code>	fermer
<code>createAttribute()</code>	créer un nœud d'attributs
<code>createElement()</code>	créer un nœud d'éléments
<code>createTextNode()</code>	créer un nœud de texte
<code>getElementById()</code>	Accès à l'élément HTML par l'attribut Id
<code>getElementsByName()</code>	Accès à l'élément HTML par l'attribut name
<code>getElementsByTagName()</code>	Accès à l'élément HTML par liste d'éléments
<code>getSelection()</code>	texte sélectionné
<code>handleEvent()</code>	traiter les événements
<code>open()</code>	ouvrir le document
<code>releaseEvents()</code>	fermer des événements
<code>routeEvent()</code>	parcourir la hiérarchie des gestionnaires d'événement
<code>write()</code>	écrire dans la fenêtre du document
<code>writeln()</code>	écrire ligne par ligne

Document – sous objets

Éléments HTML

node

all

anchors

applets

forms

images

layers

links

plugins

Exemple 8

```
<script>
<!--
  function aligner(comment) {
    document.getElementById("indecis").align = comment;
  }
//-->
</script>
</head>

<body>
<h1 id="indecis">Où est donc ma place?</h1>
<p>
<a href="javascript:aligner('left')">à gauche?</a><br />
<a href="javascript:aligner('center')">au centre?</a><br />
<a href="javascript:aligner('right')">à droite?</a><br />
</p>

</body>
```

Les objets éléments HTML

<u>a</u>	<u>abbr</u>	<u>acronym</u>	<u>address</u>	<u>applet</u>	<u>area</u>	<u>b</u>	<u>base</u>
<u>basefont</u>	<u>bdo</u>	<u>big</u>	<u>blockquote</u>	<u>body</u>	<u>br</u>	<u>button</u>	<u>caption</u>
<u>center</u>	<u>cite</u>	<u>code</u>	<u>col</u>	<u>colgroup</u>	<u>dd</u>	<u>del</u>	<u>dfn</u>
<u>dir</u>	<u>div</u>	<u>dl</u>	<u>dt</u>	<u>em</u>	<u>fieldset</u>	<u>font</u>	<u>form</u>
<u>frame</u>	<u>frameset</u>	<u>h1-h6</u>	<u>head</u>	<u>hr</u>	<u>html</u>	<u>i</u>	<u>iframe</u>
<u>img</u>	<u>input</u>	<u>ins</u>	<u>isindex</u>	<u>kbd</u>	<u>label</u>	<u>legend</u>	<u>li</u>
<u>link</u>	<u>map</u>	<u>menu</u>	<u>meta</u>	<u>noframes</u>	<u>noscript</u>	<u>object</u>	<u>ol</u>
<u>optgroup</u>	<u>option</u>	<u>p</u>	<u>param</u>	<u>pre</u>	<u>q</u>	<u>s</u>	<u>samp</u>
<u>script</u>	<u>select</u>	<u>small</u>	<u>span</u>	<u>strike</u>	<u>strong</u>	<u>style</u>	<u>sub</u>
<u>sup</u>	<u>table</u>	<u>tbody</u>	<u>td</u>	<u>textarea</u>	<u>tfoot</u>	<u>th</u>	<u>thead</u>
<u>title</u>	<u>tr</u>	<u>tt</u>	<u>u</u>	<u>ul</u>	<u>var</u>		

Accès aux éléments

▶ Par attribut name

- N'est pas autorisé pour tous les éléments HTML
- Seulement :
 - a, applet, frame, img, input, iframe, map, meta, object, param, select, textarea

`getElementsByName(« toto »);`

▶ Par attribut « id »

- Autorisé dans presque tous les éléments HTML

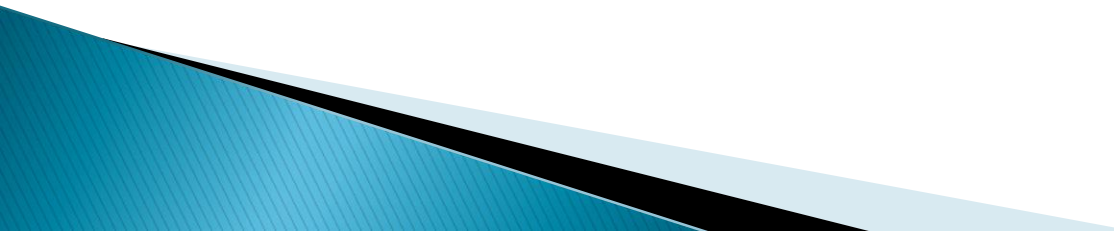
`getElementById(« monId »);`

▶ Par liste d'éléments

`getElementsByTagName()`

Propriétés universelles

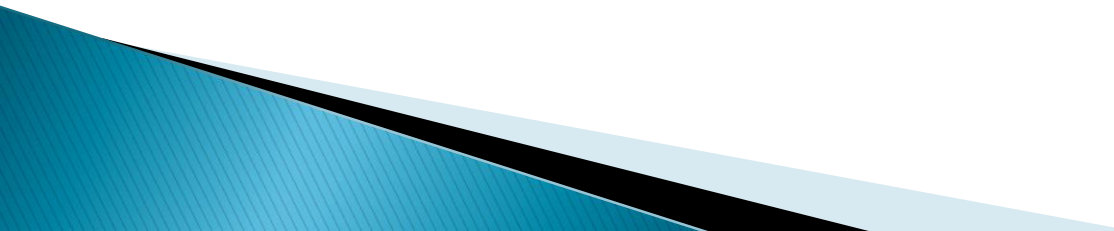
Valables pour tous les objets HTML

- ▶ className
 - ▶ dir (sens de lecture)
 - ▶ id (nom d'identification, unique)
 - ▶ lang (langue du pays : de, fr, en, it ...)
 - ▶ title (titre)
- 

Propriétés propres

Chaque élément a ensuite des propriétés et méthodes qui lui sont propres.

Exemple : élément **button**

- ▶ **accessKey** Raccourci clavier pour l'accès par clavier
 - ▶ **disabled** Le bouton ne peut pas être confirmé
 - ▶ **form** Élément de formulaire dont il fait partie
 - ▶ **name** Nom pour le bouton
 - ▶ **tabIndex** Recherche par tabulateur
 - ▶ **type** Type du bouton
 - ▶ **value** Valeur d'envoi du bouton
- 

Exemple 9

```
<script>
<!--
function bouton_heure() {
    var maintenant = new Date();
    var hours = maintenant.getHours();
    var minutes = maintenant.getMinutes();
    var seconds = maintenant.getSeconds();

    document.getElementById("mybutton").innerText =
        hours+':'+minutes+':'+seconds;
}
//-->
</script>
</head>

<body>
<fieldset>
<button id="mybutton" value=""
    onclick="bouton_heure()">Heure!</button>
</fieldset>
</body>
```