


Introduction

## Cours Web n°9

Introduction à JavaScript

Sandrine-Dominique Gouraud (gouraud@lri.fr)  
Pierre Senellart (pierre@senellart.com)



Semaine du 28 novembre 2005

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 1 / 25

Introduction

## Plan du cours

- 1 Introduction
- 2 L'objet Node
- 3 Fonctions utiles de JavaScript
- 4 Gestionnaire d'événements
- 5 Références
- 6 Application

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 1 / 25

Introduction À quoi ça sert ?

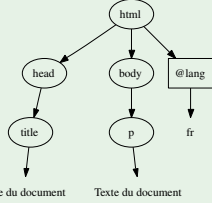
- PHP, CGI... : permettent des comportements dynamiques **côté serveur**. Nécessitent un échange entre le navigateur et le serveur Web (soumission d'un formulaire, clic sur un lien) pour chaque comportement dynamique souhaité.
- JavaScript : permet des comportements dynamiques **côté client** : manipulation des fenêtres, changement dynamique du code HTML/CSS, interaction fine avec les formulaires...
- Permet la manipulation du DOM (**D**ocument **O**bject **M**odel), la représentation du document HTML comme un arbre, les balises étant les nœuds de l'arbre.
- "Dynamic HTML" (**DHTML**) : JavaScript + DOM + CSS
- **Alternatives** : VBScript (Internet Explorer uniquement), Java (plus complet, mais plus lourd, plug-in nécessaire), Flash (nécessite un logiciel propriétaire pour le développement, plug-in nécessaire).
- Rien à voir avec Java !

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 2 / 25

Introduction Document Object Model

### Exemple

```
<html lang="fr">
  <head>
    <title>Titre du document</title>
  </head>
  <body>
    <p>Contenu du document</p>
  </body>
</html>
```



S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 4 / 25

Introduction Le langage JavaScript

- Langage de Programmation
- Normalisé sous le nom d'**Ecmascript**
- Syntaxe très proche de PHP. Différences

	PHP	JavaScript
Variables	\$toto	toto
Concaténation	chaine1.chaine2	chaine1+chaine2
Interpolation	oui ("\$toto")	non
Tableaux	\$var=array(1,'two')	var=new Array(1,'two')
Taille tableau	count(\$var)	var.length

- Liaison d'une script JavaScript avec un document XHTML
  - ▶ toto.js contenant des fonctions JavaScript (**function**)
  - ▶ Dans le head du XHTML :
 

```
<script src="toto.js" type="text/javascript"></script>
```
  - ▶ Gestionnaires d'événement sur les balises XHTML (cf plus loin).

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 5 / 25

Introduction Modèle objet

- JavaScript basé sur le **modèle objet**
- Variables : objets complexes, ayant des propriétés (**membres**) et des fonctionnalités (**fonctions membres, méthodes**)
- En JavaScript, on accède au membre blah de l'objet toto avec **toto.blah**, et on utilise la méthode bouh de l'objet toto avec **toto.bouh(arguments)**

### Exemple

Par exemple, un objet **voiture** pourrait avoir une propriété **couleur** et des fonctionnalités **tourneGauche()**, **tourneDroite()** ou **avance(distance)**. On pourrait alors marquer :

```
voiture.couleur="bleu";
voiture.avance(100);
voiture.tourneGauche();
```

- En pratique, les objets JavaScript qu'on utilisera représenteront le document XHTML, les nœuds du documents, la fenêtre...

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 6 / 25

L'objet Node

## Plan du cours

- 1 Introduction
- 2 L'objet Node
- 3 Fonctions utiles de JavaScript
- 4 Gestionnaire d'événements
- 5 Références
- 6 Application

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 7 / 25

L'objet Node Généralités

- L'objet **node** est l'objet central du modèle DOM (Document Object Models).
- Chaque élément, chaque attribut et chaque donnée en caractères représentent des nœuds distincts. Ces nœuds forment l'arborescence.
- L'objet node dispose de propriétés et de méthodes pour accéder aux différents nœuds, peu importe s'ils sont placés très bas dans l'arborescence.

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 8 / 25

L'objet Node Généralités

`node = document.getElementById("id")` accède à un élément XHTML qui possède un attribut `id`

`valeur = node.nodeValue` sauvegarde la valeur ou le contenu d'un nœud :

- pour les nœuds texte, c'est le texte,
- pour les nœuds attribut la valeur affectée à l'attribut
- pour les nœuds élément, cette propriété a pour valeur `null`

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 9 / 25

L'objet Node Manipulation CSS

`node.className="nouvelle_classe"` pour changer le nom de la classe CSS à laquelle appartient le nœud.

`node.style.borderColor="valeur"` pour changer le style de bordure d'un nœud.

`node.style.visibility="valeur"` pour changer la visibilité d'un nœud

`node.style.display="valeur"` pour changer la propriété CSS `display` d'un nœud.

**Règle générale**

On peut changer de cette façon n'importe quelle propriété CSS, d'un nœud. Le nom de la propriété en JavaScript est identique au nom CSS, sauf que les traits d'union sont remplacés par une majuscule sur la lettre suivante. Les valeurs des propriétés en JavaScripts sont identiques aux valeurs CSS (mais doivent être mis entre guillemets).

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 10 / 25

L'objet Node Manipulation CSS

**Exemple**

JavaScript :

```
function Test() {
    document.getElementById("paragraphe").style.color = "";
}
```

XHTML :

```
<p id="paragraphe" style="color:red">un texte</p>
<a href="#" onclick="Test()">Test</a>
```

**Explication** L'exemple contient un paragraphe avec le nom `id` `paragraphe` et un lien qui si on le clique appelle la fonction `Test()`. Cette fonction retire du paragraphe la mention de feuilles de style `color`, de telle sorte que le paragraphe perde sa couleur rouge et reçoive la couleur standard pour le texte.

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 11 / 25

L'objet Node Parcours d'arbre

`node.parentNode` sauvegarde le nœud parent d'un nœud.

`node.childNodes` sauvegarde un tableau de tous les nœuds enfant disponibles d'un nœud. Quand un nœud n'a pas de nœud enfant `childNodes` a la valeur `null`

`node.nodeName` sauvegarde le nom d'un nœud.

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 12 / 25

L'objet Node Manipulation d'arbres

`node.appendChild(enfant)` ajoute un nœud nouvellement créé auparavant à la structure de nœuds existante, et cela de façon à ce qu'il soit inséré comme dernier nœud enfant d'un nœud à mentionner.

`node.removeChild(enfant)` efface un nœud enfant d'un élément.

`node.cloneNode()` établit une copie identique d'un nœud, avec ou sans la structure de sous-nœuds qui en fait partie.

`node.setAttribute("name","value")` fixe à nouveau une valeur d'attribut dans un élément. Si l'attribut existe déjà, son ancienne valeur sera remplacée par la nouvelle. Si ce n'est pas le cas, il est créé et la nouvelle valeur lui est affectée.

`node.getAttribute("name")` recherche la valeur d'un attribut déterminé dans un élément.

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 13 / 25

Fonctions utiles de JavaScript

Plan du cours

- 1 Introduction
- 2 L'objet Node
- 3 **Fonctions utiles de JavaScript**
- 4 Gestionnaire d'événements
- 5 Références
- 6 Application

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 14 / 25

Fonctions utiles de JavaScript

`alert("m")` crée une fenêtre de dialogue dans laquelle le message `m` est affiché

`back()` permet de retourner à la dernière page visitée

`close("nom_de_la_fenetre")` détruit une fenêtre du client

`confirm("m")` crée une fenêtre de dialogue pour confirmer une action : elle permet le choix entre `OK` et `Annuler`

`open("URL","nom_de_la_fenetre","options_de_la_fenetre")` crée une nouvelle fenêtre client

`prompt("m","par défaut")` crée une fenêtre de dialogue permettant la saisie et dans laquelle le message `m` est affiché

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 15 / 25

Fonctions utiles de JavaScript

**Exemple**

JavaScript :

```
function OpenWindow() {
    Info = open("fichier.htm", "secondefenetre");
}
```

XHTML :

```
<body onload="OpenWindow()">
  <p><a href="#" onclick="Info.close()">Fermer la fenêtre</a></p>
</body>
```

**Explication** : L'exemple ouvre à la lecture du fichier une deuxième fenêtre du nom de `Info`. Dans le fichier est défini un bouton. Si l'utilisateur clique sur le lien, la deuxième fenêtre est fermée.

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 16 / 25

Gestionnaire d'événements

## Plan du cours

- 1 Introduction
- 2 L'objet Node
- 3 Fonctions utiles de JavaScript
- 4 Gestionnaire d'événements**
- 5 Références
- 6 Application

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 17 / 25

Gestionnaire d'événements

- Les événements sont des actions de l'utilisateur, qui vont pouvoir donner lieu à une **interactivité**.
- L'événement est le clic de souris, car c'est le seul que le XHTML gère.
- Il est possible d'associer des fonctions à des événements.
- La syntaxe est la suivante :

```
onevenement="Action_Javascript_ou_Fonction()"
```

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 18 / 25

Gestionnaire d'événements Événements courants

**onblur** se produit lorsque l'élément (*input*, *textarea*, *select*) perd le focus, c'est-à-dire que l'utilisateur clique hors de cet élément, celui-ci n'est alors plus sélectionné comme étant l'élément actif.

**onchange** se produit lorsque l'utilisateur modifie le contenu d'un champ de données (*input*, *textarea*, *select*).

**onclick** se produit lorsque l'utilisateur clique sur l'élément (*a*, *input* (*radio*, *checkbox*)) associé à l'événement. Si le gestionnaire d'événement renvoie *false*, le clic n'est pas effectué.

**onfocus** se produit lorsque l'utilisateur donne le focus à un élément (*input*, *textarea*, *select*), c'est-à-dire que cet élément est sélectionné comme étant l'élément actif.

**onload** se produit lorsque le navigateur de l'utilisateur charge la page en cours (*body*).

**onreset** se produit lorsque l'utilisateur efface les données d'un formulaire (*form*) à l'aide du bouton Reset.

**onsubmit** se produit lorsque l'utilisateur clique sur le bouton de soumission d'un formulaire (*form*). Si le gestionnaire

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 19 / 25

Gestionnaire d'événements Événements courants

**Exemple**

```
<form name="Test" action="">
  <input size="30"
    onfocus="this.value='Entrez votre nom ici'" />
  <input size="30"
    onfocus="this.value='Entrez votre adresse ici'" />
  <input size="30"
    onfocus="this.value='Entrez votre âge ici'" />
</form>
```

**Explication :** Dans l'exemple un formulaire est défini et contient trois champs de saisie. Étant donné que les champs ne portent pas d'inscription, l'utilisateur ne sait pas ce qu'il doit entrer dans les différents champs. Pourtant, s'il déplace le curseur par curiosité dans l'un des champs de saisie, le gestionnaire d'événement *onFocus* du champ concerné deviendra actif. Alors s'inscrira dans le champ concerné une invite de ce qu'il faut y mettre.

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 20 / 25

Gestionnaire d'événements Autres événements

**ondblclick** se produit lorsque l'utilisateur double-clique sur l'élément associé à l'événement (un lien hypertexte ou un élément de formulaire).

**onmouseover** se produit lorsque l'utilisateur positionne le curseur de la souris au-dessus d'un élément.

**onresize** se produit lorsque l'utilisateur redimensionne la fenêtre du navigateur.

**onselect** se produit lorsque l'utilisateur sélectionne un texte (ou une partie d'un texte) dans un champ de type "text" ou "textarea"

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 21 / 25

Gestionnaire d'événements Références

## Plan du cours

- 1 Introduction
- 2 L'objet Node
- 3 Fonctions utiles de JavaScript
- 4 Gestionnaire d'événements
- 5 Références**
- 6 Application

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 22 / 25

Gestionnaire d'événements Références

- Standards
  - ▶ Langage ECMAScript, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
  - ▶ Spécifications DOM, <http://www.w3.org/DOM/DOMTR>
- Pages Web
  - ▶ Comparatif syntaxe PHP/JavaScript, <http://phplens.com/phpewhere/node/view/30>
  - ▶ Tutorial JavaScript, <http://fr.selfhtml.org/javascript/>
- *JavaScript, la Référence*, O'Reilly

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 23 / 25

Gestionnaire d'événements Application

## Plan du cours

- 1 Introduction
- 2 L'objet Node
- 3 Fonctions utiles de JavaScript
- 4 Gestionnaire d'événements
- 5 Références
- 6 Application**

S.-D. Gouraud, P. Senellart (IFIPS) Introduction à JavaScript 28/11/2005 24 / 25

Reproduire les exemples de la page Web du cours :

- Vérification qu'un champ texte est bien rempli à la soumission d'un formulaire.
- Champs de formulaires apparaissant ou disparaissant suivant qu'un bouton radio est sélectionné ou non.
- Mise à jour dynamique de la couleur d'une boîte en fonctions de champs textes indiquant les pourcentages de Rouge, Vert, Bleu.