

# Les nouvelles technologies du Web

Paris Web 2008

Laurent Jouanneau

Conditions d'utilisation de ce document :  
[CC-nc-sa \(http://creativecommons.org/licenses/by-nc-sa/3.0/deed.fr\)](http://creativecommons.org/licenses/by-nc-sa/3.0/deed.fr)

# Qui suis-je ?

- "Senior Software Engineer" chez [Disruptive Innovations](http://www.disruptive-innovation.com) (<http://www.disruptive-innovation.com>) (W3C)
- Domaines de compétences :
  - Technologie Mozilla ([xulfr.org](http://xulfr.org))
  - Standards du web ([openweb.eu.org](http://openweb.eu.org))
  - Développement web et PHP

# Sommaire

- Médias avec HTML5
- Nouvelles API de scripts
- Technologies annexes : CSS3, Mathml, XBL...
- Dans un futur plus lointain...

# Avertissement

Les spécifications des technologies montrées sont encore sous forme de brouillon au W3C, ne sont pas encore des recommandations, mais commencent tout de même à être implémenter dans les navigateurs.

# Médias avec HTML5

# Objectifs de HTML5

- Répondre aux besoins d'hier...
  - amélioration des formulaires
  - de la sémantique..
- ... d'aujourd'hui ...
  - medias
  - ajax, DHTML...
- ... et de demain
  - mode déconnecté (offline/online)
  - template, datagrid, server events...

# HTML5 vs XHTML5

- Même spécification
- Règles de Syntaxe différente (SGML vs XML)
- XHTML : xml:base, namespaces, SVG, MathML, RDFa...
- XHTML2 : plus ou moins mort

## <canvas>

Zone dans laquelle on peut dessiner via une API.

- getContext()
- Context "2d", "3d" (dans le futur ?)
- Un context = objet qui fournit des méthodes pour dessiner
- figures simples, rotation, transformation, zoom, courbe de bezier, path, gradient, texte, ombres...

safari 3 / webkit/ opera/ firefox

# Demo <canvas> #1

Dessiner



```
<canvas id="canvas1"
  width="100" height="100"></canvas>
```

```
var ctx = document
  .getElementById("canvas1")
  .getContext("2d");
```

```
ctx.fillStyle = "rgb(200,0,0)";
ctx.fillRect (10, 10, 55, 50);
```

```
ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
ctx.fillRect (30, 30, 55, 50);
```

## Demo <canvas> #2

- [Demos canvas + souris](http://ljouanneau.com/lab/html5/canvas/MouseMove/canvas.html)  
(<http://ljouanneau.com/lab/html5/canvas/MouseMove/canvas.html>)
- [Lightweight Visual Thesaurus](http://awordlike.textdriven.com/)  
(<http://awordlike.textdriven.com/>)
- [canvas painter](http://caimansys.com/painter/)  
(<http://caimansys.com/painter/>)
- [ray caster](https://developer.mozilla.org/samples/raycaster/RayCaster.html)  
(<https://developer.mozilla.org/samples/raycaster/RayCaster.html>)

## **<embed>**

- Contenu externe pris en charge par un plugin
- Différent de <object>. <object> a le même rôle que <img>, <iframe>, et <embed> reuni, avec possibilité de paramètres(<param>)

## <video>

```
<video src="mavideo.ogv">
```

- **Contrôle de la lecture** : `start`  
`end` `loopstart` `loopend`  
`playcount` `autoplay`
- **Affichage de l'interface de contrôle du navigateur** :  
`controls`
- **Interface que l'on veut grâce à l'API normalisée** : `play()`,  
`pause()` ..
- **attributs** `poster`

## **<video> #2**

- Meilleure accessibilité que `<object>`
- Plein écran possible, mais seulement à la discretion de l'utilisateur et du navigateur (pas d'API)
- Streaming

## <video> : format

- pas de format préconisé
- decodeur embarqué et/ou utilisation des decodeurs du système
- ex Mozilla: décodeur Ogg theora embarqué et plus tard : DirectShow, Quicktime, Gstreamer..
- possibilité d'indiquer plusieurs formats : <source>

## <video> : compatibilité

- le contenu de l'élément  
<video> : pour les navigateurs  
ne prenant pas en charge  
video

- ```
<video src="mavideo.ogg">
  <p>Votre navigateur ne supporte pas l'element video</p>
  <object>
    <param name="movie"
      value="flvplayer.swf?file=mavideo.flv">
    </param>
    <embed src="flvplayer.swf?file=mavideo.flv"
      type="application/x-shockwave-flash"></embed>
  </object>
</video>
```

## <video> : demos

- [http://www.double.co.nz/video\\_test/](http://www.double.co.nz/video_test/)

## Autres nouveautés

- script : attr defer (FF 3.1)
- attribut contenteditable : pas mal buggé sur la majorité des navigateurs (safari, mozilla)

# Nouvelles API scriptables

# Évènements online, offline

- Envoyés lorsque le statut de la connexion change
- Envoyés sur window, document, document.body
- `navigator.onLine` : indique le statut de la connexion (FF2, IE)
- `addEventListener` on the window, document, or document.body
- [demo \(http://ljouanneau.com/lab/html5/offline.html\)](http://ljouanneau.com/lab/html5/offline.html)

IE8 / Firefox

## ressources offline

- fichier manifest indiquant les ressources offline
- `<html manifest="fichier.manifest">`
- utf8, "text/cache-manifest"

### CACHE MANIFEST

CACHE:

```
# v1
```

```
# This is a comment.
```

```
http://www.foo.com/index.html
```

```
http://www.foo.com/header.png
```

FALLBACK:

NETWORK:

# Storage

- sessionStorage, localStorage
- .length .key(index),  
getItem(key), setItem(key,  
data), removeItem(key),  
clear()
- globalStorage['domain.com']  
(only FF, not html5)
- [demo \(http://ljouanneau.com  
/lab/html5/storage.html\)](http://ljouanneau.com/lab/html5/storage.html)  
FF2, trunk webkit, IE8b2

# drag drop

- `draggable="true"` sur les éléments sur lesquels un drag'n'drop peut démarrer
  - Évènements sur l'élément de départ : `dragstart`, `drag` (n fois), `dragend`
  - Évènement sur les éléments que l'on survole : `dragenter`, `dragover` (n fois), `dragleave`
  - Évènement sur l'élément sur lequel on relache le bouton de la souris : `drop`
- FF3.1, trunk webkit, IE5

# drag drop : objet dataTransfer

- `getData`, `setData` : stockage des données à transférer selon leur type (text, liens, html, fichier, image...)
- `setDragImage` : spécifier l'image du drag drop
- `effectAllowed` : effet du drag drop (copy, move, link, copymove...)

# drag drop : demo

- [demo \(http://ljouanneau.com/lab/html5/demodragdrop.html\)](http://ljouanneau.com/lab/html5/demodragdrop.html)

# querySelector

- `querySelector("selecteur css")`
- `querySelectorAll("selecteur css")`
- <http://www.w3.org/TR/selectors-api/>
- (FF 3.1)  
FF3.1, webkit, IE8

# Contrôle d'accès

- Ensemble d'entête HTTP pour contrôler l'accès à une ressource
- Le client peut indiquer ce qu'il veut faire d'une ressource
- Le serveur peut indiquer ce qu'il peut faire de la ressource demandée
- Utilisé dans le cadre de XMLHttpRequest, HTML5, XBL2...

IE8 (via XDomainRequest), FF3.1, webkit en cours

# Contrôle d'accès : Réponse HTTP

En-têtes contrôlés par le serveur, et vérifiés par le navigateur

- Access-Control-Allow-Origin
- Access-Control-Max-Age
- Access-Control-Allow-Credentials
- Access-Control-Allow-Methods

# Contrôle d'accès : Réquêtes HTTP

En-têtes envoyés par le navigateur, et éventuellement prise en compte par le serveur.

- Origin
- Access-Control-Request-Method
- Access-Control-Headers

# Géolocalisation

- API pour connaître la position géographique de l'utilisateur
- Repose sur les capacités du périphérique
- Le navigateur est censé demandé l'autorisation à l'utilisateur

FF3.1, webkit en cours

# Géolocalisation : position

Dernière position  
enregistrée :

```
var pos = navigator.geolocation.lastPosition;  
pos.latitude  
pos.longitude  
pos.altitude  
pos.accuracy  
pos.altitudeAccuracy  
pos.heading  
pos.speed  
pos.timestamp
```

# Géolocalisation : méthodes

```
navigator.geolocation.getCurrentPosition(function(position) {
    do_something(position.latitude, position.longitude);
});
```

```
var watchID = navigator.geolocation.watchPosition(function(position) {
    do_something(position.latitude, position.longitude);
});
navigator.geolocation.clearWatch(watchID);
```

# Worker Threads

- execute un script js en tâche de fond
  - pas de manipulation sur le document
  - xmlhttprequest possible
  - envoi de message entre processus maitre et fils
- FF3.1, webkit en cours

# HTML5: Cross domain messaging

- Permettre de communiquer entre deux documents d'un domaine différent
- Ex: doc A qui a une iframe contenant un doc B
- `window.postMessage('mon message', 'http://mondomaine')`

# HTML5: Cross domain messaging #2

```
//doc A
var o = document.getElementsByTagName('iframe')[0];
o.contentWindow.postMessage('Hello world', 'http://b.e

//doc B
window.addEventListener('message', function (e) {
  if (e.origin == 'http://example.com')
    if (e.data == 'Hello world') {
      e.source.postMessage('Hello', e.origin);
    } else { alert(e.data); }
}, false);
```

## Autres

- focus :  
document.activeElement,  
document.hasFocus() (FF3)
- ElementTraversal :  
firstElementChild,  
lastElementChild,  
previousElementSibling,  
nextElementSibling,  
childElementCount,
- JSON object

# Technologies annexes

# SMIL

- Synchronized Multimedia Integration Language
- SMIL = langage XML pour décrire la manière d'afficher plusieurs éléments graphiques/multimedia dans l'espace et le temps
- -> SVG, XHTML, Mathml, videos, images...
- Pour des animations "informatives": slides, presentations

## SMIL #2

- Implémenté dans IE 5.5, IE6, Realplayer, Quicktime et nombreux playeurs multimedia
- Implémentation en cours dans webkit et Firefox
- <http://www.w3.org/AudioVideo/>
- [www.multimedia4everyone.com](http://www.multimedia4everyone.com): des presentations de SMIL en SMIL...
- Version 3 en cours de spécification

# SVG

- Format XML pour des images vectorielles
- Nombreux effets graphiques et filtres possibles
- Intégration d'autres éléments non SVG via `<foreignobject>`
- Implémenter dans de nombreux logiciels. Ex: Inscap, éditeur graphique
- Implémenté (en partie) dans Firefox, webkit (en cours), Opera. Vieux plugin Adobe pour IE
- <http://www.w3.org/TR/SVG11/>

# Animations SVG

- Reprends beaucoup de fonctions animations de SMIL
- Principalement pour des animations "informatives" alors que les animations CSS : pour de l'animation purement décorative
- Implémentation en rapport très étroit avec l'implémentation de SMIL : implémentation en cours de webkit et Firefox.
- Fait parti du test acid3
- <http://www.w3.org/TR/SVG11/animate.html>

## CSS 2.1/3

Implémentation en cours  
ou finalisée dans Firefox  
(2,3,3.1), webkit (3,3.x),  
et Opéra

- selecteurs CSS 3, CSS media queries
- text-shadow, box-shadow
- border-image
- column
- transformations, animations (uniquement webkit)
- fontes téléchargeables
- [Conférence de Daniel Glazman à PW2008](http://disruptive-innovations.com/zoo/slides/20081114-PW2008/pw2008.xhtml) (<http://disruptive-innovations.com/zoo/slides/20081114-PW2008/pw2008.xhtml>)
- [Quelques démos](http://ljouanneau.com/lab/css3/) (<http://ljouanneau.com/lab/css3/>)

# Javascript 1.7, 1.8...

- generateur
- itérateur
- let
- Assignement déstructuré:  
[a, b] = [b, a] = f()
- expression closure

# MATHML

- Langage XML pour écrire des formules mathématiques et scientifiques
- Implémenté dans Firefox 1.0 et suivant.
- [Exemples \(http://www.albany.edu/~hammond/gellmu/dmml.xml\)](http://www.albany.edu/~hammond/gellmu/dmml.xml), <http://www.w3.org/Math/>

$$x = \frac{-b \pm (b^2 - 4ac)^{1/2}}{2a}$$

# XBL2

- Un langage XML déclaratif
- Permet de modifier/ajouter sur des éléments d'un document :
  1. du contenu
  2. des comportements
  3. des styles CSS

## XBL2 : Utilisation

- Widgets, composants d'interface
- Améliorer des balises existantes
- Implémentation d'un langage XML non reconnu par le navigateur

## XBL2 : Avantages

- Ne "pollue" pas le document
- Facilite le développement :  
séparation entre le contenu et  
le comportement
- Réutilisation
- Solution plus légère que les  
équivalents en pure javascript

# XBL2 : en savoir plus

- [Ma présentation à ParisWeb 2007 \(http://ljouanneau.com/standards/pw2007/\)](http://ljouanneau.com/standards/pw2007/)
- <http://www.w3.org/TR/xbl/>

# Le futur...

# xmlHttpRequest

- Level 1 : spécification, harmonisation de l'existant
- Level 2 : améliorations.
  - propriétés `onabort`, `onerror`, `onload`, `onloadstart`, `onprogress`
  - `responseXML` : doit pouvoir retourner aussi un fragment DOM HTML
  - `overrideMimeType` (mime)
  - `responseBody` (?)
  - `upload` (?)

## HTML5 : <figure>

Illustration d'un texte.

```
<figure>
  
  <legend>Bubbles at work</legend>
</figure>
```

# HTML5 : Sections

- `<section>`, `<nav>`, `<article>`,  
`<aside>`, `<header>`, `<footer>`
- Juste de la sémantique

# HTML5 : Elements textuels

- `<mark>`, `<time>`,
- `<dfn>`
- `<meter value="" max="" min="">`
- `<progress value="" max="">80%</progress>`

# HTML5 : Formulaires #1

- validation automatique
- widgets supplémentaires
- spécifications encore en rédaction, aucune implémentation

## HTML5 : Formulaire #2

- **attributs** `type`, `autocomplete`, `list`, `pattern`, `min`, `max`, `step`, `autofocus`
- **Éléments** `<datalist>`, `<output>`
- **support de GET, POST, PUT, DELETE**
- **propriété** `validity` -> `ValidityState`

# HTML5 : Autres éléments et attributs

- groupes : dialog  
(=conversation) (même  
groupe que ul, ol, p, pre etc)
- details, datagrid, command,  
bb, menu,

# Le mot de la fin

## **Poussez ces nouveautés !**

Utilisez-les pour améliorer le web !

Cette présentation est disponible à

<http://ljouanneau.com/standards/pw2008/>