

Web 2.0

Web 2.0 qu'est-ce que c'est ?

Web « dynamique » :

Confort d'utilisation / confort visuel → Fluidité

Echanges asynchrones :

Communication serveur « en tâche de fond »

Ergonomie / style / Design:

Igoogle / Netvibes (webparts) / boutons « **flashis** » / effets miroirs...

Communauté/Réseau social / Partage; L'internaute peut AGIR sur le WEB
Flux RSS/Interconnections/echanges, avis (Exemple ->bookmarks ...)

Accessibilité → Respect des normes → Web « Semantique »

(CSS)

(XHTML)
Du sens

(Javascript)
Non intrusif

Dans l'idéal le site doit pouvoir
fonctionner sans CSS et sans JS

Web 2.0

Mais ...

Un terme abusif

contrairement aux termes techniques qui font références à des normes précises, WEB2.0 n'a pas de sens précis.

Beaucoup de réseaux sociaux :

(Tweeter, Youtube, Flickr, Facebook, Myspace, Facebook, copains d'avant)

.... Un gigantesque bruit de fond

Je n'ai rien à dire, mais je vais le faire savoir

Il existe des entreprises de « désinformation » et/ou de fabrication de 'e-réputation'



sommaire

JAVASCRIPT : DES OUTILS FACILITANTS

Zoom sur le concept de LightBox

Etude de la librairie jQuery

‘LightBox’

Ou « boîte de contenu virtuelle »



From Wikipedia, the free encyclopedia

Lightbox, and the newer **Lightbox 2**, is a **JavaScript** application used to display large images using **modal dialogs**. The script has gained widespread popularity due to its simple yet elegant style and easy implementation. While it was initially developed from scratch, Lightbox has since been modified to use a number of JavaScript libraries (such as the **Prototype Javascript Framework**^[1] and **script.aculo.us**^[2] for its animations and positioning), in order to reduce the size of the code.

Petit tutorial pour comprendre

Principe :

Un div qui prend toute la place écran pour griser le fond

Un div pour faire apparaître la LightBox

Du Javascript pour rendre visible et invisible

```
<body>

<div id="fond_ecran_gris" style="visibility:hidden;"></div>

<div id="ma_lightbox" style="visibility:hidden;">
  <h3>Contenu de<br>ma belle LightBox !</h3>
  <a href="#" onclick="fermer_boite();return false;">Fermer</a>
</div>

<a href="#" onclick="montrer_boite();return false;">
  Faire afficher la LightBox en cliquant ici.</a>

</body>
```

Les styles utilisés

Les styles utilisés

```
#fond_ecran_gris
{
    background-color: #CCCCFF; /* gris */
    /* pour avoir un effet de transparence utiliser opacity */
    position: absolute;
    left: 0px; top: 0px;
    width: 100%; height: 100%;
    z-index: 100; /* Pour recouvrir d'éventuelles zones */
}

#ma_lightbox
{
    position: absolute;
    /* pourrait dynamiquement être modifié par javascript */
    left: 150px; top: 150px;
    width: 200px; height: 150px;
    background-color: lightblue;
    z-index: 101; /* Pour être sur l'écran gris */
}
```

Le javascript pour montrer et cacher

Le javascript pour montrer et cacher

```
<script type="text/javascript" >
function montrer_boite()
{
    document.getElementById('fond_ecran_gris').style.visibility='visible';
    document.getElementById('ma_lightbox').style.visibility='visible';
}

function fermer_boite()
{
    document.getElementById('fond_ecran_gris').style.visibility='hidden';
    document.getElementById('ma_lightbox').style.visibility='hidden';
}
</script>
```

```

<style>
#fond_ecran_gris
{
    background-color: #CCCCFF;      opacity: .75; filter: alpha(opacity=75);
    position: absolute;      left: 0px;      top: 0px;
    width: 100%;height: 100%;      z-index: 100;
}
#ma_lightbox
{
    position: absolute;      left: 150px;top: 150px;
    width: 200px;height: 150px;      background-color: lightblue;
    z-index: 101;
}
</style>
<body>
<div id="fond_ecran_gris" style="visibility:hidden;"></div>
<div id="ma_lightbox" style="visibility:hidden;">
<h3>Contenu de<br>ma belle LightBox !</h3> <a href="#" onclick="fermer_boite();return false;">Fermer</a>
</div>
<a href="#" onclick="montrer_boite();return false;">Faire afficher la LightBox en cliquant ici.</a>
</body>
<script type="text/javascript" >
function montrer_boite()
{
    document.getElementById('fond_ecran_gris').style.visibility='visible';
    document.getElementById('ma_lightbox').style.visibility='visible';
}

function fermer_boite()
{
    document.getElementById('fond_ecran_gris').style.visibility='hidden';
    document.getElementById('ma_lightbox').style.visibility='hidden';
}
</script>

```

Source complet
DEMO
Cliquer ici

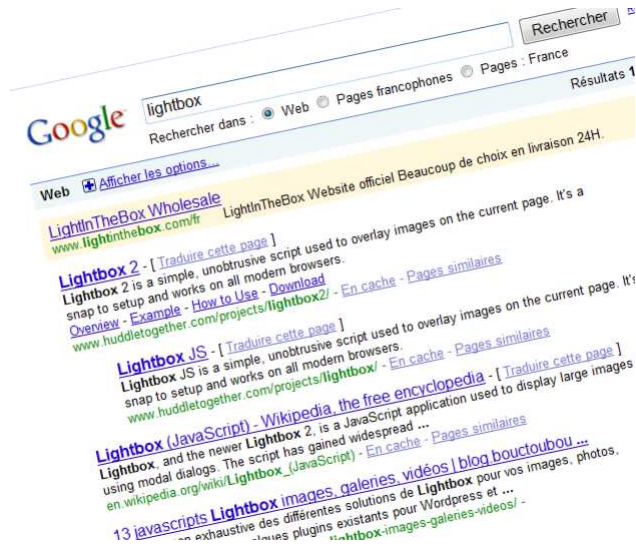
Un effet LightBox



Pourquoi "réinventer la roue" ?

Pourquoi "réinventer la roue" ?

Beaucoup de réponses pour la recherche sur lightbox



Une petite sélection à voir ...

Une petite sélection à voir ...

<http://planetozh.com/projects/lightbox-clones/>

<http://www.coders.me/ejemplos/sexy-lightbox-2/>

<http://www.moord.it/>

<http://blog.bouctoubou.com/13-javascripts-lightbox-images-galleries-vidéos/>

<http://www.dreamcss.com/2009/03/31-mind-blowing-collection-of-jquery.html>

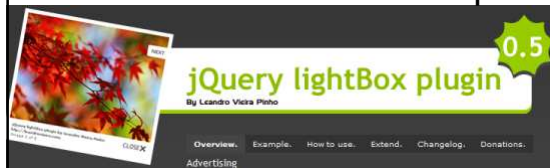
<http://www.smashingmagazine.com/2007/05/18/30-best-solutions-for-image-galleries-slides/>
(voir commentaires également)

<http://www.dottony.com/30-useful-ajax-lightbox-and-modal-dialog-solutions/>

<http://blog.insicdesigns.com/2009/01/10-promising-jquery-lightbox-plugin/>



Utiliser une bibliothèque s'appuyant sur jQuery



Une option raisonnable :
Utiliser une bibliothèque
s'appuyant sur **jQuery**

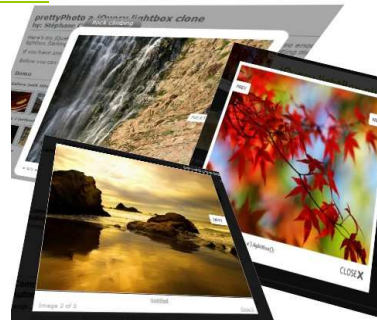
<http://leandrovieira.com/projects/jquery/lightbox/>

<http://colorpowered.com/colorbox/>

<http://www.ericmmartin.com/projects/simplemodal-demos/>
(voir demo formulaire de contact)

<http://jquery.com/demo/thickbox/>

↓
Demo locale



votre première librairie JavaScript



jquery.js est une librairie JavaScript
(appelé aussi quelquefois bibliothèque ou framework).

C'est un ensemble de fonctions, méthodes et propriétés
JavaScript dont l'objectif est de simplifier les développements
avec ce langage.

Il existe d'autres librairies

mootools
extjs
script.aculo.us
dojo
yui
qooxdoo
midori
uize
archetype
qooxdoo
...



Il n'existe pas une librairie « meilleure » que les autres car elles ont toutes leurs qualités.
On peut en utiliser plusieurs à la fois (à tester toutefois ; conflits possibles)

Statistiques d'utilisation

The Two Popular Javascript Frameworks

Right now, the two commonly used Javascript frameworks are: **jQuery** and **MooTools**. Check out the poll below:

POLL Your favorite Javascript framework?

Dojo (4%, 707 Votes)

jQuery (69%, 11,423 Votes)

MooTools (15%, 2,560 Votes)

Prototype (8%, 1,247 Votes)

Other.. (4%, 625 Votes)

Total Voters: **16,560**

Source:

<http://www.webdesignerwall.com/general/javascript-in-modern-web-design/>

Principales qualités de jQuery

- * Compatible avec tous les navigateurs
- * **Compacte et rapide** *(son créateur, John Resig est un des principaux développeurs du moteur JavaScript de Firefox)*
- * **Syntaxe élégante et peu verbeuse**
- * **Documentation à jour**
- * **Richesse d'extensions développées**

(Choisi par Microsoft pour ASP.NET : <http://msdn.microsoft.com/fr-fr/magazine/dd453033.aspx>)

Un premier essai

Utilisation de l'objet jQuery **\$** (ou fonction)

```
<head>
<script type="text/javascript" src="jquery-1.3.2.js" /></script>
</head>

<body>

<h2>alert("Version de jQuery : " + $.jquery);</h2>

<script type='text/javascript'>
alert("Version de jQuery : " + $.jquery);
</script>

</body>
```

[Demo](#)

Fonction jQuery ready()

Ce qu'on peut voir souvent

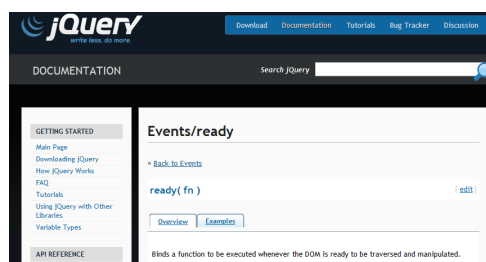
```
<h1 id="un_id" onclick="alert('click !');">du texte</h1>
```

Ce qui est moins « intrusif »

```
var x = document.getElementById('un_id');  
x.onclick = function () { alert('click !'); }
```

A condition que le D.O.M soit chargé

```
window.onload = function(){ ... }
```



Fonction **jquery ready()**

```
$(document).ready( function(){ ... } );
```

<http://docs.jquery.com/Events/ready>

L'objet \$

La fonction **\$** est un alias de la fonction `jQuery()`

L'objet **\$** permet d'accéder à un élément de l'arbre DOM, puis de travailler sur celui-ci via des fonctions jQuery.

Si on utilise d'autres librairies, il peut être nécessaire d'éviter les conflits d'utilisation du nom **\$**

```
jQuery.noConflict();
```

A l'appel de cette fonction, "**\$**" ne sera plus considérée comme l'alias de jQuery, il faudra utiliser le nommage **jQuery**.

Utilisation de la fonction \$ avec des arguments



`$('#monId')`

Ramène l'objet jQuery correspondant à l'élément d'identifiant 'monId'

`$('.uneClasse')`

Ramène le(s) objet(s) jQuery correspondants aux élément(s) de classe 'uneClasse'

`$('p')`

Ramène tous le(s) objet(s) jQuery correspondants aux éléments 'P'

`$('p.titre')`

Ramène tous le(s) objet(s) jQuery correspondants aux éléments 'P' qui ont la classe 'titre'



la fonction \$

Attention

La fonction jQuery **\$** ne ramène pas un objet du D.O.M mais un objet jQuery !

```
<head><script type="text/javascript" src="jquery-1.3.2.js" /></script></head><body>
```

Attention : la fonction jQuery **\$** ramène un **objet jQuery** et
et non un noeud du D.O.M

```
<p>
```

```
<div id="monDiv" attributbidon="toto">Ici le div Coucou</div>
```

```
</p>
```


Grace à la méthode **attr** de l'objet jQuery (et non du D.O.M !)

On recherche les attributs de la div id="monDiv1" qui possède un attributbidon="toto"

```
<br>
```

```
<a href="#" onClick="alert($('#monDiv').attr('attributbidon')) ;return false;"> Cliquer ici  
pour tester </a>
```

```
</body>
```

[Démo](#)

Utilisation de la fonction **each**



Fonction qui boucle sur chaque élément trouvé, et exécute une fonction a chaque passage.
L'argument est un entier représentant la position de l'élément en cours.
Un "return false" a l'intérieur de la fonction définie stoppera l'exécution de la boucle.
En revanche "return true" forcera le passage a l'itération suivante
(de la même manière que l'instruction "continue" dans une boucle normale).
this représente l'élément jQuery trouvé.

```
$("#img").each( function(i)
{
  this.src = "test" + i + ".jpg";
} )
```

[Démon](#)

Quelques fonctions en vrac...



html() Récupère le contenu du premier élément trouvé avec les balises.

Exemple :

sur le source `<div><input/></div>`

`$("#div").html();` renverra la chaîne `<input/>`

`Html(arg)` Remplace le contenu HTML par l'argument

text() idem que `html()` mais sans les balises (uniquement le contenu texte).

sur le source `<p>ancien texte</p>`

`$("#p").text("Nouveau texte");`

Impactera au niveau du source

`<p>Nouveau texte</p>`

Démo sur text() et html()

<body>

attention à ce que retourne jQuery !

```
<div id="monDiv1">Coucou c'est moi 1 </div>
<div id="monDiv2">Coucou c'est moi 2 <div> &nbsp; ici un autre
texte</div></div>
<div id="monDiv3">Coucou c'est moi 3</div>
```


La méthode jQuery `.html()` récupère le contenu HTML du noeud et de tous les noeud enfants

Pour récupérer le contenu texte du noeud et de tous les noeud enfants utiliser la méthode `.text()`

```
<script type='text/javascript'>
alert( $('#monDiv2').html() + "\n" + $('#monDiv2').text() );
// Attention ; $ de jQuery ne ramène pas vraiment un élément DOM !!
var x= $('#monDiv2');
var y = document.getElementById("monDiv2"); // Utilisation du DOM
alert("x[0].firstChild.nodeValue\n" + x[0].firstChild.nodeValue);
alert("y.firstChild.nodeValue\n" + y.firstChild.nodeValue);
</script>
</body>
```

Démo

Quelques fonctions en vrac...



removeAttr(arg) Supprime un attribut

Exemple :

Sur le source `<input disabled="disabled"/>`
`$("input").removeAttr("disabled")`
Impactera au niveau du source
`<input/>`

val() Récupère le contenu de l'attribut "value"

Attention ; dans certains cas peuvent retourner un tableau de valeurs
(cas du select multiple)

Exemple :

Sur le source `<input type="text" value="ceci est du texte"/>`
`$("input").val();`
Renvoiera la chaîne `ceci est du texte`
--> Avec un argument, permet de remplacer la valeur

hasClass(arg) Teste si l'élément possède la classe recherchée

...

documentation officielle

Il existe beaucoup de fonctions !

Se référer à la doc officielle

exemple

Un petit mot sur les événements

```
<script type="text/javascript" src="jquery-1.3.2.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("a").click(function(){
        alert("Vous venez de cliquer sur "+ $(this).html());
        return false;
    });
});
</script>

<body>
    <div id="div1">
        <p>Entete de ma page avec un <a href="#">lien 1</a></p>
    </div>
    <div id="div2">
        <p>Contenu de ma page avec un <a href="#">lien 2</a></p>
    </div>
    <div id="div3">
        <p>Pied de page avec un <a href="#">lien 3</a></p>
    </div>
</body>
```

Démo

les événements (suite)

- Les événements
- jQuery.bind()
- jQuery.blur()
- jQuery.change()
- jQuery.click()
- jQuery.dblclick()
- jQuery.die()
- jQuery.error()
- jQuery.focus()
- jQuery.hover()
- jQuery.keydown()
- jQuery.keypress()
- jQuery.keyup()
- jQuery.live()
- jQuery.load()
- jQuery.mousedown()
- jQuery.mousemove()
- jQuery.mouseout()
- jQuery.mouseover()
- jQuery.mouseup()
- jQuery.one()
- jQuery.ready()
- jQuery.resize()
- jQuery.scroll()
- jQuery.select()
- jQuery.submit()
- jQuery.toggle()
- jQuery.trigger()
- jQuery.unbind()
- jQuery.unload()

jQuery.mouseover()

Description :

Associe une fonction à l'évènement mouseover. Cet évènement se déclenche lorsque l'utilisateur positionne le curseur de la souris au dessus d'un élément.

Liste des paramètres acceptés :

mouseover(fonction)

mouseover(fonction)

Paramètres:

fonction (Fonction): fonction à associer à l'évènement.

Cette methode renvoie: objet jQuery

Prenons l'exemple suivant:

```
$(p).mouseover(function) { alert("Hello"); };
```

Testons sur le script suivant:

```
<p>Hello</p>
```

On obtiendra :

```
<p onmouseover="alert('Hello');">Hello</p>
```

Source : <http://jquery.com/web.fr/>

Voir aussi <http://www.learningjquery.com/2008/03/working-with-events-part-1>

Exemple de synthèse sur les possibilités offertes

synthèse sur les possibilités

```
<head>
```

```
<script type="text/javascript" src="jquery.js"></script> — inclusion du fichier jQuery
```

```
<script type="text/javascript">
```

```
$(document).ready(function(){ — Chargement de la fonction quand le D.O.M sera en mémoire
```

```
    $(".button").click(function(){
        $("#panel").slideDown("slow");
    });
```

```
});
```

```
</script>
```

```
</head>
```

A quoi rattacher l'évènement ?
Une classe ou un ID CSS ... Une balise HTML ...

Evenement qui provoquera l'appel de la fonction

Fonction jQuery qui animera l'élément dont l'ID est 'panel'

jQuery ramène le noeud du D.O.M dont l'ID est 'panel'

Les effets

Les effets

- ▶ `animate()`
- ▶ `fadeIn()`
- ▶ `fadeOut()`
- ▶ `fadeTo()`
- ▶ `hide()`
- ▶ `show()`
- ▶ `slideDown()`
- ▶ `slideToggle()`
- ▶ `slideUp()`
- ▶ `toggle()`

Voir <http://jquery.developpeur-web2.com/documentation/effets.php>

Ajax

Demo locale en utilisant un tutorial de
http://www.snoupix.com/initiation-a-ajax-avec-jquery-partie-1_tutorial_20.html

Lancement de wampserver
Paramétrage de Mysql via phpMyAdmin
Paramétrage du script

Voir répertoire « Ajax » pour dézipper les sources

webographie

<http://www.learningjquery.com/2012/10/What-Is-jQuery/>
<http://docs.jquery.com/Tutorials>
<http://www.webdesignerwall.com/tutorials/jquery-tutorials-for-designers/>
<http://www.babylon-design.com/site/index.php/2008/01/21/211-apprendre-et-comprendre>
http://speckyboy.com/2009/03/11/25-powerful-and-useful-jquery-tutorials-for-developers
<http://www.noupe.com/tutorial/51-best-of-jquery-tutorials-and-examples.html>
<http://jquery.developpeur-web2.com/>
<http://www.clochix.net/post/2009/02/09/Presentation-de-jQuery>
<http://www.extratuts.com/25-jquery-tutorials-roundup-with-examples>
<http://www.noupe.com/tutorial/51-best-of-jquery-tutorials-and-examples.html>

Elements visuels pour interface

<http://jqueryui.com/demos/>

Pas jQuery mais très intéressant

<http://www.sencha.com/products/extjs/examples/>

Question?



TP (3 heures +)

- 1) Grafikart - Tutoriel jQuery UI Datepicker à visionner
- 2) Lire et jouer les exemples du didacticiel de w3schools
<http://www.w3schools.com/jquery/default.asp>
- 3) Si le temps butiner via veille-techno.txt