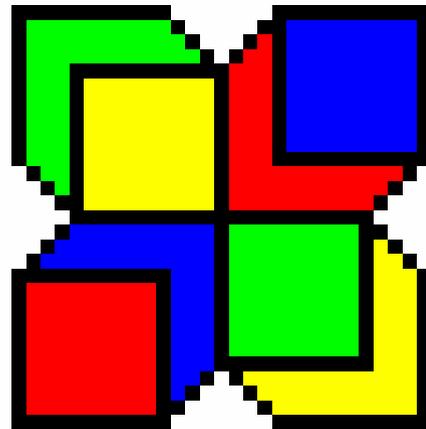


## Présentation de



# JavaScript



# *Sommaire*

## ■ **Présentation de JavaScript**

- ✓ Introduction
- ✓ Java et JavaScript
- ✓ Quelques aspects techniques
- ✓ Conclusion

- **Contrôle du contenu et de l'apparence de pages Web**
- **Contrôle de fonctionnalités du Navigateur Internet**
- **Gestion du contenu d'une page Web**
- **Gestion d'interactions avec l'utilisateur**

- **Liens avec les applets Java**
- **Manipulation et gestion d'images**
- **Lecture et écriture de Cookies**
- **Autres fonctionnalités intéressantes**



# *Sommaire de la section*

## ■ Introduction

- ✓ JavaScript : quelques repères fondamentaux
- ✓ Client-side et Serveur-side
- ✓ Qu'est ce que JavaScript sait faire
- ✓ Qu'est ce JavaScript ne sait pas faire

## ■ Java et JavaScript

- ✓ Complémentarité et Intégration
- ✓ Différences entre Java et JavaScript

## ■ Exemples et aspects techniques

- ✓ Un exemple élémentaire
- ✓ Intégration avec les formulaires HTML
- ✓ Gestion des aspects de sécurité
- ✓ Comparatif avec d'autres environnements

## ■ Bibliographie & @ internet



# *Quelques repères fondamentaux (1)*

## ■ Contexte d'utilisation

- ✓ JavaScript(JS) s'utilise dans le contexte du Web principalement au travers d'un navigateur Internet
- ✓ JS permet l'exécution de code informatique intégré à des pages Web
- ✓ JS est un des outils permettant de développer du « Web Dynamique »

## ■ Contexte technique

- ✓ JS n'est pas simple, et nécessite une relative bonne connaissance des techniques de programmation
- ✓ JS n'est pas un sous-ensemble de Java
- ✓ JS n'est pas une alternative à Java mais une technologie complémentaire
- ✓ JS est un Langage de troisième génération cousin de Java, du langage C ou du Basic en terme de forme de programmation



# *Quelques repères fondamentaux (2)*

## ■ Principales caractéristiques

- ✓ *JS est Interprété, faiblement typé et sans notion de pointeurs*
- ✓ *Types de base simples (nombre, chaîne et booléen)*
- ✓ *Les fonctions et les variables peuvent être déclarées n'importe où*
- ✓ *Les fonctions admettent un nombre de paramètre variable*
- ✓ *JS comme Java est indépendant du matériel et donc portable*

## ■ JavaScript et l'Objet

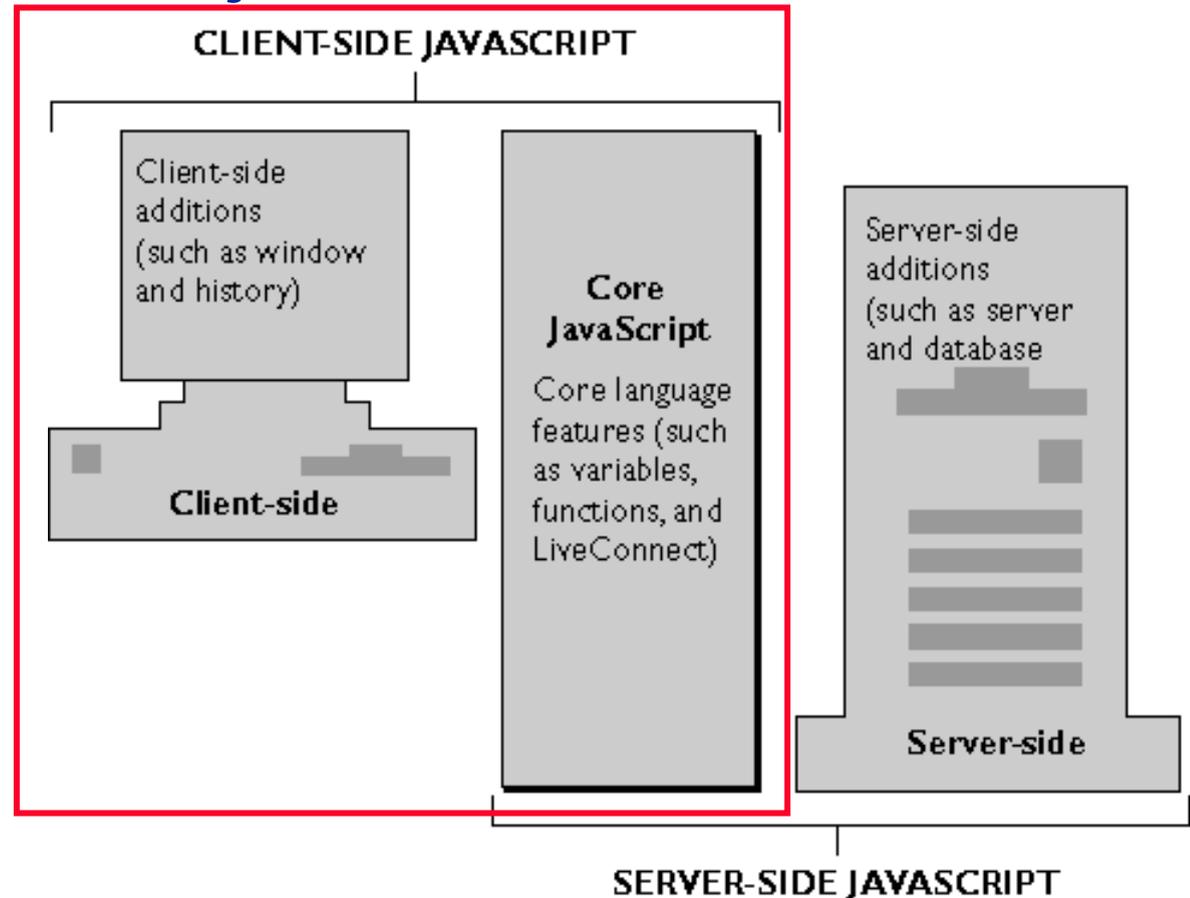
- ✓ *Langage basé sur les objets mais non orienté objet*
- ✓ *Les objets sont simples et informels, il possèdent des propriétés et des méthodes associées*
- ✓ *Existence d'un système de ramasse-miettes*

***JS est en général intégré dans d'autres logiciels (navigateurs) qui fournissent des données sous forme d'objets utilisés par les scripts JavaScript***



# JavaScript Client-side et Server-side

## Objet de ce cours





# *Ce que JavaScript sait faire (1)*

## **Le réel intérêt de JS tient de son étroite liaison avec les navigateurs Internet**

- ✓ Contrôle du contenu et de l'apparence de page Web
  - write()... faire apparaître un texte supplémentaire dynamiquement
  - construire une page dynamiquement à partir de rien
- ✓ Contrôle de fonctionnalités du Navigateur Internet
  - affichage de message par boite de dialogue ou dans la barre de status
  - saisie de données simples
  - ouverture et gestion de l'affichages d 'URL dans des fenêtres du navigateur
- ✓ Gestion du contenu d'une page Web
  - accès à la liste des liens hypertextes, images, applets Java,... d'une page
  - accès à l'ensemble des contrôles boutons, zone de saisie,... d'une page
- ✓ Gestion d'interactions avec l'utilisateur
  - gestions d 'actions liées à des événements utilisateur : déplacement de la souris sur un contrôle, validation d'une saisie, d'un *Submit* sur un formulaire...



## *Ce que JavaScript sait faire (2)*

- ✓ Liens avec les applets Java
  - fonctionnalité Netscape « LiveConnect »
  - invocation de méthode d'applet Java depuis JS
  - lecture et écriture de propriétés d 'applet Java depuis JS
- ✓ Manipulation et gestion d'images
  - possibilité de réaliser des effets sur images : passage de la souris sur une image ou lors d 'un clique sur image...
- ✓ Lecture et écriture de Cookies
  - Cookies : initialement prévu pour les scripts CGI, code écrit et lu par le serveur CGI et mémorisé sur le client pour générer des pages en fonction du contexte client.
  - JS peut lire et écrire des Cookies et peut parfois remplacer l 'utilisation de script CGI
- ✓ etc..
  - Utilisation de délai `window.setTimeout()`
  - Utilisation de fonctions sophistiquées de gestion de la Date et Heure



# *Ce que JavaScript NE sait PAS faire*

**JS est prévu pour être utilisé au travers de navigateur Internet et ne comporte pas de propriétés propres aux développements d'applications indépendantes « standalone »**

- ✓ JS ne comporte pas de fonctionnalités graphiques exceptées celles permettant le formatage d'éléments de pages HTML (images, fontes,...)
- ✓ JS ne permet pas la lecture et l'écriture de fichiers quelconques sur le disque
- ✓ JS ne supporte pas de services réseaux, il peut seulement imposer au navigateur de télécharger une URL
- ✓ JS ne gère pas de multitâches
- ✓ ...

Note : JS s'appuie sur les procédures d'entrées/sorties d'autres logiciels lors de l'exécution des scripts



# Complémentarité et Intégration

## ■ Complémentarité

- ✓ JS est facile à utiliser et à maintenir car intégré à une page HTML
  - ✓ Un code Java s'écrit dans un fichier spécifique réutilisable...
- 
- ✓ JS peut contrôler le navigateur et le contenu des pages affichées mais ne peut pas réaliser d'opérations graphiques ou nécessitant de la communication réseau
  - ✓ Java n'offre aucune interface de contrôle de navigateurs, mais permet de réaliser des fonctions graphiques, des appels réseaux, du multitâche

## ■ Intégration

- ✓ *LiveConnect* de Netscape permet à JavaScript de communiquer avec l'interpréteur java intégré au navigateur et de contrôler et accéder aux attributs manipulés par les applets Java de la page.



# *Différences entre JavaScript et Java*

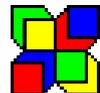
## **JavaScript**

- Interprété par le client - non compilé
- Basé sur les objet, pas de classes ni d'héritage,...
- Encapsulé dans HTML
- Déclaration optionnelle des variables (typage faible)
- Références vérifiées à l'exécution
- Pas de fonctionnalité graphiques
- Intégration dans l'architecture sous-jacente de navigateurs
- Issu de Netscape Corporation

## **Java**

- compilé avant exécution sur le client
- Orienté objet
- Applets distinctes de HTML
- Déclaration obligatoire des variables (typage fort)
- Références vérifiées à la compilation
- Fonctionnalités graphiques complètes de Java
- Environnement indépendant de navigateurs (sauf LiveConnect)
- Issu de Sun Microsystem

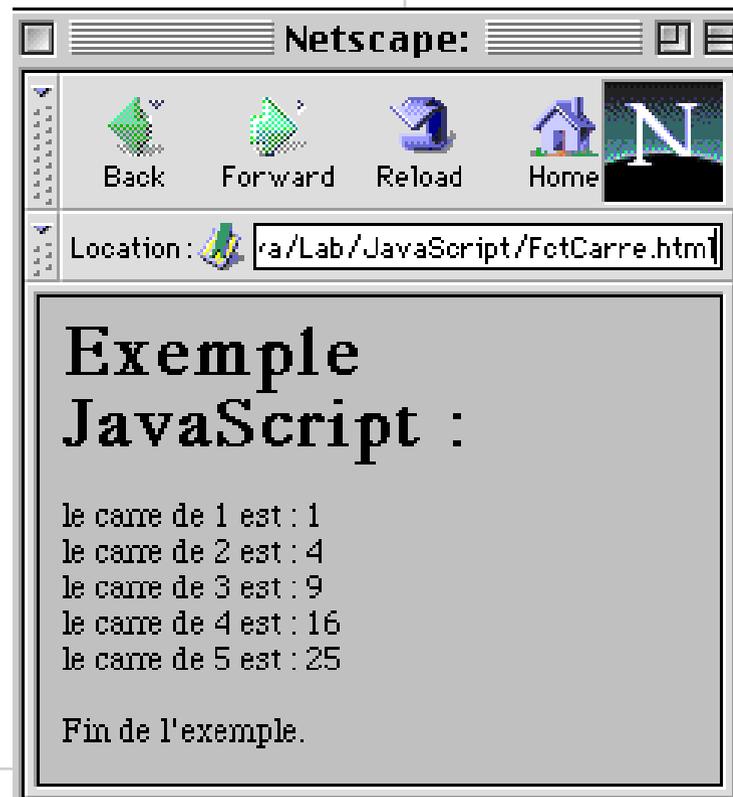
***JavaScript utilise un nombre restreint de types de données du langage Java***

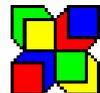


# Un exemple élémentaire

```
<SCRIPT LANGUAGE="JavaScript">
  <!-- commentaire pour masquer le script pour les anciens
  navigateur
  function carre (i) {
    document.write ("le carre de " + i + " est : ");
    return i * i;
  }
  // fin du masquage -->
</SCRIPT>

<BODY>
  <H1>Exemple JavaScript :</H1>
  <SCRIPT>
    document.write(carre(1), "<BR>")
    document.write(carre(2), "<BR>")
    document.write(carre(3), "<BR>")
    document.write(carre(4), "<BR>")
    document.write(carre(5), "<BR>")
  </SCRIPT>
  <BR>
  Fin de l'exemple.
</BODY>
```





# Intégration avec les formulaires HTML

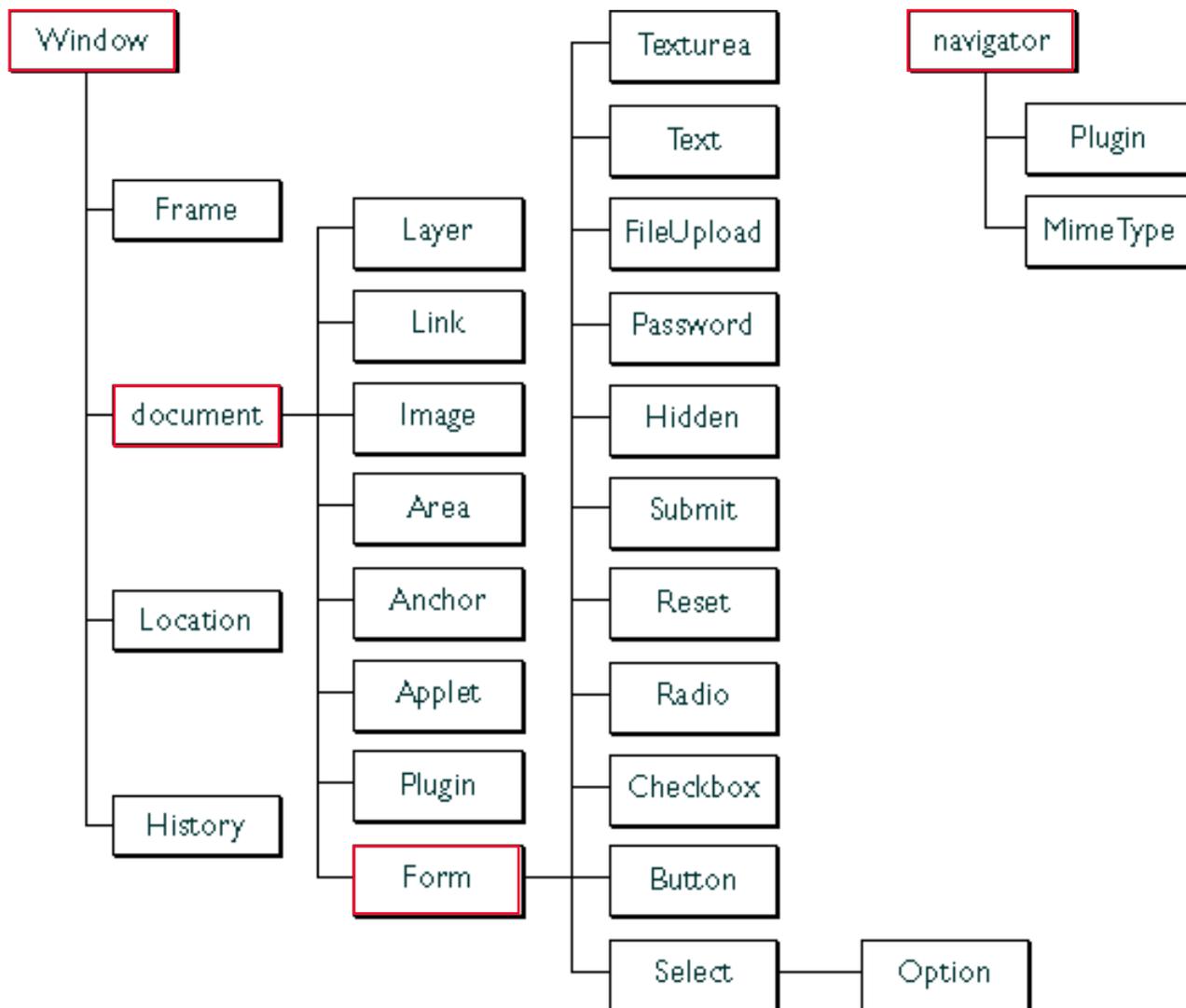
- ✓ JS s'intègre aisément dans une page HTML

```
<HEAD>
<TITLE> Exemple d'utilisation de JavaScript</TITLE>
<SCRIPT LANGUAGE="JavaScript">
  var v1 = 50;
  var tab1 = new Object();
  tab1[0] = 4000; tab2 = 6000; ...
  function f1 () { var f = document.maForm;
                  var num = parseInt(f.num.value);...}
  function fn() { ... }
</SCRIPT>
</HEAD>
<BODY>
  <FORM NAME = "maForm">
    ...
    <INPUT TYPE=text NAME="num" SIZE=2 onChange="f1 () ">
    ...
  </FORM>
</BODY>
```

- ✓ JS supporte le chargement d'un script

```
<SCRIPT SRC="../../../javascript/courdier.js"></SCRIPT>
```

# Windows/Documents/Forms/Browser





# *La sécurité*

## JavaScript est un code téléchargé et exécuté sur une machine cliente : problème de sécurité

- ✓ Limite du langage : pas de possibilité de lire, écrire, supprimer, lister des fichiers ou répertoires sur le poste d'exécution client  
pas d'altération de données ni de greffe de virus possible
- ✓ Pas d'accès direct à d'autres machines au travers du réseau  
impossible de cracker des password intranet sans être vu par le firewall
- ✓ Pas d'accès aux propriétés privées du navigateur  
impossible d'accéder à l'historique de navigation, aux sites favoris, etc.

■ **Il existe des brèches... à chaque version des navigateurs on en trouve !**



# *Bibliographie & @ internet*

## ■ Quelques références

- ✓ David Flanagan, "JavaScript The Definitive Guide", ed. O'REILLY, 1997
- ✓ Philippe CHALEAT et Daniel CHARNAY, "Programmation HTML et JavaScript", ed. Eyrolles, 1999
- ✓ Nigel McFarlane, "Le guide du programmeur JavaScript", ed. Erolles, 1999
- ✓ Paul Wilton, "Maîtriser JavaScript", ed. CampusPress, 2001

## ■ Sites Netscape

- ✓ [http://home.netscape.com/one\\_stop/intranet\\_apps/index.html](http://home.netscape.com/one_stop/intranet_apps/index.html)  
Source for starter applications, technical information, tools, and expert forums for quickly building and dynamically deploying open intranet applications.
- ✓ <http://help.netscape.com/kb/tools/index.html>  
Technical support page for information on the LiveWire Database Service.
- ✓ <http://developer.netscape.com/tech/javascript/ssjs/ssjs.html>  
Support page for information on server-side JavaScript.

## ■ Autres @ Internet

- ✓ <http://javaScript.netscape.com>
- ✓ <http://www.javasoft.com> ( *...et tous les liens associés...* )
- ✓ <http://scriptsearch.internet.com/JavaScript/>