

Technologies du Web

JavaScript & contenus dynamiques riches



JavaScript

Introduction

Programmation

L'objet Node

Fonctions utiles de JavaScript

AJAX

Frameworks JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références



JavaScript

Introduction

Programmation

L'objet Node

Fonctions utiles de JavaScript

AJAX

Frameworks JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références





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/Flash/Silverlight (plus isolé du reste de la page).
- Rien à voir avec Java !

23 octobre 2012

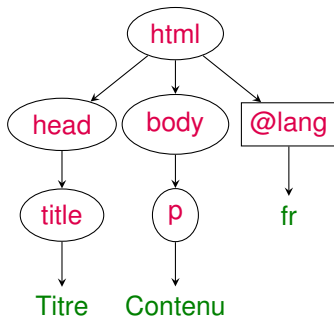




Document Object Model

Example

```
<html lang="fr" xmlns="...">  
  <head><title>Titre</title></head>  
  <body><p>Contenu</p></body>  
</html>
```





Langage JavaScript

- Langage de Programmation
- Script (ou programme) JavaScript : fichier texte, instructions terminées par des points-virgules
- Normalisé sous le nom d'**EcmaScript** (**JavaScript** est historiquement le nom de l'implémentation Netscape, **JScript** étant l'équivalent chez Microsoft)
- En pratique, actuellement, seuls les navigateurs graphiques supportent JavaScript. Conséquence : sauf applications complexes, **un site doit être utilisable sans JavaScript.**





Liaison avec HTML

- toto.js contenant des fonctions JavaScript (**function**)
- Dans le `<head>` du HTML :

```
<script src="toto.js" type="text/javascript"></script>
```

- Gestionnaires d'événement comme attributs des balises HTML (cf plus loin).
- On peut aussi mettre directement du code JavaScript à l'intérieur des balises `<script>` mais :
 - Mélange de plusieurs langages dans le même document : ajoute de la complexité.
 - Le document complet doit rester du HTML/XHTML valide !
 - Impossible de partager les mêmes fonctions dans plusieurs pages Web sans les recopier à chaque fois.
- Dernière possibilité : utiliser le pseudo-protocole **javascript:** dans un lien.

23 octobre 2012



JavaScript

Introduction

Programmation

L'objet Node

Fonctions utiles de JavaScript

AJAX

Frameworks JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références





Modèle objet

- JavaScript basé sur le **modèle de programmation 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()**, **avance(distance)**.

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

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

23 octobre 2012





JavaScript

Introduction

Programmation

L'objet Node

Fonctions utiles de JavaScript

AJAX

Frameworks JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références





Généralités

- L'objet **Node** est l'objet central du modèle DOM (Document Object Model).
- Chaque élément, chaque attribut et chaque donnée en caractères représentent des nœuds distincts. Ces nœuds forment une 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.
- La propriété **nodeType** permet de connaître le type du nœud (élément, attribut, texte, document, commentaire. . .)

Exemple

```
if (node.nodeType===Node.ELEMENT_NODE) {  
    ...  
}
```





Nœuds du document

- **document** est un objet prédéfini représentant le document actuel affiché dans le navigateur.
- `node = document.documentElement` met dans la variable `node` l'**élément racine** (c'est-à-dire le nœud correspondant à l'élément `html`)
- `node = document.getElementById("titi")` accède à un élément HTML qui possède un attribut `id` valant `titi`.
- `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
- Il existe aussi une fonction `document.getElementsByTagName` qui renvoie un tableau d'éléments.



Manipulation CSS

- `node.className="nouvelle_classe"` pour changer le nom de la classe CSS à laquelle appartient le nœud.
- `node.style.borderStyle="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'unions sont remplacés par une majuscule sur la lettre suivante. Les valeurs des propriétés en JavaScript sont identiques aux valeurs CSS (mais doivent être mis entre guillemets).





Exemple

JavaScript :

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

HTML :

```
<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 change la propriété CSS *color* du paragraphe, de telle sorte que le paragraphe perde sa couleur rouge et devienne bleu.





Parcours d'arbre

node.parentNode nœud parent d'un nœud.

node.childNodes 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 le nom d'un nœud.





Manipulation d'arbre

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

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

`node.cloneNode()` construit une copie à l'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")` renvoie la valeur d'un attribut déterminé dans un élément.



JavaScript

Introduction

Programmation

L'objet Node

Fonctions utiles de JavaScript

AJAX

Frameworks JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références





Fonctions utiles

`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

Ces fonctions sont en fait des méthodes de l'objet `window`, un objet prédéfini correspondant à la fenêtre du navigateur.

23 octobre 2012





Exemple

Exemple

JavaScript :

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

HTML :

```
<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.





Gestionnaires d'événements

- Les événements sont des actions de l'utilisateur, qui vont pouvoir donner lieu à une **interactivité**.
- L'événement correspond à un clic de souris, une sélection d'un champ, une touche pressée au clavier. . .
- Il est possible d'associer des fonctions à des événements.
- Dans le gestionnaire d'événements, **this** représente le nœud courant.
- La syntaxe est la suivante :

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





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 type="radio">` , `<input type="checkbox">`) associé à l'événement. Si le gestionnaire d'événement **renvoie** `false` (avec `return 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 d'événement **renvoie** false, le formulaire n'est pas soumis.





Exemple

```
<form name="Test" action="">
  <input type="text"
    onfocus="this.value='Entrez votre nom ici'">
  <input type="text"
    onfocus="this.value='Entrez votre adresse ici'">
  <input type="text"
    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.





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"



JavaScript

Introduction

Programmation

L'objet Node

Fonctions utiles de JavaScript

AJAX

Frameworks JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références



- Asynchronous Javascript and XML
- Pas une technologie, mais un ensemble de technologies, comme DHTML :
 - DHTML HTML+CSS+JavaScript
 - AJAX HTML+CSS+JavaScript+XML+XMLHttpRequest
- Partie importante : la classe JavaScript XMLHttpRequest
- Permet des échanges à l'intérieur d'une même page Web entre code JavaScript et programme tournant sur un serveur
- Asynchrone : données traitées quand elles arrivent, de manière non bloquante
- XML : données renvoyées par le serveur en général en XML (cf. cours ultérieur)
- Utile pour autocomplétion, Webmails, rafraîchissement des informations d'une partie de la page, etc.





Support dans les navigateurs

- Introduit par Microsoft dans IE5, sous une forme un peu instable
- Très bonne idée ! D'autres manières de faire la même chose (p. ex., `<iframe>`), mais beaucoup moins pratique.
- Repris par l'ensemble des navigateurs graphiques depuis, sous une forme simplifiée et standardisée (y compris IE7)
- Travail en cours du W3C pour la normalisation (a posteriori)





Principe d'utilisation

- On crée un objet **XMLHttpRequest** en lui donnant l'URL d'un script (avec éventuellement des paramètres) à contacter.
- On associe à cet objet une fonction qui traitera les données récupérées. En général, les données récupérées sont sous forme **XML**, et on les traite en JavaScript avec les fonctions de **manipulation DOM**, comme on traiterait du HTML.
- On envoie la requête.
- Une fois la requête terminée, la fonction de traitement s'exécute (de manière **asynchrone**).





Exemple d'utilisation de XMLHttpRequest

Exemple

```
request = new XMLHttpRequest();
request.open("GET", url);

request.onreadystatechange = function() {
  if (request.readyState == 4 && request.status == 200) {
    result=request.responseXML;
    /* faire quelque chose avec responseXML */
  }
};
request.send(null);
```





Compatibilité avec Internet Explorer 5 et 6

```
request=null;
if(typeof XMLHttpRequest != "undefined")
    request = new XMLHttpRequest();
else {
    try {
        request = new ActiveXObject("Msxml2.XMLHTTP.6.0");
    } catch(e) {
        try {
            request = new ActiveXObject("Msxml2.XMLHTTP.3.0");
        } catch(e) {
            try {
                request = new ActiveXObject("Msxml2.XMLHTTP");
            } catch(e) {
                request = new ActiveXObject("Microsoft.XMLHTTP");
            }
        }
    }
}
```





Fonctionnalités JavaScript utiles

- Minuterie en JavaScript (utile pour retarder l'exécution d'une requête, en attendant de voir si une autre requête ne doit pas être lancée tout de suite après)

```
timer=window.setTimeout("fonction_js()",200); /* 200ms */  
window.clearTimeout(timer);
```

- `document.createTextNode(data)` pour créer un nouveau nœud textuel
- `document.importNode(element, true)` pour pouvoir transférer un élément d'un document à un autre (d'un document XML au document HTML courant par exemple)
- Des **variables globales** peuvent être déclarées et utilisées à travers plusieurs fonctions en mettant (p. ex.) `toto=null` en dehors de toute fonction.



JavaScript

Introduction

Programmation

L'objet Node

Fonctions utiles de JavaScript

AJAX

Frameworks JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références





Support de JavaScript

- **Très grande disparité** dans le support JavaScript des différents navigateurs.
- La norme EcmaScript n'est arrivée que très tardivement.
- Comme d'habitude, IE est le plus éloigné de la norme, mais même au sein de Firefox/Opera/Safari, nombreuses différences de comportement.
- Implémentation encore en cours de nombreuses fonctionnalités.
- Présenté dans ce cours : fonctionnalités élémentaires, fonctionnant dans tous les navigateurs.





Frameworks JavaScript

- Bibliothèques de fonctions JavaScript permettant de travailler à un plus haut niveau et d'abstraire les différences de comportement des navigateurs.
- Fonctionnalités :
 - Détection du type de navigateur
 - Effets d'animation
 - Interactions complexes (p. ex., cliquer/glisser)
 - Meilleure gestion des événements
 - AJAX
 - Outils prêts être utilisés comme calendriers, tables dynamiques, etc.
- Important de vérifier quelles versions de navigateurs sont pris en charge !





Quelques noms de frameworks

- Yahoo! User Interface Library
- Prototype/Scriptaculous/Rico
- jQuery/jQuery UI
- Qooxdoo
- Dojo
- voir aussi certains frameworks d'applications Web (notamment, Google Web Toolkit)



JavaScript

Contenus multimédias riches

Applets Java

Adobe Flash

Alternatives

Son et vidéo sur le Web

Outils et références



- Le Web n'est plus seulement utilisé pour présenter des informations textuelles, mais héberge de véritables **applications**.

Exemples

- Messagerie électronique
 - Applications bureautiques
 - Cartographie
 - Jeux
 - Contenu multimédia
-
- En général, applications conçues uniquement pour fonctionner avec des **navigateurs graphiques**.
 - HTML, CSS, et même JavaScript n'ont pas **du tout** été conçus pour ça : d'autres technologies sont (souvent) nécessaires.





Web 2.0

- **Web 2.0**, terme marketing ne correspondant pas à un concept technique précis, mais recouvrant :
 - Phénomène **social**. Un Web dans lequel l'utilisateur **contribue à la production de contenu** (wikis, blogs et systèmes de commentaires, partage de fichiers multimédias, réseaux sociaux. . .)
 - Phénomène **technique**. Des interactions de plus en plus riches à **l'intérieur du navigateur** : applications Web similaires aux applications sur ordinateur de bureau, complétion automatique, etc.
- En particulier, interactions utilisant souvent l'ensemble de technologies **AJAX**.



Dangers potentiels

- **Accessibilité restreinte** . . .
- . . . justifiable quand il s'agit de construire des applications complexes, mais **pas pour présenter du contenu textuel** !
- Perte des fonctionnalités habituelles d'un navigateur (marque-pages, historique, gestion des mots de passe, etc.)
- Quid des autres agents Web, en particulier **robots des moteurs de recherche** ?
- **Complexification** du développement Web (empilement de technologies), entraînant potentiellement bugs, problèmes de conception et de maintenance
- Plug-ins à ajouter à un navigateur : multiplication des **failles de sécurité potentielles**



JavaScript

Contenus multimédias riches

Applets Java

Adobe Flash

Alternatives

Son et vidéo sur le Web

Outils et références





Le langage Java

- **Java** : langage de programmation **orienté objet**, utilisé pour écrire des logiciels complets
- Syntaxe proche de JavaScript et PHP, mais plus rigoureuse (langage fortement typé, toutes les variables doivent être déclarées, etc.)
- Langage **compilé** (et non interprété comme JavaScript ou PHP) vers du code pour une **machine virtuelle**
- Nécessite donc un environnement de développement ou **JDK** (pour transformer le code source en code pour machine virtuelle) et un environnement d'exécution ou **JRE** (pour exécuter le code de la machine virtuelle)
- Implémentation de référence : **Sun**, récemment devenue libre
- JRE disponibles pour de très nombreuses plate-formes, y compris téléphones mobiles basiques
- Assez lourd !



Applets Java

- Programme Java à l'intérieur du navigateur dans une zone de taille fixe
- Intégration via un plugin de navigateur, à installer séparément (disponible pour tous les navigateurs graphiques modernes)
- Intégration avec HTML :

```
<applet code="toto.class" width="300" height="400">  
  <param name="param" value="value">  
  <!-- Contenu alternatif -->  
</applet>
```

- `<applet>` est normalement déprécié en faveur de `<object>`, mais pas de manière simple de faire fonctionner des applets Java avec `<object>` dans tous les navigateurs graphiques
- Accès à toute la bibliothèque de fonctions standards du langage Java (conséquente) et possibilité d'utiliser les très nombreuses bibliothèques supplémentaires existantes





Système de confiance

Deux niveaux de confiance en une applet :

- Par défaut, une applet (tout comme JavaScript, Flash. . .) est confinée dans un **bac à sable**, n'a pas accès à l'extérieur du navigateur (système d'exploitation, contenu du disque dur de l'ordinateur, etc.)
- Une **trusted applet** (demande de confirmation faite par le navigateur) aura par contre les mêmes droits d'accès qu'un programme classique s'exécutant sur la machine locale.





Intérêts/inconvénients

- Assez ancien, et initialement conçu comme **la** façon d'avoir une application complexe à l'intérieur d'un navigateur
- Bon support pour divers navigateurs et plates-formes, mais à condition d'installer le plugin et le JDK
- Assez lourd à mettre en place, les applets mettent en général du temps à démarrer
- Trusted applets : Bonne manière de déployer une application lourde sur plusieurs plates-formes (cf. Impôt sur le revenu)
- Possibilités graphiques existantes, mais non mises en avant



JavaScript

Contenus multimédias riches

Applets Java

Adobe Flash

Alternatives

Son et vidéo sur le Web

Outils et références





Flash

- Adobe Flash (anciennement, Macromedia Flash)
- À l'origine prévu pour des **animations vectorielles** : images vectorielles + **ActionScript** (basé sur ECMAScript)
- Comme les applets Java, s'exécute dans une zone délimitée du navigateur
- Programmes flash compilés dans un format binaire (SWF)
- Support correct mais pas exceptionnel :
 - Pas de support pour iOS et d'autres smartphones
 - Pas de support des Unix exotiques : AIX, HP-UX, OpenBSD, etc.
- La description du format SWF est publique, mais jusque mai 2008, Adobe interdisait de l'utiliser pour créer des lecteurs Flash : pas d'alternative libre crédible au plugin officiel Adobe Flash.
- Environnement de développement : payant (octobre 2009 : 699\$ pour la version professionnelle).
- Développement du plugin Flash abandonné pour les smartphones





- Langage de programmation permettant de créer des applications Flash, de manière programmatrice (et non plus graphique comme traditionnellement)
- Rien ne change pour l'utilisateur
- Possibilité
- Compilateur (mais pas environnement de développement) libre mis à disposition par Adobe





Intégration avec HTML

Avec la balise `<object>` :

```
<object data="toto.swf"  
  type="application/x-shockwave-flash"  
  width="300" height="400" >  
  <param name="param" value="value">  
  Contenu alternatif  
</object>
```

Méthode standard, peut nécessiter des workarounds pénibles pour Internet Explorer (utilisation de `<embed>`)



JavaScript

Contenus multimédias riches

Applets Java

Adobe Flash

Alternatives

Son et vidéo sur le Web

Outils et références





- Technologie Microsoft assez récente (première version décembre 2006)
- Alternative à Flash, fonctionne sur le même principe et suivant le même modèle
- Support encore très léger (Windows et MacOS X uniquement, pas tous les navigateurs)
- Programmation reposant sur l'architecture .NET de Microsoft



- Technologie Microsoft
- Plugins du navigateur permettant une interaction très forte avec le système d'exploitation
- Pas de système de bac à sable, mais l'exécution des plugins doit être confirmée par l'utilisateur
- Uniquement sous Windows, principalement sous Internet Explorer
- Un peu passé de mode maintenant, tout comme les applets Java qu'ActiveX était censé concurrencer
- Utilisé pour Windows Update



- Balise `<canvas>` définissant une zone permettant de faire du dessin bitmap
- JavaScript permet de rendre le contenu dynamique
- Fait partie de HTML 5
- Supporté dans Safari, Opera, Firefox, Chrome, IE9
- Exemples :
 - <http://www.abrahamjoffe.com.au/ben/canvascape/>
 - <http://radikalfx.com/files/collage/demo.html>
 - <http://www.canvasdemos.com/>



- Formats d'images vectorielles, manipulables en JavaScript
- Scalable Vector Graphics
- Vector Markup Language
- SVG : cf. cours sur XML, supporté (à des degrés variés) dans Firefox, Opera, Safari, Chrome, IE9 ; Adobe SVG Plugin pour Internet Explorer ≤ 8
- VML : Internet Explorer uniquement
- Encore un peu expérimental
- SVG+JavaScript : Peut être vu comme une alternative à Flash, dans des formats ouverts et standards
- Utilisé par exemple dans Google Maps pour l'affichage des itinéraires (VML dans IE, SVG ailleurs)
- **Exemple** : <http://isthis4real.com/orbit.xml>





Et aussi...

Shockwave initialement une contrepartie à Flash dans l'offre de Macromedia, maintenant utilisée surtout pour des jeux interactifs et environnements 3D

VRML, WebGL Images et modèles 3D

...



JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références





HTML et contenu multimédia

- HTML ne propose que la représentation de texte (structuré) et d'images (``)
- Rien pour le son et la vidéo ! (à part un `<bgsound>` , non standard, à oublier)
- Solution standard : recourir au mécanisme général de `<object>`
- HTML5 introduit les balises `<audio>` et `<video>`





Conteneurs et codecs

Codecs audio MPEG-1/I, MPEG-1/II, MPEG-1/III (MP3), AAC, AC3, RealAudio. . .

Manière dont des données sonores sont compressées (avec ou sans perte) par blocs

Codecs vidéo MPEG-2, H.264, Theora, RealVideo, DivX, VP8. . .

Manière dont des séquences vidéo sont compressées (avec ou sans perte) par blocs

Conteneurs WAV, ASF. . . (son), AVI, MPEG-2, MPEG-4, MOV, OGG, RealMedia, DivX, WebM. . . (son+vidéo)

Manière dont données sonores et séquences vidéo compressées par des codecs sont agencés dans un fichier, avec des fonctionnalités permettant l'entrelacement son/vidéo, le positionnement aléatoire. . .

Beau mic-mac! Difficile de trouver un logiciel supportant tous ces formats. . .

23 octobre 2012





Brevets logiciels et restrictions d'utilisation

- Nombreux **brevets logiciels** protégeant l'encodage ou le décodage selon tel ou tel codec (licence accordée contre redevances)
- Peu clair quels brevets sont vraiment valides ! De nombreuses entreprises prétendent avoir un brevet sur la compression/décompression MP3.
- **Note** : les brevets logiciels n'existent (normalement) dans l'UE que s'ils décrivent un **processus industriel**. Peu clair si ça s'applique à l'encodage/décodage son et vidéo.
- **Conséquence** : un logiciel libre comme vlc est peut-être illégal ! Peu de chance de voir un support vidéo et audio pour des formats propriétaires (H.264/MPEG-4) dans des navigateurs libres
- **Exempts** de ces problèmes :
 - Les **vieux** formats (RIFF, MPEG-1...)
 - Les formats **conçus comme libres** (Ogg, Vorbis, Theora, VP8, WebM) ; mais certaines entreprises ont peur que ces formats violent quand même des brevets existants non encore connus !

21 octobre 2012





Alternatives pour son et vidéo sur le Web (1/2)

- Balise `<embed>` : non standard, vieillerie, à oublier
- Utilisation d'un plugin spécifique à un type de média donné (ne marche pas toujours avec IE, nécessite le plugin !)

```
<object type="video/quicktime" data="test.mov"  
    width="320" height="240">  
  <a href="test.mov">test.mov</a>  
</object>
```

- Utilisation d'un contrôle ActiveX spécifique (IE seulement)

```
<object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"  
    width="320" height="240">  
  <param name="src" value="test.mov" >  
  <a href="test.mov">test.mov</a>  
</object>
```





Alternatives pour son et vidéo sur le Web

(2/2)

- Simple lien vers le contenu (fonctionne si un logiciel correspondant est installé, mais **pas d'intégration au navigateur**)

```
<a href="test.mov">test.mov</a>
```

- Balises `<audio>` et `<video>` (HTML 5, implémentées dans les versions les plus récentes d'Opera/Chrome/Firefox/Safari + IE9, avenir prometteur, mais incertitudes sur les formats ; possible de référencer plusieurs formats)

```
<video src="test.mov" width="320" height="240">  
  <a href="test.mov">test.mov</a>  
</video>
```

Demo : <http://htmlfive.appspot.com/static/video.html>

- Intégré à une application Flash ; fonctionne... si Flash fonctionne

23 octobre 2012





Meilleure solution

- Utiliser le tag `<video>` pour les navigateurs modernes avec formats multiples (H.264 pour Safari/Android, Theora pour les anciennes versions de Chrome/Firefox/Opera, WebM pour les versions récentes de Chrome/Firefox/Opera et IE9 quand un plugin est installé)
- Utiliser Flash si `<video>` n'est pas supporté
- Un lien si ni Flash ni `<video>` n'est pas supporté





Au final...

```
<video id="movie" width="320" height="240" preload controls>
  <source src="pr6.webm" type='video/webm; codecs="vp8, vorbis"' />
  <source src="pr6.ogv" type='video/ogg; codecs="theora, vorbis"' />
  <source src="pr6.mp4" />
  <object width="320" height="240" type="application/x-shockwave-flash"
    data="flowplayer-3.2.1.swf">
    <param name="movie" value="flowplayer-3.2.1.swf" />
    <param name="allowfullscreen" value="true" />
    <param name="flashvars" value='config={"clip": {"url":
"http://wearehugh.com/dih5/pr6.mp4", "autoPlay":false,
"autoBuffering":true}}' />
    <p>Download video as <a href="pr6.mp4">MP4</a>, <a
href="pr6.webm">WebM</a>, or <a href="pr6.ogv">Ogg</a>.</p>
  </object>
</video>
```

Tiré de <http://diveintohtml5.info/video.html>

23 octobre 2012



JavaScript

Contenus multimédias riches

Son et vidéo sur le Web

Outils et références



- **Multi**ples navigateurs graphiques.
- N'importe quel éditeur de texte.
- Console JavaScript de nombreux navigateurs.
- Extension **Firebug** de Firefox (Console, inspecteur DOM, débogueur JavaScript)





- Standards (beaucoup moins lisibles que pour HTML ou CSS)
 - Langage ECMAScript, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - Spécifications DOM, <http://www.w3.org/DOM/DOMTR>
- *HTML5, Up and Running*, O'Reilly
- *HTML5, Canvas*, O'Reilly





Licence de droits d'usage



Contexte public } avec modifications

Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après et à l'exclusion expresse de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage à destination de tout public qui comprend :

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document au public sur support papier ou informatique, y compris par la mise à la disposition du public sur un réseau numérique,
- le droit de modifier la forme ou la présentation du document,
- le droit d'intégrer tout ou partie du document dans un document composite et de le diffuser dans ce nouveau document, à condition que :
 - L'auteur soit informé.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel et non exclusif.

Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : sitopedago@telecom-paristech.fr

