



Javascript

programme javascript

- script définissant des fonctions, résidant sur le serveur web
 - inclus directement dans une page html (dans l'en-tête)

```
<SCRIPT LANGUAGE=javascript>
function Hello() {
    alert('Bonjour');
}
</SCRIPT>
```
 - ou dans un fichier à part référencé dans la page html :

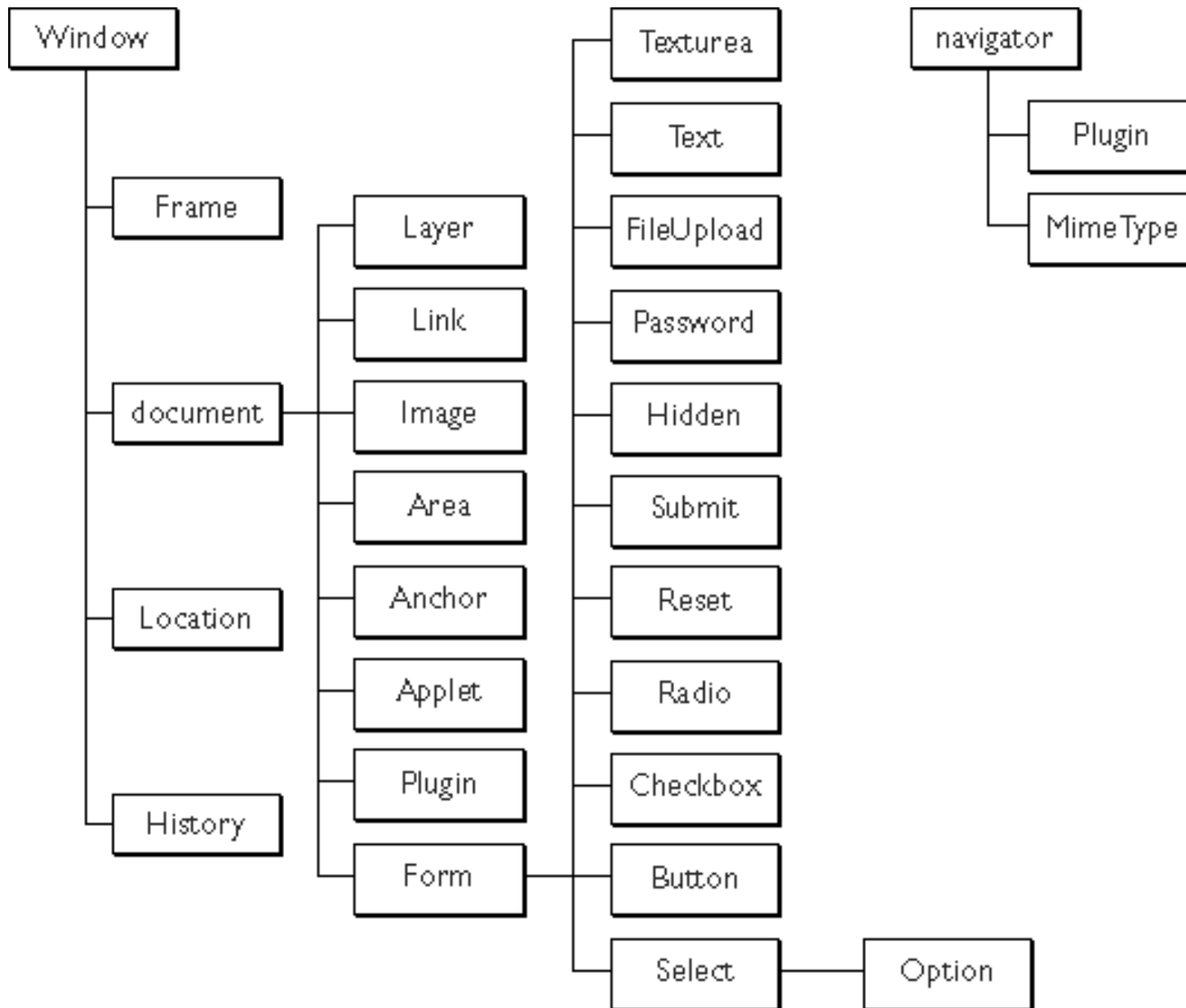
```
<SCRIPT SRC="monfichier.js" LANGUAGE="javascript"> </SCRIPT>
```
- script inclus dans le corps de la page html pour en fabriquer une partie

```
<SCRIPT LANGUAGE=javascript>
    document.write((new Date()).toLocaleString());
</SCRIPT>
```
- téléchargé depuis le serveur web, puis interprété et exécuté par le navigateur
- agissant sur les éléments des pages affichées

exécution d'une fonction

- par clic sur lien
` On y va `
- juste après le chargement complet de la page
`<BODY ... ONLOAD="Hello();">`
- par clic sur objet de formulaire
`<INPUT ... ONCLICK="Hello();">`
- par "soumission" de formulaire
`<FORM onSubmit="Hello(); return(false)">`
- par passage sur un lien
` On y va `
- d'une manière générale, par réaction à un événement sur un objet de la page html (programmation événementielle ou réactive)
- exemple sur
<http://www.info.univ-angers.fr/pub/pn/JavaScript/ouvre.html>

hiérarchie des objets du navigateur

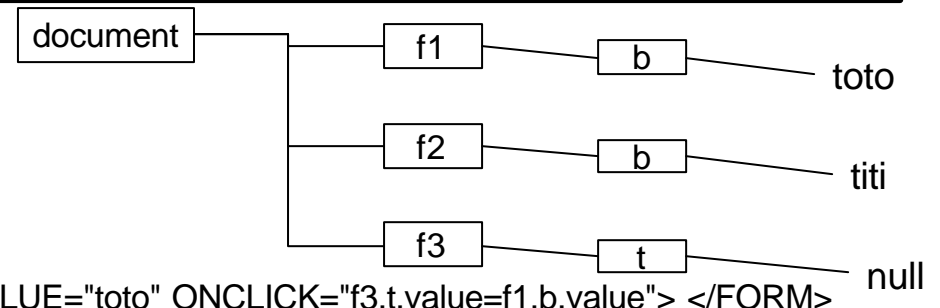
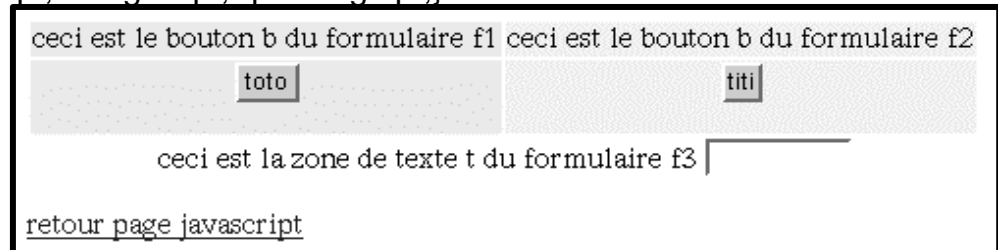


gestion des formulaires

- soit F un formulaire
 - F.length est le nombre de composants dans le formulaire
 - F.elements est le tableau des composants du formulaire, indicé de 0 à F.length – 1
- soit T une boite de type text, password ou textarea
 - T.value = valeur saisie dans la boite
- soit C une case à cocher de type checkbox
 - C.checked = booléen indiquant si la case est cochée ou non
 - C.value = valeur de la case
- soit R un bouton radio avec N positions
 - R[i].value = valeur de la position n°i, i=0,...,N-1
 - R[i].checked = booléen indiquant si la position n°i, i=0,...,N-1, est cochée
- soit S une liste de choix de type select avec N options
 - S.selectedIndex = n° de l'option choisie (entre 0 et N-1)
 - S.options = tableau des options
 - S.options[i].value = valeur de l'option n°i, i=0,...,N-1.
 - S.options[S.selectedIndex].value = valeur de l'option choisie
- les instructions javascript peuvent agir sur les différentes propriétés des composants d'un formulaire.
- Ex sur <http://www.info.univ-angers.fr/pub/pn/JavaScript/index.html>

<http://www.info.univ-angers.fr/pub/pn/JavaScript/exboutons.html>

```
<HTML>
<HEAD>
<TITLE>exemples boutons</TITLE>
<STYLE TYPE="text/css">
  TD {text-align:center; vertical-align:middle; border-width:0pt; margin:0pt; padding:5pt;}
  TD.bleu {background-color:#ffff3e;}
  TD.jaune {background-color:#aaffff;}
  TD.blanc {background-color:#ffffff;}
</STYLE>
</HEAD>
<BODY BGCOLOR="white">
<TABLE>
<TR>
  <TD class="jaune">ceci est le bouton b du formulaire f1</TD>
  <TD class="bleu">ceci est le bouton b du formulaire f2 </TD>
</TR>
<TR>
  <TD class="jaune">
    <FORM NAME="f1"><INPUT TYPE="button" NAME="b" VALUE="toto" ONCLICK="f3.t.value=f1.b.value"> </FORM>
  </TD>
  <TD class="bleu">
    <FORM NAME="f2"><INPUT TYPE="button" NAME="b" VALUE="titi" ONCLICK="f3.t.value=f2.b.value"> </FORM>
  </TD>
</TR>
<TR>
  <TD class="blanc" COLSPAN="2">
    <FORM NAME="f3">ceci est la zone de texte t du formulaire f3 <INPUT TYPE="text" NAME="t" SIZE="10"> </FORM>
  </TD>
</TR>
</TABLE>
<A HREF="index.html">retour page javascript</A>
</BODY>
</HTML>
```



passage de paramètres

- Problème : saisir un nom (ex : toto) dans le champ d'un formulaire et écrire "bonjour toto" dans un autre champ de ce formulaire
- le formulaire doit contenir
 - un champ nommé (nomsaisi par exemple) pour la saisie du nom
 - un champ nommé (aff par exemple) pour écrire le message
 - un bouton pour déclencher l'action
- la fonction d'affichage est $f(X, Y)$
 - X est le nom saisi (donnée)
 - Y est le champ où écrire le message (donnée modifiée)
- il faut passer le nom saisi à la fonction d'affichage $f(X, Y)$
 - si le paramètre formel X de la fonction f est une chaîne de caractères, alors le paramètre effectif est nomsaisi.value
 - si le paramètre formel X de la fonction est un objet, alors le paramètre effectif est l'objet nomsaisi lui-même et la fonction doit utiliser S.value pour accéder à la chaîne de caractères saisie
 - le paramètre formel Y est obligatoirement l'objet (puisqu'il est modifié) et le paramètre effectif est l'objet aff.

première solution

```
<form>
```

```
  entrez un nom :<input type="text" name="nomsaisi">
```

```
  <input type="text" name="aff">
```

```
  <input type="button" value="OK" onclick= " f(nomsaisi.value, aff)">
```

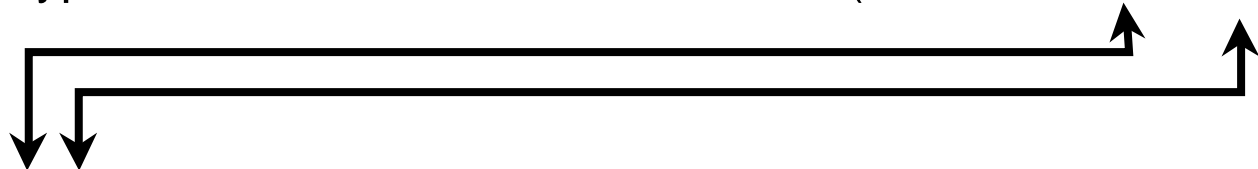
```
</form>
```

```
function f(X,Y){
```

```
// X est une chaîne de caractères
```

```
  Y.value=X;
```

```
}
```



deuxième solution

```
<form>
  entrez un nom :<input type="text" name="nomsaisi">
  <input type="text" name="aff">
  <input type="button" value="OK" onclick= " f(nomsaisi, aff)">
</form>
```

```
function f(X,Y){
// X est un objet
  Y.value=X.value;
}
```

éléments de syntaxe

- ; est un séparateur d'instructions
- { ... } regroupe des instructions dans un bloc unique
- // commentaire sur une seule ligne
- /* commentaire
sur plusieurs
lignes */
- déclaration de variable : `var X = expression;`
 - la variable est du type de l'expression, mais ce type peut évoluer au fur et à mesure des affectations successives (langage faiblement typé)
 - la casse (majuscule/minuscule) est prise en compte par le langage (X et x sont deux variables différentes)
- affectation : `X = expression;`
- définition de fonction
 - fonction nom(liste de paramètres) {
corps de la fonction
return résultat; // éventuellement
}
 - la liste de paramètres formels
 - peut être vide ()
 - ou être de la forme p1, ..., pn

types primitifs

- logiques (Boolean)
 - deux valeurs possibles true et false
 - nombres (Number)
 - on ne distingue pas les entiers et les décimaux
 - `parseInt("123")` retourne le nombre 123
 - `parseFloat("123.45")` retourne le nombre 123.45
- indispensables pour convertir
en nombres des données
saisies dans un champ texte de
formulaire

opérateurs

- arithmétiques
 - +, -, *, / (éventuellement division entière, 14 / 3 retourne 4), % (modulo 14 % 3 retourne 1), ++ (incrémente un entier)
- de comparaison
 - == (égalité), != (différence), <, >, <=, >=
- logiques
 - && (et), || (ou), ! (non)

classes

- chaînes de caractères (String)
 - `var X="bonjour";` ou `'bonjour'`
 - `var Y="toto dit : \"bonjour\"";` pour insérer un " dans une chaîne il faut le faire précéder de \
 - `X.length` est la propriété qui est égale au nombre de caractères dans X
 - `+` opérateur de concaténation
`X+" titi"` est la chaîne "bonjour titi"
 - de très nombreuses méthodes sont disponibles pour les objets String ex :
`X.charAt(4)` retourne "j" le caractère à la 4^{ème} place dans X
- Documentation complète sur le serveur interne de documentation
 - http://doc.info-ua/langages/guide_javascript/
 - http://doc.info-ua/langages/reference_javascript/
- NB : *ce cours n'a pas vocation à être un cours de programmation objet, les concepts de classe, méthodes, propriétés ne seront pas détaillés formellement*

type tableau

- déclaration de tableau
 - `var T = new Array(10);` crée un tableau de 10 cases, indicées de 0 à 9
 - `var P = new Array("Pierre", "Paul", "Jacques");` crée un tableau de 3 cases contenant les valeurs spécifiées
 - `T[k]` désigne la valeur d'indice k dans T
 - un tableau à 2 dimensions est un tableau de tableaux, chaque "ligne" doit être créée avec sa longueur particulière
 - `T[i][j]` désigne la valeur d'indice j dans la ligne d'indice i
 - `var Cal = new Array(13); // pour une année (Cal[0] inutilisé)`
`Cal[1] = new Array(32); // pour janvier (Cal[1][0] inutilisé)`
`Cal[2] = new Array(30); // pour février (Cal[2][0] inutilisé)`
...
`Cal[5][17] = info pour le 17 mai`

structures de contrôle

- conditionnelle
if (condition) {
 instructions
}
- alternative
if (condition) {
 instructions1
}
else {
 instructions2
}
- cas
switch (expression) {
 case val1 : instructions1; break;
 ...
 case valn : instructionsn; break;
 default : instructions par défaut;
}
équivalent à
if (expression==val1) { instructions1 }
else if (expression==val2) { instructions2 }
 else ...
 ...
 else { instructions par défaut }

- boucle *pour i=valdéb croissant jusqu'à valfin*

```
for (i=valdéb; i<=valfin; i++) {
    instructions
}
```
- boucle pour "générale"

```
for (initialisation; condition; mise à jour) {
    instructions
}
```

la boucle s'arrête quand la condition n'est plus respectée
- boucle tant que

```
while (condition) {
    instructions
}
```

la boucle s'arrête quand la condition n'est plus vérifiée
- boucle répéter jusqu'à

```
do {
    instructions
}
while (condition)
```

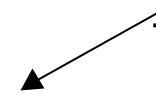
la boucle s'arrête quand la condition n'est plus vérifiée (au moins une itération)

ouverture de nouvelle fenêtre

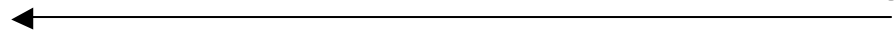
- affichage d'une page html existante
`window.open('f.html','toto', 'width=100,height=200')`
 - nom du fichier contenant la page
 - nom de la fenêtre
 - largeur et hauteur en pixelson peut fixer de nombreux autres paramètres
- affichage d'une page html construite par le script
 - algorithme
 - créer la fenêtre
 - ouvrir son document
 - écrire dans le document le code html
 - fermer le document
 - positionner la fenêtre au 1^{er} plan


```
function cree_et_ouvrefenetre(){
var Fen=window.open("",'toto','width=100,height=200');
Fen.document.open();
Fen.document.writeln('<html><head><title>');
Fen.document.writeln('toto</title></head>');
Fen.document.writeln('<body bgcolor="white" text="red">');
Fen.document.writeln('fenêtre construite par un script ');
Fen.document.writeln('</body> </html> ');
Fen.document.close();
Fen.focus();
}
```

création de la
fenêtre Fen



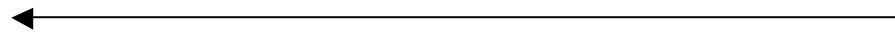
ouverture du
document



écriture
de la
page



fermeture du
document



positionnement
de la fenêtre



modification du contenu d'une fenêtre

```
function modifie_fenetre(){  
document.clear();  
document.open();  
document.writeln('<html><head><title>');  
document.writeln('toto</title></head>');  
document.writeln('<body bgcolor=white text=red>');  
document.writeln('fenêtre construite par un script ');  
document.writeln('</body> </html> ');  
document.close();  
}
```

effacement
du document

ouverture du
document

écriture
de la
page

fermeture du
document

exemple : jeu du nombre inconnu

- Il s'agit de fournir à l'utilisateur une page web dans laquelle il essaye de découvrir un nombre entier inconnu, choisi au hasard par l'ordinateur et compris entre 0 et 1000. Pour chaque nombre proposé par le joueur, l'ordinateur indique si ce nombre est plus petit, plus grand ou égal au nombre inconnu.
- Le joueur doit pouvoir abandonner et commencer un autre jeu quand il veut et il voit le nombre de coups joués lorsqu'il trouve la solution.
- jeu disponible à partir de <http://www.info.univ-angers.fr/pub/pn/JavaScript/nombreinconnu.html>

entrez votre nombre ci-dessous et
appuyez sur "Entrée"

ou cliquez sur
OK

OK

BRAVO !!! le nombre caché était bien 972
vous l'avez découvert en 8 coups
Cliquez sur le bouton Nouveau Jeu pour démarrer
970 est trop petit
960 est trop petit
975 est trop grand
950 est trop petit
825 est trop petit
750 est trop petit
500 est trop petit

Solution Nouveau Jeu

source de la page web : nombreinconnu.html

```
<HTML>
<HEAD>
<TITLE>jeu du nombre inconnu</TITLE>
<SCRIPT SRC="nombreinconnu.js" LANGUAGE="javascript">
  </SCRIPT>
</HEAD>
<BODY BGCOLOR="white">
<CENTER>
<FORM NAME="jeu">
<!-- champ caché pour mémoriser le nombre inconnu -->
<INPUT TYPE="hidden" NAME="nbcache">
<!-- champ caché pour mémoriser le nombre de coups joués -->
<INPUT TYPE="hidden" NAME="nbcoups" VALUE="0">
```

```
<TABLE WIDTH="50%" CELLSPACING="5" CELLPADDING="5">
<TR>
<TD ALIGN="center">entrez votre nombre ci-dessous</TD>
<TD ALIGN="center">puis cliquez sur OK</TD>
</TR>
<TR>
<!-- champ de saisie du nombre essayé par le joueur -->
<TD ALIGN="center">
<INPUT TYPE="text" NAME="nbessai" SIZE="4" MAXLENGTH=4> </TD>
<!-- bouton pour déclencher la vérification du nombre -->
<TD ALIGN="center">
<INPUT TYPE="button" ONCLICK="verif(jeu)" VALUE="OK"> </TD>
</TR>
<TR>
<TD ALIGN="center" COLSPAN=2>
<!-- zone de texte pour afficher les commentaires -->
```

```
<TEXTAREA COLS=50 ROWS=10 NAME="resultat" READONLY>
```

Cliquez sur le bouton Nouveau Jeu pour démarrer

```
</TEXTAREA></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD ALIGN="center">
```

```
<!-- bouton pour abandonner le jeu -->
```

```
<INPUT TYPE="button" ONCLICK="solution(jeu)"> </TD>
```

```
<!-- bouton pour recommencer une partie -->
```

```
<TD ALIGN=center>
```

```
<INPUT TYPE="button" ONCLICK="initialisation(jeu)"></TD>
```

```
</TR>
```

```
</TABLE>
```

```
</FORM>
```

```
</CENTER>
```

```
<A HREF="index.html">retour vers page JavaScript</A>
```

```
</BODY>
```

```
</HTML>
```

source des fonctions de nombreinconnu.js

```
function initialisation(J) {  
    // initialise le formulaire de jeu  
    var NC = Math.floor(1001 * Math.random());  
    J.nbcache.value=NC;  
    J.resultat.value="";  
    J.nbessai.value="";  
    // ajoutetexte(J.resultat,NC); pour voir le nombre tiré au sort  
}
```

```
function ajoutetexte(TeA, T) {  
    // ajoute le texte T au début du TextArea TeA  
    TeA.value=T+ TeA.value;  
}
```

```
function solution(J) {  
    // affiche la solution du jeu  
    ajoutetexte(J.resultat, "\nle nombre caché était : "+J.nbcache.value);  
}
```



```
function verif(J) {  
  // contrôle le nombre essayé  
  if (isNaN(J.nbessai.value)) {  
    ajoutetexte(J.resultat, "\n saisissez un nombre");  
  }  
  else {  
    var NEssai = parseInt(J.nbessai.value);  
  
    if ((NEssai > 1000) || (NEssai < 0)) {  
      ajoutetexte(J.resultat, "\n Allons, le nombre est entre 0 et 1000");  
    }  
  }  
}
```

```
else {
    NCache = parseInt(J.nbcache.value);
    J.nbcoups.value = 1+parseInt(J.nbcoups.value);
    if (NEssai > NCache) {
        ajoutetexte(J.resultat, "\n"+NEssai+" est trop grand");
    }
    else {
        if (NEssai < NCache) {
            ajoutetexte(J.resultat, "\n"+NEssai+" est trop petit");
        }
    }
}
```

```
else {
    ajoutetexte(J.resultat, "\nBRAVO !!! le nombre caché était
        bien "+NEssai);
    ajoutetexte(J.resultat, "\nvous l'avez découvert en
        "+parseInt(J.nbcoups.value)+" coups");
    ajoutetexte(J.resultat, "\nCliquez sur le bouton Nouveau
        Jeu pour démarrer");
}
}
}
}
```