

# COMPLÉMENTS SUR JAVASCRIPT

## AJAX, BIBLIOTHÈQUE JQUERY

\*

### Objectifs

- Connaître les principes d'Ajax et un exemple de bibliothèque de développement (cas de jQuery)
- Pour un autodidacte, contrôler et mettre à niveau ses connaissances

### Compléments

- Un éditeur adapté aux pages Web (HTML-Kit, NotePad++, etc.)

### Niveau de compétences en pré-requis

- Connaissance des bases de la réalisation de pages Web et de la programmation avec Javascript (cf. documents de l'auteur \*)

\*

*Ce document d'enseignement est diffusé pour un usage individuel.  
Il est librement téléchargeable sur le site de l'auteur \*.*

Michel Cartereau - Décembre 2009

# AJAX

Présentation de Ajax

Spécifications de XMLHttpRequest

Exemple en Javascript

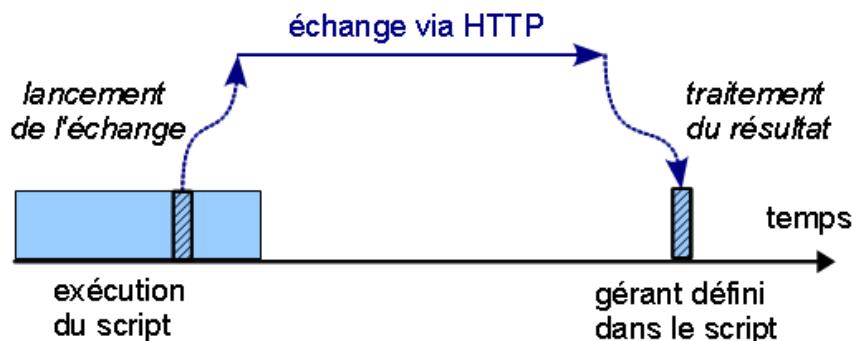
# PRÉSENTATION DE AJAX

## ASYNCHRONOUS JAVASCRIPT AND XML<sup>1</sup>

### • ORIGINES

- FONCTION XMLHTTPREQUEST CRÉÉE PAR MICROSOFT EN 1999<sup>2</sup>  
récupération par un script, d'un texte distant via le protocole HTTP, au format XML ou non, réalisée en mode asynchrone
- AVANTAGES POUR UNE APPLICATION DANS UNE PAGE WEB  
récupération d'information distante sans (re)chargement de page, économie par réduction du volume des échanges avec le serveur
- TECHNIQUES RÉUNIES DANS L'APPELLATION AJAX  
scripts en Javascript avec échanges en XML via XMLHttpRequest, application notamment dans les applications Internet (RIA<sup>3</sup>)
- GÉNÉRALISATION DANS LES NAVIGATEURS RÉCENTS  
incompatibilités existantes et normalisation en cours par le W3C

### • PRINCIPE



- EXÉCUTION EN MODE ASYNCHRONE  
échange via HTTP en parallèle du déroulement du script, avec définition d'un gérant, une fonction appelée lors de la fin de l'échange
- CONTRAINTES DE SÉCURITÉ  
accès limité au site d'origine<sup>4</sup> de la page (ou au dossier si en local)

<sup>1</sup> Terme forgé en 2005 par Jesse James Garrett ; voir l'article publié sur le site de sa société en : <http://www.adaptivepath.com/ideas/essays/archives/000385.php>

<sup>2</sup> Apparition pour la première fois dans la version 5 du navigateur Internet explorer.

<sup>3</sup> RIA : *rich Internet application*, application communiquant via l'Internet et s'exécutant dans un navigateur ou dans un environnement local sûr ; cas notamment des outils de Google : recherche d'information, messagerie (*Gmail*) et cartographie (*Google maps*)

<sup>4</sup> Politique de sécurité (*same origin policy*) mise en place afin d'éviter des mauvaises utilisations.

# SPÉCIFICATIONS DE XMLHTTPREQUEST

---

## PRÉSENTATION ICI DES SPÉCIFICATIONS MINIMALES (NORMALISATION EN COURS PAR LE W3C <sup>1</sup>)

### • UN OBJET POUR L'ÉCHANGE

#### ▪ CLASSE XMLHttpRequest

cas particulier de Internet explorer : `ActiveXObject (paramètre)`  
avec en *paramètre* : "`Msxml2.XMLHTTP`" ou bien "`Microsoft.XMLHTTP`"

#### ▪ PROPRIÉTÉS

`onreadystatechange` ..... gérant associé, une fonction appelée lors de chaque étape de l'avancement de l'échange

`readyState` ..... état d'avancement de l'échange, valeurs :  
0 (non initialisé), 1 (prêt), 2 (envoyé),  
3 (en cours), 4 (achevé)

`status` ..... code du résultat de l'échange en HTTP <sup>2</sup>, bon si valeur comprise entre 200 et 299  
(une valeur nulle en cas d'accès local)

`statusText` ..... libellé du résultat de l'échange en HTTP

`responseText` ..... texte <sup>3</sup> du contenu reçu via HTTP

`responseXML` ..... élément (DOM) du contenu reçu via HTTP

#### ▪ MÉTHODES (PRINCIPALES)

`open (méthode, adresse)` ..... préparation de l'échange avec indication du mode d'envoi ("GET" ou "POST") et de l'adresse du contenu demandé (URL)

`send (données)` ..... lancement de l'échange avec indication des éventuelles données <sup>4</sup> à envoyer dans le cas de la méthode "POST" (null sinon)

`abort ()` ..... abandon de l'échange

*MANIPULATIONS POSSIBLES AUSSI VIA DES BIBLIOTHÈQUES (CF. JQUERY)*

---

<sup>1</sup> Voir en <http://www.w3.org/TR/XMLHttpRequest/>

<sup>2</sup> Codes normalisés par le W3C : <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

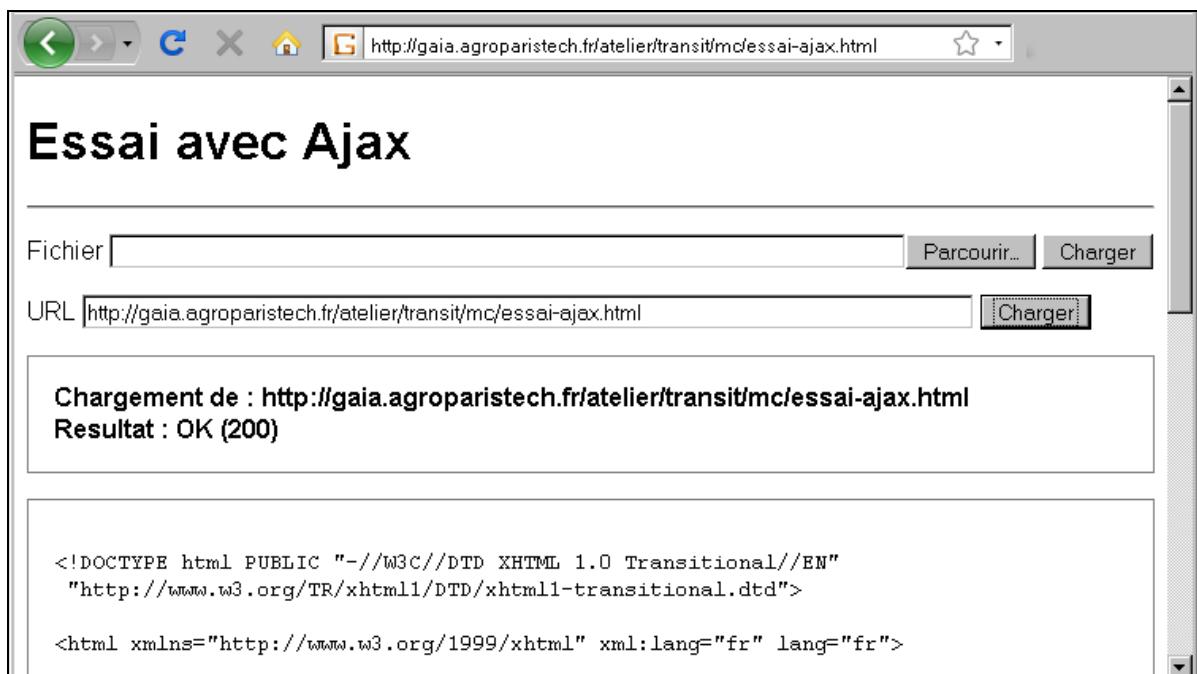
<sup>3</sup> Le texte reçu est a priori codé en Unicode (UTF8) et non pas en ISO-8859-1.

<sup>4</sup> Les données doivent être a priori encodées en Unicode (UTF8).

# EXEMPLE D'UTILISATION DE AJAX

## FORMULAIRE DE CHARGEMENT DANS LA PAGE VIA AJAX

- CAS D'UN FICHIER DISTANT (SUR LE MÊME SITE)



**Essai avec Ajax**

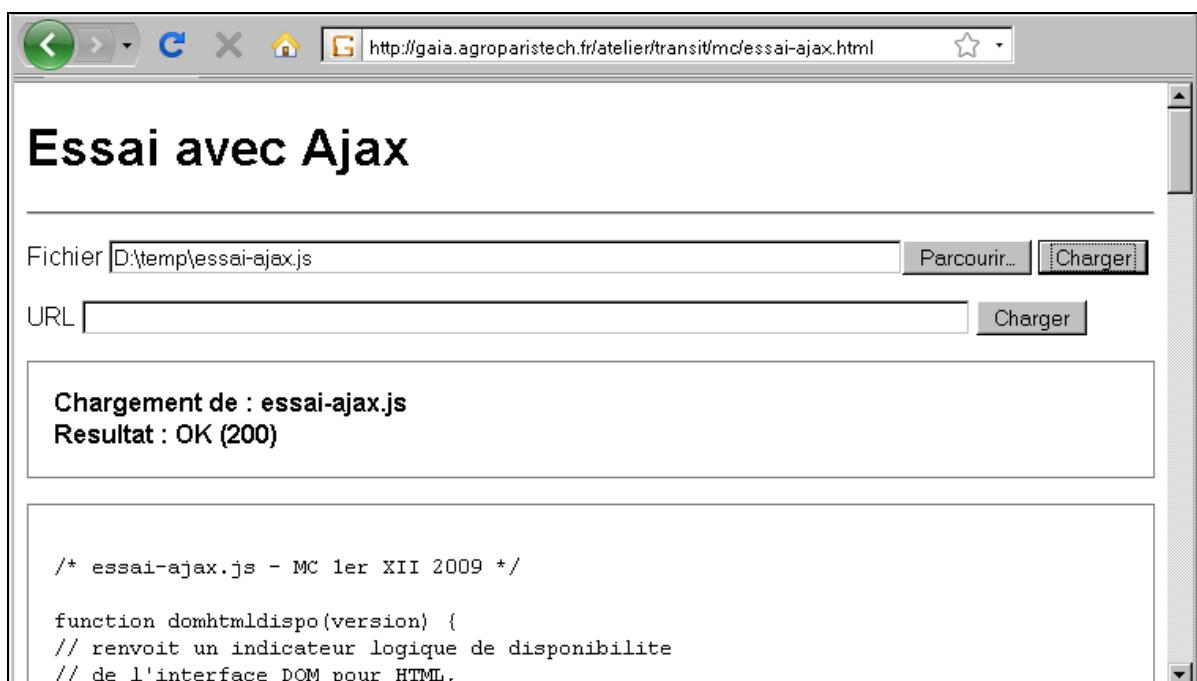
Fichier  Parcourir... Charger

URL  Charger

Chargement de : http://gaia.agroparistech.fr/atelier/transit/mc/essai-ajax.html  
Resultat : OK (200)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
```

- CAS D'UN FICHIER SITUÉ SUR LE POSTE



**Essai avec Ajax**

Fichier  Parcourir... Charger

URL  Charger

Chargement de : essai-ajax.js  
Resultat : OK (200)

```
/* essai-ajax.js - MC 1er XII 2009 */  
  
function domhtmldispo(version) {  
    // renvoie un indicateur logique de disponibilite  
    // de l'interface DOM pour HTML,
```

## EXEMPLE D'UTILISATION DE AJAX : CODE XHTML

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
 xml:lang="fr" lang="fr">

<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1" />
  <title>Essai avec Ajax</title>
  <style type="text/css"><!--
    #contenu, #traces
    { border: gray solid thin; padding: 1em; margin: 1em 0; }
    #traces p { font-weight: bold; margin: 0; }
  --></style>
</head>

<body onload="document.forms['f'].elements['zurl'].value =
  window.location;">
<h1>Essai avec Ajax</h1><hr />
<form action="" onsubmit="return false;" id="f" >
  <p>Fichier
    <input type="file" name="zfichier" id="zfichier" size="80" />
    <input type="button" value="Charger"
      id="boutonfichier" onclick=
      "lancerRequeteAjax(this.form.elements['zfichier'].value)" />
  </p>
  <p>URL
    <input type="text" name="zurl" id="zurl" size="90" value="" />
    <input type="button" value="Charger" id="boutonurl" onclick=
      "lancerRequeteAjax(this.form.elements['zurl'].value)" /></p>
</form>
<div id="traces">
  <noscript>Désolé ! pas de javascript disponible.</noscript>
</div>
<div id="contenu"></div>
<hr />
<address>MC 2 XII 2009</address>
<script type="text/javascript" src="essai-ajax.js"></script>
</body>

</html>
```

## EXEMPLE D'UTILISATION DE AJAX : SCRIPT

---

```
/* essai-ajax.js - MC 2 XII 2009 */

function lancerRequeteAjax(adresse) {
// chargement via Ajax du contenu en adresse,
// traces d'avancement dans la division 'resultat'
// et si recuperé, contenu mis dans la division 'contenu'
var xhr;

// contrôle minimal sur l'adresse
if (! adresse) {
    return;
}

// vérification de la compatibilité avec DOM
if (! domhtmldispo('1.0') ) {
    window.alert('Anomalie : DOM inutilisable !');
    return;
}

// nettoyages
supprimerContenu('traces');
supprimerContenu('contenu');

// création de l'objet associé à la requête
xhr = false; // a priori
if ('function' == (typeof window.XMLHttpRequest) ) {
    xhr = new XMLHttpRequest();
} else if ('function' == (typeof window.ActiveXObject) ) {
    // cas de Internet explorer
    try {
        xhr = new ActiveXObject("Msxml2.XMLHTTP");
    } catch(erreur) {
        try {
            xhr = new ActiveXObject("Microsoft.XMLHTTP");
        } catch(erreur) {
            xhr = false;
        }
    }
}
if (! xhr) {
    ajoutContenu('P', "Anomalie : pas d'Ajax !", 'traces');
    return;
}
```

## EXEMPLE D'UTILISATION DE AJAX : SCRIPT (SUITE)

---

```
// creation du gerant associe a la requete
xhr.onreadystatechange = function () {
    if (xhr.readyState == 4) {
        ajoutContenu('P', "Résultat : " +
            xhr.statusText + " (" + xhr.status + ")", 'traces');
        if ( (xhr.status == 0) ||
            (xhr.status >= 200 && xhr.status < 300) ) {
            ajoutContenu(
                'PRE', recoderBalisesHtml(xhr.responseText), 'contenu' );
        }
    }
} // function

// lancement de la requete
ajoutContenu('P', 'Chargement de : ' + adresse, 'traces');
xhr.open('GET', adresse);
try {
    xhr.send(null);
} catch (erreur) {
    ajoutContenu('P', 'Anomalie : ' + erreur.toString(), 'traces');
}
}

function recoderBalisesHtml(extrait) {
// renvoie l'extrait de code HTML
// avec codage de "<", ">" en "<" et ">"
    return extract.replace(/</g, "<").replace(/>/g, ">");
}

function domhtmldispo(version) {
// renvoie un indicateur logique de disponibilite
// de l'interface DOM pour HTML,
// dans la version indiquee ('1.0' ou '2.0')
// exemple d'appel : domhtmldispo('1.0')
    var resultat = false;
// action
    if (document.implementation &&
        document.implementation.hasFeature) {
        resultat = document.implementation.hasFeature('HTML', version);
    }
    return resultat;
} // function
```

## EXEMPLE D'UTILISATION DE AJAX : SCRIPT (FIN)

---

```
function supprimerContenu(idParent) {
// suppression du contenu du noeud idParent dans DOM
var parent, rang, fils;

parent = document.getElementById(idParent);
if (null == parent) {
    return false;
}
if (undefined != (typeof parent.innerHTML) ) {
    // cas notamment de Internet explorer
    parent.innerHTML = '';
} else {
    for (rang = 0; rang < parent.childNodes.length; ++rang) {
        parent.removeChild(parent.childNodes[rang]);
    } // for
}
}

function ajoutContenu(balise, texte, idParent) {
// ajoute un element selon balise avec texte,
// a l'element identifie par idParent dans DOM
var conteneur, noeudTexte, noeudBalise;

conteneur = document.getElementById(idParent);
if (null == conteneur) {
    return false;
}
if (undefined != (typeof conteneur.innerHTML) ) {
    // cas notamment de Internet explorer
    conteneur.innerHTML += '<' + balise + '>' +
        texte + '</' + balise + '>';
} else {
    noeudBalise = document.createElement('balise');
    noeudTexte = document.createTextNode(texte);
    noeudBalise.appendChild(noeudTexte);
    conteneur.appendChild(noeudBalise);
}
}
```

# JQUERY

## Présentation de jQuery

### Exemples

### Principales caractéristiques

# PRÉSENTATION DE JQUERY

---

UNE BIBLIOTHÈQUE POUR JAVASCRIPT



## • ORIGINES

- CRÉATION EN 2006 PAR JOHN RESIG
  - objectif : simplifier le développement de pages Web avec Javascript
- BIBLIOTHÈQUE EN CODE LIBRE <sup>1</sup>
  - développement et documentation assurés par une équipe

## • PRINCIPALES CARACTÉRISTIQUES

- ECRITURE EN JAVASCRIPT DE MANIÈRE NON INTRUSIVE <sup>2</sup>
  - séparation nette des scripts d'avec le code XHTML
- ORIENTATION VERS LA MANIPULATION D'ÉLÉMENTS D'UNE PAGE WEB
  - extension du mécanisme de sélection existant dans CSS
- SIMPLIFICATION DE L'UTILISATION DE DOM
  - mécanismes puissants avec une notation minimalist
- INTÉGRATION DE TECHNIQUES POUR DES APPLICATIONS INTERNET (RIA <sup>3</sup>)
  - effets d'animation, interfaçage avec Ajax, etc.
- BIBLIOTHÈQUE EXTENSIBLE
  - nombreuses extensions disponibles (formulaires, menus, etc.)
- GESTION POUSSÉE DE LA COMPATIBILITÉ AVEC LES NAVIGATEURS
  - prise en charge de particularités hors norme de navigateurs
  - mais fonctionnement avec des versions relativement récentes : Firefox ≥ 2, Internet explorer ≥ 6, Safari ≥ 3, Opera ≥ 9, Chrome

SITE DE RÉFÉRENCE : <http://jquery.com/>

---

<sup>1</sup> Autres bibliothèques existantes : Dojo (<http://www.dojotoolkit.org/>), MooTools (<http://mootools.net/>), Prototype (<http://www.prototypejs.org/>), ExtJS (<http://www.extjs.com/>), Yahoo user interface library (<http://developer.yahoo.com/yui/>)

<sup>2</sup> Mode non intrusif (*unobtrusive*) : instructions en javascript en dehors du contenu en XHTML (séparation de la structure et du comportement).

<sup>3</sup> RIA : *rich Internet application*, application communiquant via l'Internet et s'exécutant dans un navigateur ou dans un environnement local sûr.

# EXEMPLE DU COLORIAGE (VERSION N° 1)

## COLORIAGE D'UN PARAGRAPHE

### Exemple avec jQuery : coloriage (version n° 1)

Ce texte doit s'afficher en couleur !

MC - 25 XI 2009

#### • CARACTÉRISTIQUES

- DOUBLE SCRIPT

Bibliothèque jQuery + script particulier à la page : application d'un style de couleur à la division identifiée par `contenu`

- IDENTIFICATION D'UNE SÉLECTION D'ÉLÉMENT(S) DE LA PAGE

notation : `jQuery(sélection)` ou `$(sélection)` avec `sélection`<sup>1</sup> exprimée a priori comme en CSS, dans une chaîne de caractères

exemple : `jQuery("div#contenu")` ou `$( "div#contenu" )`

- MANIPULATION DE LA SÉLECTION PAR APPLICATION D'UNE MÉTHODE

définition d'un style via `.css(propriété, valeur)`

exemple : `$( "div#contenu" ).css("color", "red") ;`

#### • CODE EN XHTML DU CONTENU DE LA PAGE

```
<body>
<h1>Exemple avec jQuery : coloriage (version n° 1)</h1>
<hr />
<div id="contenu">Ce texte doit s'afficher en couleur !</div>
<hr />
<address>MC - 25 XI 2009</address>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jq-colorier-v1.js"></script>
</body>
```

#### • CODE EN JAVASCRIPT DANS `jq-colorier-v1.js`

```
$( "div#contenu" ).css("color", "red");
```

<sup>1</sup> Le résultat de la fonction `jQuery()` ou `$( )` est en fait un objet (*wrapper set*) contenant la représentation dans DOM du ou de chacun des éléments sélectionnés sous la forme d'un tableau.

# EXEMPLE DU COLORIAGE (VERSION N° 2)

---

## RENFORCEMENT DE LA ROBUSTESSE DU SCRIPT

### • AMÉLIORATION

- EXISTENCE D'UN CAS DE MAUVAISE EXÉCUTION  
script inopérant si chargement avant le contenu de la page
- GARANTIR LA BONNE EXÉCUTION DANS TOUS LES CAS  
exécution du code seulement après le chargement de la page
- MÉCANISME PARTICULIER DANS JQUERY  
indication d'une fonction anonyme <sup>1</sup> exécutée après le chargement :

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

ou en raccourci : \$( function () { code } );

### • CODE EN XHTML DE L'ENTÊTE DE LA PAGE

déplacement des balises de chargement de script dans l'entête <sup>2</sup> afin de vérifier le bon fonctionnement dans ce cas particulier :

```
<head>
<meta http-equiv="Content-Type"
      content="text/html; charset=iso-8859-1" />
<title>Exemple avec jQuery : coloriage (version n° 2)</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jq-colorier-v2.js"></script>
</head>
```

### • CODE EN JAVASCRIPT DANS jq-colorier-v2.js

```
$( function () {
    $("div#contenu").css("color", "red");
});
```

---

<sup>1</sup> Cette particularité de Javascript (*function literal*) est liée à la représentation d'une fonction comme un objet (avec le paramétrage et le code mais sans le nom) ; une déclaration ordinaire affecte en fait l'objet créé à une variable portant le nom de la fonction : `function f() { code }` est équivalent à `f = function () { code }`

<sup>2</sup> De manière générale, il est conseillé de placer les balises de chargement de scripts plutôt à la fin du corps de la page en XHTML afin de ne pas retarder l'affichage du contenu de la page.

# EXEMPLE DU COLORIAGE (VERSION N° 3)

COLORIAGE AVEC UNE COULEUR SAISIE DANS UN FORMULAIRE

## Exemple avec jQuery : coloriage (version n° 3)

Ce texte doit s'afficher en couleur !

Code de couleur :

MC - 25 XI 2009

### • CARACTÉRISTIQUES

- IDENTIFICATION DE LA ZONE DE SAISIE ET DU BOUTON D'ACTIVATION  
attribut `id` dans les balises `input` en XHTML  
récupération de la valeur en jQuery : `$(sélection).val()`
- INDICATION DU TRAITEMENT ASSOCIÉ AU CLIC SUR LE BOUTON  
méthode en jQuery portant le nom de l'événement : `.click(fonction)`  
fonction de traitement <sup>1</sup> de l'événement avec un paramètre : `event` <sup>2</sup>

### • CODE EN XHTML DU FORMULAIRE

```
<form action="">
  Code de couleur : <input type="text" name="zcouleur"
    id="zcouleur" value="#FF0000" size="7" />
  <input type="button" value="Colorier" id="bouton" />
</form>
```

### • CODE EN JAVASCRIPT DANS jq-colorier-v3.js

```
$( function() {
  $('input#bouton').click( function (event) {
    // application, apres le clic sur le bouton, de la couleur
    // definie dans la zone identifiee par #zcouleur,
    // au texte de la division identifiee par #contenu
    $("div#contenu").css("color", $("input#zcouleur").val() );
  } );
} );
```

<sup>1</sup> Cette fonction de traitement s'appelle le gestionnaire d'événement (*event handler* ou *listener*)

<sup>2</sup> Ce paramètre décrit l'événement produit ; il est normalisé (DOM) mais Internet explorer ne respecte pas cette norme ; jQuery garantit cependant un jeu de propriétés pour un événement.

# EXEMPLE DU MARQUAGE (VERSION N° 1)

## MARQUAGE D'UN TYPE D'ÉLÉMENT DE LA PAGE AU CHOIX

### Marquage via jQuery (version n°1)

---

Marquage des occurrences d'un élément de la page.

---

Elément HTML à marquer :

---

MC 26 XI 2009

- **CARACTÉRISTIQUES**

- AJOUT À UN ÉLÉMENT D'UNE CLASSE DE STYLE

- méthode en jQuery : `.addClass (nom de la classe)`

- **CODE EN XHTML (EXTRAITS)**

```
<style type="text/css"><!--
.intro { font-weight: bold; }
.marque { border: thin dotted #ff0000; }
--></style>

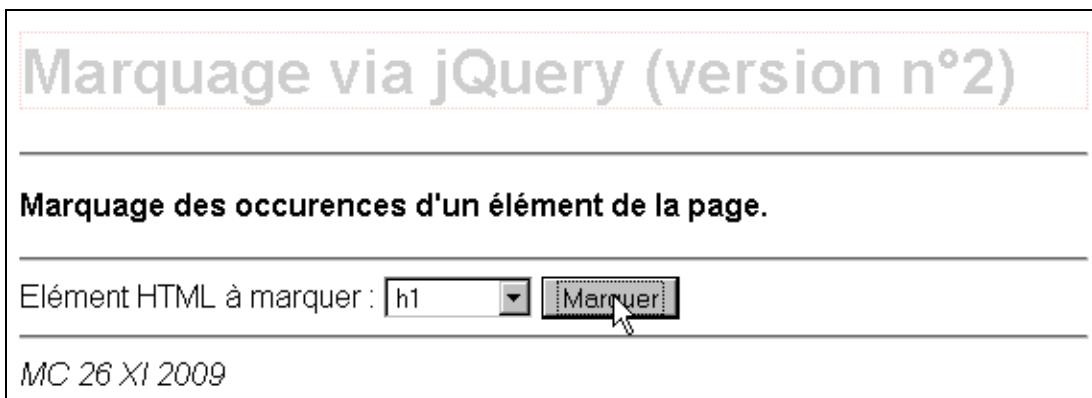
<h1>Marquage via jQuery (version n°1)</h1> <hr />
<p class="intro">Marquage des occurrences d'un élément
de la page.</p> <hr />
<form action="">
  <label>Elément HTML à marquer :
    <select name="zelem" id="zelem">
      <option>h1</option> <option>p</option>
      <option>label</option> <option>address</option>
    </select></label>
  <input type="button" value="Marquer" id="bouton" />
</form>
```

- **CODE EN JAVASCRIPT DANS jq-marquer-v1.js**

```
$( function () {
  $("input#bouton").click ( function (event) {
    // marquage des éléments, choisis via le bouton identifié
    // par "zelem", avec la classe de style "marque"
    $( $("select#zelem").val() ).addClass("marque") ;
  } );
} );
```

## EXEMPLE DU MARQUAGE (VERSION N° 2)

MARQUAGE AVEC EFFET D'ANIMATION DE L'ÉLÉMENT  
(EFFACEMENT PUIS RÉAPPARITION PROGRESSIFS)



Marquage via jQuery (version n°2)

Marquage des occurrences d'un élément de la page.

Elément HTML à marquer :

MC 26 XI 2009

### • CARACTÉRISTIQUES

#### ▪ MÉTHODES POUR UN EFFET D'ANIMATION EN JQUERY

effacement progressif : `.fadeOut(vitesse)`

tracé progressif : `.fadeIn(vitesse)`

avec la vitesse exprimée en nombre de millisecondes ou  
en utilisant les valeurs prédéfinies : "slow" "normal" "fast"

#### ▪ ENCHAÎNEMENT D'ACTIONS EN JQUERY

succession d'actions : `$(sélection).méthode1.méthode2. ... .méthodeN` ;

chaque méthode s'applique à son tour sur les éléments de la  
sélection <sup>1</sup>

### • CODE EN JAVASCRIPT DANS `jq-marquer-v2.js`

```
$( function () {  
    $("input#bouton").click ( function (event) {  
        // marquage des éléments, choisis via le bouton identifié  
        // par "zelem", avec la classe de style "marque" et  
        // une animation (effacement puis reaparition progressifs)  
        $( $("select#zelem").val() )  
            .addClass("marque")  
            .fadeOut("slow")  
            .fadeIn("slow");  
    } );  
} );
```

<sup>1</sup> Par convention dans jQuery, une méthode renvoie a priori la sélection qu'elle a reçue afin de permettre ces actions en chaîne.

# EXEMPLE DU MARQUAGE (VERSION N° 3)

## MARQUAGE AVEC EFFET D'ANIMATION DE L'ÉLÉMENT PUIS ANNULATION DU MARQUAGE À LA FIN

### • CARACTÉRISTIQUES

#### ▪ RETRAIT À UN ÉLÉMENT D'UNE CLASSE DE STYLE

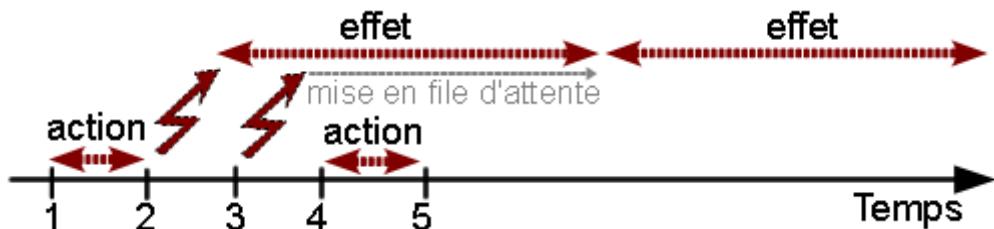
méthode en jQuery : `.removeClass(nom de la classe)`

#### ▪ SOUCI DE DÉROULEMENT TEMPOREL

marquage invisible si simple succession d'actions ! (en fait trop bref)

```
$ ( $("select#zelem").val() )  
  .addClass("marque").fadeOut("slow")  
  .fadeIn("slow").removeClass("marque");
```

les actions sont bien lancées dans l'ordre d'écriture mais les effets d'animation se déroulent en parallèle (asynchronisme) et à la suite



#### ▪ PARAMÈTRAGE D'UNE ACTION EXÉCUTÉE À LA FIN DE L'ANIMATION

fonction indiquée en second paramètre : `.effet(vitesse, fonction)`  
qui sera alors exécutée à la fin de l'effet d'animation (*callback*)

### • CODE EN JAVASCRIPT DANS `jq-marquer-v3.js`

```
$( function () {  
  $("input#bouton").click ( function (event) {  
    // marquage des éléments, choisis via le bouton identifié  
    // par "zelem", avec la classe de style "marque"  
    // mais seulement pendant l'animation  
    $( $("select#zelem").val() )  
      .addClass("marque")  
      .fadeOut("slow")  
      .fadeIn("slow", function () {  
        $( $("select#zelem").val() ).removeClass("marque");  
      } );  
  } );  
});
```

# EXEMPLE DU MARQUAGE (VERSION N° 4)

MARQUAGE À PARTIR D'UNE SÉLECTION D'ÉLÉMENT(S) SAISIE, ET AVEC CHARGEMENT D'UNE PAGE AU CHOIX DANS UN CADRE

## Marquage via jQuery (version n°4)



Page à charger : D:\transithtml-dynalessais\jquery\formulaire.html Parcourir... Charger

Sélection à marquer : textarea Marquer

Eléments en résultat

TEXTAREA#zcommentaire

Exemple de formulaire

Nom :

Commentaire :

### • CARACTÉRISTIQUES

- PAGE INSÉRÉE DANS UN CADRE INTERNE (`<iframe>`)
  - chargement par modification de l'attribut `src` de `<iframe>` avec l'événement `load` déclenché à la fin de chargement
  - arborescence de la page chargée, en fils de l'élément `<iframe>` accessible via la méthode `contents()`
  - recherche d'un élément d'une arborescence via la méthode `find()`
  - ajout manuel du style de marquage dans l'entête de la page chargée
- MANIPULATIONS DES ÉLÉMENTS EN RÉSULTAT DE LA SÉLECTION
  - parcours de l'ensemble des éléments par via la méthode `each()` avec en paramètre une fonction s'appliquant à chacun des membres de l'ensemble identifié par son rang (compté à partir de zéro et donné en paramètre de cette fonction) et accessible via l'objet `this`

```
each( function (rang) { actions } )
```
  - récupération des balise et identificateur d'un élément via DOM : `élément.nodeName` et `élément.getAttribute('id')`

## EXEMPLE DU MARQUAGE (VERSION N° 4) : HTML

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">

<head>
<meta http-equiv="Content-Type"
      content="text/html; charset=iso-8859-1" />
<title>Marquage via jQuery (version n°4)</title>
<style type="text/css"><!--
.intro { font-weight: bold; }
.marque { border: thin dotted red; }
div#formulaire, div#contenu { margin: 1em; padding: 1em;
    background: #ffffcc; border: thin solid gray; }
div#resultats {background: white; padding: 0.5em;
    border: thin solid gray; }
iframe#cadre {background: white; }
--></style>
</head>

<body>
<h1>Marquage via jQuery (version n°4)</h1>
<hr />

<div id="formulaire" class="intro">
<form action="">
<p><label>Page à charger :
<input type="file" name="zurl" id="zurl" size="70" /></label>
<input type="button" value="Charger" id="chargement" /></p>
<p><label>Sélection à marquer :
<input type="text" name="zelem" id="zelem" size="30" /></label>
<input type="button" value="Marquer" id="marquage" /></p>
</form>
<p>Eléments en résultat</p>
<div id="resultats"></div>
</div>

<div id="contenu">
<iframe src="" width="100%" height="400px" id="cadre"></iframe>
</div>
<noscript>
<strong>Désolé, impossible d'utiliser Javascript !</strong>
</noscript>
<hr />
<address>MC 29 XI 2009</address>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="jq-marquer-v4.js"></script>
</body>

</html>
```

## EXEMPLE DU MARQUAGE (VERSION N° 4) : SCRIPT

---

```
/* jq-marquer-v4.js - MC 29 XI 2009 */

$( function () {

    $("input#chargement").click ( function (event) {
        // chargement du fichier choisi via la zone de saisie "#url",
        // dans le cadre interne "#cadre"
        var cadre = $('iframe#cadre');
        // preparation du traitement post chargement
        cadre.load( function () {
            // ajout de style dans l'entete de la page chargee
            cadre.contents().find('head')
                .append('<style type="text/css">' +
                    '.marque {border: thin dotted red;}</style>');
        } ); // load
        // chargement
        cadre.attr('src', $('input#zurl').val() );
    } ); // input#chargement

    $("input#marquage").click ( function (event) {
        // marquage des elements choisis dans le bouton identifie
        // par "zelem", avec la classe de style "marque"
        // mais seulement pendant l'animation
        var contenu = $('iframe#cadre').contents();

        // nettoyage de la zone des resultats
        $('div#resultats').html('');
        // annulation des eventuels marquages precedents
        contenu.find('*').removeClass("marque");
        // recherche et traitement des elements
        contenu.find( $('input#zelem').val() )
            .addClass("marque")
            .each( function (rang) {
                // affichage dans la division "#resultats" du nom
                // de la balise et si existant son identificateur,
                // pour chaque element trouve
                $('div#resultats').append(this.nodeName +
                    (this.getAttribute('id') ?
                        '#' + this.getAttribute('id') : '') +
                    ' ');
                return true;
            } ); // each
    } ); // input#marquage

} );
```

# EXEMPLE DE LA LISTE AVEC JQUERY

## LISTE DÉROULANTE PAR SOUS-LISTES

### Liste déroulante avec jQuery

- Bactéries
- Algues
- Champignons et lichens
- **Plantes**
  - **Spermatophyta**
    - Gymnospermes
    - **Angiospermes** 
  - Bryophyta
  - Pteridophyta
- Animaux
- Autres ou inclassables

MC 2 XII 2009

## • CARACTÉRISTIQUES

### ▪ MANIPULATIONS DE STYLES

masquer ou réafficher un élément : `.hide()` ou `.show()`

avec possibilité de faire basculer entre chaque état : `.toggle()`

modification de l'apparence du pointeur de la souris (CSS 2) :

`cursor:` avec apparence normale (`default`) ou sur un lien (`pointer`)

### ▪ GESTION DE L'ÉVÉNEMENT DU CLIC SUR UNE TÊTE DE SOUS-LISTE

sélection des éléments en tête de sous-liste : `$('.li:has(ul)')`

et à l'opposé ceux sans sous-liste : `$('.li:not( :has(ul) )')`

masquer ou réafficher un élément : `.hide()` ou `.show()`

avec possibilité de faire basculer entre chaque état : `.toggle()`

gestion de l'événement du clic sur une tête de sous-liste

avec dans le gérant, identification de l'élément cliqué : `event.target`

afin de distinguer une propagation à partir d'un élément fils

et avec annulation de la remontée aux ancêtres de la tête cliquée

déclenchement initial pour masquer les sous-listes : `.click()`

# EXEMPLE DE LA LISTE AVEC JQUERY

---

## • CODE XHTML DE LA LISTE AVEC LES SOUS-LISTES

```
<ul>
<li>Bactéries</li>
<li>Algues</li>
<li>Champignons et lichens</li>
<li>Plantes<ul>
  <li>Spermatophyta<ul>
    <li>Gymnospermes</li>
    <li>Angiospermes<ul>
      <li>Monocotylédones</li>
      <li>Dicotylédones</li>
    </ul></li>
  </ul></li>
<li>Bryophyta</li>
<li>Pteridophyta</li>
</ul></li>
<li>Animaux</li>
<li>Autres ou inclassables</li>
</ul>
```

## • SCRIPT

```
$(function() {
  $('li:has(ul)') // tete de sous-liste
    .css('font-weight', 'bold')
    .css('cursor', 'pointer')
    .click( function (event) {
      if (this == event.target) {
        $(this).children().toggle();
      }
      // annulation de la propagation
      return false;
    })
    .click(); // masquage initial
  $('li:not( :has(ul) )') // element sans sous-liste
    .css('font-weight', 'normal')
    .css('cursor', 'default');
});
```

# EXEMPLE D'AJAX AVEC JQUERY

## FORMULAIRE DE CHARGEMENT DANS LA PAGE VIA AJAX



### • CARACTÉRISTIQUES

#### ▪ MANIPULATIONS DE XMLHTTPREQUEST EN JQUERY

opérations simplifiées : `$.get()`, `$.post()` et `.load()`

opération avec contrôle complet : `$.ajax()`

#### ▪ INSTRUCTIONS `$.get()` ET `$.post()`

récupération du contenu à une adresse via HTTP selon la méthode,  
avec l'envoi éventuel de données et l'indication d'un gérant de la fin  
de l'échange (appelé uniquement si aucune erreur)

appels :

```
$.get(adresse, données, gérant)
```

```
$.post(adresse, données, gérant)
```

paramétrage :

*adresse* ..... URL du contenu à récupérer

*données* ..... données dans un tableau associatif, sinon `null`

*gérant* ..... fonction avec en paramètres le texte du contenu  
reçu et le libellé du résultat de l'échange en HTTP

## EXEMPLE D'AJAX AVEC JQUERY : XHTML

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
 xml:lang="fr" lang="fr">

<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=iso-8859-1" />
  <title>Essai avec Ajax et jQuery</title>
  <style type="text/css"><!--
    #contenu, #resultat
    { border: gray solid thin; padding: 0.5em; margin: 0.5em 0; }
  --></style>
</head>

<body>
<h1>Essai avec Ajax et jQuery</h1><hr />
<form action="" id="f">
  <p>Fichier
    <input type="file" name="zfichier" id="zfichier" size="80" />
    <input type="button" value="Charger" id="boutonfichier" /></p>
  <p>URL
    <input type="text" name="zurl" id="zurl" size="90" value="" />
    <input type="button" value="Charger" id="boutonurl" /></p>
</form>
<div id="resultat"></div>
<div><pre id="contenu"></pre></div>
<hr />
<address>MC 2 XII 2009</address>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript"
  src="jq-essai-ajax-v1.js"></script>
</body>

</html>
```

## EXEMPLE D'AJAX AVEC JQUERY : SCRIPT

---

```
/* jq-essai-ajax-v1.js - MC 3 XII 2009 */

$(function () {
  $('#f').submit( function() {
    return false;
  });
  $('#boutonfichier').click( function () {
    charger( $('#zfichier').val() );
  });
  $('#boutonurl').click( function () {
    charger( $('#zurl').val() );
  });
  $('#zurl').val(window.location);
});

function resultatAjax(texte, etat) {
  // traitement de la fin de l'echange via Ajax
  // avec un message dans l'element #resultat
  $('#resultat').append('<p>R&eacute;sultat : ' + etat + '</p>');
  try {
    $('#contenu').text(texte);
  } catch(erreur) { // cas de Internet explorer
    $('#contenu').html(texte);
  }
}

function charger(adresse) {
  // chargement du texte issu de adresse dans l'element #contenu
  // avec des messages dans l'element #resultat
  $('#resultat').html('<p>Chargement de <strong>' + adresse
  + '</strong></p>');
  $.get(adresse, null, resultatAjax);
}
```

# EXEMPLE D'AJAX AVEC JQUERY (SUITE)

---

## AUTRES OPÉRATIONS

### ■ INSTRUCTION `$.ajax (paramétrage)`

récupération du contenu à une adresse via HTTP avec de nombreuses possibilités (gérant d'erreur, type de données etc.)

paramétrage sous la forme d'un tableau associatif :

```
{ nom1: valeur1, nom2: valeur2, ... nomN: valeurN }
```

paramètres (principaux) :

`url: adresse`..... URL du contenu à récupérer

`type: méthode`..... méthode de l'envoi : "GET" (par défaut) ou "POST"

`dataType: type`..... nature du contenu à récupérer, principalement :

"html", "xml", "text" ; si non indiqué, choix entre "html" et "xml" fixé par examen du contenu

`cache: logique`..... autorisation de récupérer la copie du contenu si déjà dans le cache du navigateur (vrai par défaut)

`error: gérant`..... fonction appelée en cas d'erreur lors de l'échange, avec en paramètres l'objet XMLHttpRequest, le libellé du résultat de l'échange en HTTP et un objet associé à l'erreur

`success: gérant`.... fonction appelée en fin normale de l'échange, avec en paramètres le texte du contenu reçu et le libellé du résultat de l'échange en HTTP

### ■ MÉTHODE `.load (adresse, données, gérant)`

récupération du contenu à une adresse avec envoi éventuel de données et possibilité de définir un gérant de la fin de l'échange ; le contenu récupéré est inséré au format HTML dans l'élément considéré

notations :

`.load (adresse)` ..... échange via la méthode "GET"

`.load (adresse, null, gérant)` ..... échange via la méthode "GET" avec définition d'un gérant

`.load (adresse, données)` ..... échange via la méthode "POST" avec des données dans un tableau associatif

`.load (adresse, données, gérant)` ..... échange via la méthode "POST" avec des données et définition d'un gérant

## EXEMPLE D'AJAX AVEC JQUERY : SCRIPT (BIS)

---

```
/* jq-essai-ajax-v2.js - MC 2 XII 2009 */

$(function () {
  $('#f').submit( function() {
    return false;
  });
  $('#boutonfichier').click( function () {
    charger( $('#zfichier').val() );
  });
  $('#boutonurl').click( function () {
    charger( $('#zurl').val() );
  });
  $('#zurl').val(window.location);
});

function resultatAjax(texte, etat) {
// traitement du resultat du chargement via Ajax
// placement du texte charge dans l'element #contenu
// avec un message dans l'element #resultat
var erreur;
$('#resultat').append('<p>R&acute;sultat : ' + etat + '</p>');
try {
  $('#contenu').text(texte);
} catch(erreur) { // cas de Internet explorer
  $('#contenu').html(texte);
}

function erreurAjax(xhr, etat, erreur) {
// avec un message dans l'element #resultat
// traitement du resultat du chargement via Ajax
var message = 'Anomalie : ' +
  (etat ? etat + ', ' : '') +
  'HTTP ' + xhr.status + ' ' + xhr.statusText + ' ' +
  ('undefined' != typeof erreur) ?
  ', ' + erreur.toString() : '';
$('#resultat').append('<p>' + message + '</p>');
}

function charger(adresse) {
// chargement du texte issu de adresse dans l'element #contenu
// avec des messages dans l'element #resultat
$('#resultat').html(
  '<p>Chargement de <strong>' + adresse + '</strong></p>');
$('#contenu').html('');
$.ajax( {
  url: adresse, type: "GET", dataType: "text", cache: false,
  success: resultatAjax, error: erreurAjax
} );
}
```

# JQUERY : APPELS

---

## PRÉSENTATION SIMPLIFIÉE DE LA VERSION 1.3 DE JQUERY

### • APPELS DE JQUERY

#### ▪ DÉFINITION D'UNE FONCTION INITIALE

fonction exécutée à l'issue du chargement du document :

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

ou en raccourci : `$( function () { actions } ) ;`

possibilités de plusieurs fonctions initiales (exécution en séquence)

exemple :

```
$( function () {
  $('#extra').hide();
}) ;
```

#### ▪ SÉLECTION D'ÉLÉMENTS DU DOCUMENT

soit via la notation d'une sélection selon la syntaxe de jQuery :

`jQuery(sélection)` ou en raccourci : `$(sélection)`

soit en indiquant un objet du document (DOM 0) :

`jQuery(objet)` ou en raccourci : `$(objet)`

le résultat de la sélection est un tableau des éléments sélectionnés, chaque élément correspondant à un noeud de l'arbre DOM

exemples :

```
$('#extra').append('<p>Alerte !</p>');
$(document.links).css('color', 'red');
```

#### ▪ CRÉATION D'ÉLÉMENTS

éléments créés à partir d'un fragment codé en XHTML :

`jQuery(fragment)` ou en raccourci : `$(fragment)`

exemple : `$( '<p>Alerte !</p>' ).appendTo("#extra") ;`

#### ▪ ACTIONS SUR DES ÉLÉMENTS

possibilités de séquence d'actions appliquées l'une après l'autre

exemple :

```
$('#div#extra').addClass("marque").show();
```

# JQUERY : CONTENU ET ATTRIBUTS D'ÉLÉMENT

---

## PRINCIPALES MÉTHODES

### • CONTENU D'ÉLÉMENT

- CONTENU (EN HTML) D'ÉLÉMENT (ET LES FILS) : `.html()`

**exemple** : `texte = $('div#contenu').html();`

- MODIFICATION DU CONTENU (EN HTML) D'ÉLÉMENT : `.html(texte)`

**exemple** : `($('div#contenu').html(<p>Calendrier</p>);`

- CONTENU DU TEXTE FILS D'ÉLÉMENT : `.text()`

**exemple** : `signature = $('address').text();`

- MODIFICATION DU CONTENU D'ÉLÉMENT : `.text(texte)`

avec recodage des occurrences de `<` et `>` (par `$lt;` et `$gt;`)

**exemple** : `($('div#contenu').text(extrait);`

- AJOUT EN FIN OU AU DÉBUT DU CONTENU :

appliqué soit à la sélection à modifier : `.append(texte)` .`prepend(texte)`

soit à la sélection à ajouter : `.appendTo(sélection)` .`prependTo(sélection)`

**exemples** : `($('div#contenu').append('<p>Salle</p>');`

`($('p').prependTo('div');`

### • ATTRIBUTS D'ÉLÉMENT

- TEXTE DE LA VALEUR D'UN ATTRIBUT : `.attr(nom)`

si appliqué à plusieurs éléments sélectionnés, seul le 1<sup>er</sup> est considéré

**exemple** : `texte = $('table#fleurs').attr('summary');`

- MODIFICATION D'UN ATTRIBUT : `.attr(nom, valeur)`

**exemple** : `($('img').attr('alt', ' (photo) ');`

- MODIFICATION DE PLUSIEURS ATTRIBUTS : `.attr(liste)`

liste indiquée sous la forme d'un tableau associatif

**exemple** : `($('img').attr({ alt: ' (photo) ', title: 'Photo' }));`

- MODIFICATION ITÉRATIVE D'UN ATTRIBUT : `.attr(nom, fonction)`

fonction retournant la valeur pour chaque élément de la sélection, avec l'élément en contexte (`this`) et en paramètre son rang (à partir de 0)

**exemple** : `($('div').attr('id', function(rang) { return 'd' + rang; }));`

- SUPPRESSION D'UN ATTRIBUT : `.removeAttr(nom)`

**exemple** : `($('img').removeAttr('title'));`

# JQUERY : STYLES D'ÉLÉMENT

---

## PRINCIPALES MÉTHODES

### • PROPRIÉTÉS DE STYLES

- VALEUR D'UNE PROPRIÉTÉ D'ÉLÉMENT : `.css (propriété)`  
**exemple** : `$( 'body' ).css('color');`
- MODIFICATION D'UNE PROPRIÉTÉ : `.css (propriété, valeur)`  
**exemple** : `$( 'body' ).css('color', 'black');`
- MODIFICATION DE PLUSIEURS PROPRIÉTÉS : `.css (propriétés)`  
avec les propriétés notées sous la forme d'un tableau associatif  
**exemple** : `$( 'body' ).css( { color: 'black', font: '90%' } );`
- HAUTEUR OU LARGEUR D'ÉLÉMENT (PIXELS) : `.height() .width()`  
**exemple** : `hauteur = $( 'div#menu' ).height();`
- MODIFICATION DE HAUTEUR, LARGEUR : `.height (valeur) .width (valeur)`  
**exemple** : `$( 'div#menu' ).width(largeur);`
- POSITION D'ÉLÉMENT (EN PIXELS) : `.offset()`  
renvoi d'un tableau associatif avec position verticale (`top`) et  
horizontale (`left`) en pixels, à partir du coin haut gauche du parent  
**exemple** : `positionVerticale = $( 'h1' ).offset().top;`

### • CLASSES DE STYLES

- AJOUT DE CLASSE(S) : `.addClass (liste)`  
liste : chaîne avec un nom, ou plusieurs séparés par espace(s)  
**exemple** : `$( 'div#p' ).addClass('fort libre');`
- RETRAIT DE CLASSE(S) : `.removeClass (liste)`  
liste : chaîne avec un nom, ou plusieurs séparés par espace(s)  
**exemple** : `$( 'div' ).removeClass('fort libre');`
- AJOUT SI ABSENT OU RETRAIT SI PRÉSENT DE CLASSE : `.toggleClass (nom)`  
le nom correspond à une seule classe  
**exemple** : `$( 'div' ).toggleClass('libre');`
- TEST DE PRÉSENCE D'UNE CLASSE : `.hasClass (nom)`  
si plusieurs éléments, renvoit vrai dès qu'au-moins l'un a la classe  
**exemple** : `$( 'div' ).hasClass('libre');`

# JQUERY : ÉVÉNEMENTS ET ZONE DE FORMULAIRE

---

## PRINCIPALES MÉTHODES

### • ÉVÉNEMENTS

#### ▪ ASSOCIATION D'UN GÉRANT À UN ÉVÉNEMENT : `.nom (gérant)`

**principaux événements disponibles** : blur change click focus load  
mouseover mouseout select submit unload

**gérant** : fonction avec en paramètre l'objet associé à l'événement,  
possibilités de plusieurs gérants pour le même événement

**principales propriétés** de l'objet associé à un événement :

**type** (nom de l'événement) et **target** (élément en cible)

**et méthodes** : `.preventDefault()` (pas de prise en charge ordinaire)

`.stopPropagation()` (arrêt de la propagation)

**exemple** :

```
function gererClic (evenement) {  
    afficher(evenement.target);  
    evenement.stopPropagation();  
}  
$('input#boutonVoir').click(gererClic);
```

#### ▪ DÉCLENCHEMENT D'UN ÉVÉNEMENT : `.nom()` ou `.trigger(nom)`

**principaux événements disponibles** : blur change click focus  
mouseover mouseout select submit

**exemple** : `($('input#boutonVoir').click();`

#### ▪ ANNULATION D'UN GÉRANT : `.unbind(nom, gérant)`

**exemple** : `($('input#boutonVoir').unbind('click', gererClic);`

#### ▪ ANNULATION DU OU DES GÉRANTS POUR UN ÉVÉNEMENT : `.unbind(nom)`

**exemple** : `($('input').unbind('click');`

### • ZONE DE FORMULAIRE

#### ▪ VALEUR D'UNE ZONE DE FORMULAIRE : `.val()`

tableau de valeurs dans le cas d'une liste à choix multiple

**exemple** : `nom = $('input#znom').val();`

#### ▪ MODIFICATION DE LA VALEUR D'UNE ZONE DE FORMULAIRE : `.val(valeur)`

**exemple** : `($('input#znom').val( nom.toUpperCase() ) ;`

#### ▪ MODIFICATION DE LA VALEUR D'UNE ZONE DE FORMULAIRE : `.val(valeur)`

un tableau en valeur correspond à des éléments de liste (`<select>`) ou  
des cases d'option ("checkbox") ou des cases à cocher ("radio")

**exemples** : `($('input#znom').val( nom.toUpperCase() ) ;`

`($('form').val( ['etudiant', 'F'] ) ;`

# JQUERY : MANIPULATIONS DE SÉLECTION

---

## PRINCIPALES MÉTHODES

### • MODIFICATION D'UNE SÉLECTION

- RÉDUCTION À L'ÉLÉMENT EN RANG (COMPTÉ À PARTIR DE ZÉRO) : `.eq(rang)`  
**exemple** : `($('p').eq(0).css('color', 'gray');`
- RÉDUCTION AUX ÉLÉMENTS DE LA SOUS-SÉLECTION : `.filter(sous-sélection)`  
**exemple** : `($('ul').show().filter('.detail').hide();`
- EXCLUSION DES ÉLÉMENTS SELON LA SOUS-SÉLECTION : `.not(sous-sélection)`  
**exemple** : `($('div').show().not('#menu').css('font', '80%');`
- AJOUT D'ÉLÉMENTS SELON LA SOUS-SÉLECTION : `.add(sous-sélection)`  
**exemple** : `($('p').show().add('h3').css('margin', '1em');`

### • FAMILLE D'UNE SÉLECTION

- ÉLÉMENTS DES FILS DIRECTS : `.children()`  
avec la possibilité de les filtrer : `.children(sous-sélection)`  
**exemple** : `($('div.partie').children().css('padding', '0.5em');`
- ÉLÉMENTS DES DESCENDANTS : `.contents()`  
le document contenu dans le cas d'un cadre interne (`<iframe>`)  
**exemple** : `page = $('iframe').contents().html();`
- ÉLÉMENTS DES DESCENDANTS SELON SOUS-SÉLECTION : `.find(sous-sélection)`  
**exemple** : `($('p').find('span').css('background-color', 'gray');`
- ÉLÉMENTS DES PARENTS DIRECTS : `.parent()`  
avec la possibilité de les filtrer : `.parent(sous-sélection)`  
**exemple** : `($('p').parent().css('padding', '0');`
- ÉLÉMENTS DES ANCÊTRES (SAUF LA RACINE) : `.parents()`  
avec la possibilité de les filtrer : `.parents(sous-sélection)`  
**exemple** : `($('h1').parents().css('padding', '0');`
- ÉLÉMENTS DU FRÈRE IMMÉDIATEMENT APRÈS, AVANT : `.next() .prev()`  
**exemple** : `($('hr').next().css('margin-top', '0.5em');`
- ÉLÉMENTS DES FRÈRES SITUÉS APRÈS, AVANT : `.nextAll() .prevAll()`  
**exemple** : `($('div#menu').nextAll().hide();`
- ÉLÉMENTS DE TOUS LES FRÈRES : `.siblings()`  
**exemple** : `($('h3').siblings().css('padding', '1em');`

# JQUERY : ANIMATIONS

---

## PRINCIPALES MÉTHODES

### • ANIMATIONS

- NOTATION D'UNE VITESSE : 'slow' ou 'normal' ou 'fast' ou *nombre*  
le nombre correspond à la durée totale exprimée en millisecondes
- DISPARITION PROGRESSIVE PAR RÉDUCTION : `.hide(vitesse)`  
possibilité de gérant de fin de l'animation : `.hide(vitesse, gérant)`  
exemple : `$('#h1').hide('slow');`
- APPARITION PROGRESSIVE PAR AGRANDISSEMENT : `.show(vitesse)`  
possibilité de gérant de fin de l'animation : `.show(vitesse, gérant)`  
exemple : `$('#h1').show('slow');`
- EFFACEMENT PROGRESSIF : `.fadeOut(vitesse)`  
possibilité de gérant de fin de l'animation : `.fadeOut(vitesse, gérant)`  
exemple : `$('#h1').fadeOut('slow');`
- TRACÉ PROGRESSIF : `.fadeIn(vitesse)`  
possibilité de gérant de fin de l'animation : `.fadeIn(vitesse, gérant)`  
exemple : `$('#h1').fadeIn('slow');`
- DISPARITION PROGRESSIVE PAR REMONTÉE : `.slideUp(vitesse)`  
possibilité de gérant de fin de l'animation : `.slideUp(vitesse, gérant)`  
exemple : `$('#h1').slideUp('slow');`
- APPARITION PROGRESSIVE PAR DESCENTE : `.slideDown(vitesse)`  
possibilité de gérant de fin de l'animation : `.slideDown(vitesse, gérant)`  
exemple : `$('#h1').slideDown('slow');`
- ARRÊT DES ANIMATIONS COURANTES : `.stop()`  
exemple : `$('#h1').stop();`

# JQUERY : SÉLECTIONS À BALISAGE

---

CHAÎNE DE CARACTÈRES EN PARAMÈTRE DE `jQuery()` OU `$()`

## • EXPRESSIONS DE SÉLECTION À L'AIDE DU BALISAGE

### ■ NOTATION SIMILAIRE À CELLE DU LANGAGE CSS

sélection portant sur le balisage en XHTML (balises, attributs, valeurs)

### ■ GRAMMAIRE

nom

`*` ..... n'importe quel élément

`balise` ..... le type d'élément désigné par sa balise

chaîne

`texte` ..... le texte

`"texte"` ou `'texte'` ... le texte délimité

filtre

`#identificateur` ..... identifié par l'identificateur

`.classe` ..... avec la classe de style

`[attribut]` ..... avec l'attribut présent

`[attribut = chaîne]` ..... avec l'attribut valant exactement la chaîne

`[attribut != chaîne]` ..... avec l'attribut de valeur différente de la chaîne

`[attribut *= chaîne]` ..... avec l'attribut de valeur contenant la chaîne

`[attribut ^= chaîne]` ..... avec l'attribut de valeur débutant par la chaîne

`[attribut $= chaîne]` ..... avec l'attribut de valeur finissant par la chaîne

`:has (nom)` ..... avec au-moins un descendant désigné par nom

filtrage

`filtre` ..... correspondant au filtre

`filtrage filtre` ..... correspondant au filtrage puis au filtre <sup>1</sup>

terme

`filtrage` ..... tous les éléments correspondant au filtrage

`nom filtrage` ..... le type d'élément correspondant au filtrage <sup>1</sup>

facteur

`terme` ..... le terme

`facteur terme` ..... le terme en descendant du facteur

`facteur > terme` ..... le terme en filiation directe du facteur

`facteur ~ terme` ..... le terme en frère du facteur

`facteur + terme` ..... le terme en frère immédiat du facteur

expression

`facteur` ..... le facteur

`expression, facteur` ..... l'expression et le facteur

---

<sup>1</sup> Dans l'écriture, il n'y a pas d'espace entre les filtres successifs qui sont accolés les uns aux autres.

# JQUERY : SÉLECTIONS À BALISAGE (SUITE)

---

## EXEMPLES DE SÉLECTIONS À L'AIDE DU BALISAGE

`$( 'h1' )` ..... titres de 1<sup>er</sup> niveau

`$( 'h1, h2, h3' )` ..... titres de 1<sup>er</sup>, 2<sup>ème</sup> et 3<sup>ème</sup> niveau

`$( 'p.important' )` ..... paragraphes avec la classe de style `important`

`$( '*.important.decale' )` ..... éléments avec les classes de style `important` et `decale`

`$( 'div#menu.anglais' )` ..... division identifiée par `menu` et avec la classe de style `anglais`

`$( 'a[id]' )` ..... liens avec l'attribut `id`

`$( 'a[href$=.exe]' )` ..... liens avec une adresse finissant en `.exe`

`$( '[title*=photo]' )` ..... éléments avec un titre contenant `photo`

`$( 'img[alt][href$=gif]' )` .... images avec l'attribut `alt` et avec une adresse finissant en `gif`

`$( 'ul:has(a)' )` ..... listes à puces contenant un lien

`$( 'ul:has(a).decale' )` ..... listes à puces contenant un lien, et aussi avec la classe de style `decale`

`$( 'ul a' )` ..... liens contenus dans une liste

`$( 'div > a' )` ..... liens présents directement dans une division (mais pas dans un élément intermédiaire comme un titre de la division)

`$( 'h2 + h3' )` ..... titres de 3<sup>ème</sup> niveau immédiatement précédés par un titre de 2<sup>ème</sup> niveau

`$( 'li ~ li' )` ..... élément de liste mais pas le 1<sup>er</sup> de la liste

# JQUERY : SÉLECTIONS SELON DOM

---

## FILTRES COMPLÉMENTAIRES POUR UNE EXPRESSION DE SÉLECTION

### • EXPRESSIONS DE SÉLECTION SELON DOM

#### ■ RÉFÉRENCE À DOM

sélection portant sur la position dans l'arbre du document selon DOM et sur tous les éléments du document (à partir du noeud `<html>`)

#### ■ FILTRES

<code>:first</code>	.....	la première occurrence dans l'arbre
<code>:last</code>	.....	la dernière occurrence dans l'arbre
<code>:first-child</code>	.....	en position de 1 <sup>er</sup> fils
<code>:last-child</code>	.....	en position de dernier fils
<code>:only-child</code>	.....	en position de fils unique
<code>:nth-child(K)</code>	.....	en position de K <sup>ième</sup> fils (compté à partir de 1)
<code>:nth-child(even)</code>	.....	en position de fils de rang pair
<code>:nth-child(odd)</code>	.....	en position de fils de rang impair
<code>:nth-child(Kn)</code>	.....	en position de fils de rang multiple de K
<code>:nth-child(Kn+D)</code>	.....	fils au rang D après un multiple de K
<code>:eq(rang)</code>	.....	noeud en rang (compté à partir de 0, <code>&lt;html&gt;</code> )
<code>:even</code>	.....	à un rang pair de noeud
<code>:odd</code>	.....	à un rang impair de noeud
<code>:gt(rang)</code>	.....	noeuds au-delà de celui en rang (non compris)
<code>:lt(rang)</code>	.....	noeuds avant celui en rang (non compris)

#### ■ EXEMPLES

<code>\$('.h2:first')</code>	.....	le premier titre de deuxième niveau
<code>\$('.tr:even')</code>	.....	les lignes en rang pair d'un tableau
<code>\$('.tr:even')</code>	.....	les lignes en rang pair d'un tableau
<code>\$('.div p:only-child')</code>	..	les paragraphes seuls éléments d'une division
<code>\$('.li:nth-child(2)')</code>	....	les éléments de liste en 2 <sup>ème</sup> position

# JQUERY : SÉLECTIONS SPÉCIFIQUES

---

## FILTRES COMPLÉMENTAIRES POUR UNE EXPRESSION DE SÉLECTION

### • EXPRESSIONS DE SÉLECTION SPÉCIFIQUES À JQUERY

#### ■ FILTRES PORTANT SUR LA NATURE DES ÉLÉMENTS

axé principalement sur la manipulation des composants de formulaire

#### ■ CAS DE FORMULAIRE

:input .....	composant (<input>, <select>, <textarea>, <button>)
:text .....	composant de saisie sur une seule ligne
:password .....	composant de saisie en aveugle
:radio .....	case de choix exclusif
:checkbox .....	case d'option
:submit .....	bouton de validation
:reset .....	bouton de réinitialisation
:button .....	bouton de commande (<input> avec type à "submit", "reset" ou "button", et aussi <button>)
:image .....	bouton à image (<input> avec type à "image")
:file .....	composant d'envoi de fichier
:checked .....	case de choix exclusif ou d'option, cochées
:selected .....	option de liste, sélectionnée
:enabled .....	composant actif (<input> <textarea> <option> <button>)
:disabled .....	composant inactif (<input> <textarea> <option> <button>)

#### ■ AUTRES CAS

:header .....	élément de titre (<h1> à <h6>)
:parent .....	élément possédant au-moins un fils (ou un contenu)
:empty .....	élément sans descendance (et sans contenu)
:contains( <i>texte</i> ) ...	élément où texte est présent dans le contenu de l'élément ou d'un de ses descendants
:hidden .....	élément caché (y compris dans l'entête, <head>)
:visible .....	élément visible (non caché)
:animated .....	élément avec une animation en cours
:not( <i>filtre</i> ) .....	négation d'un filtre

## JQUERY : SÉLECTIONS SPÉCIFIQUES (SUITE)

---

### • EXEMPLES DE SÉLECTIONS SPÉCIFIQUES

`$( 'option:selected' )` ..... options sélectionnées dans une liste

`$( ':input[name*=CP]' )` ..... composants de formulaire dont le nom contient le fragment de texte CP

`$( ':password:visible' )` ... composants visibles de saisie en aveugle

`$( ':text[size]' )` ..... composants de saisie sur une ligne avec une largeur spécifiée

`$(input:disabled')` ..... cases d'options ou de choix désactivées

`$( 'fieldset :input' )` ..... composants dans un regroupement

`$( 'form#f :reset' )` ..... bouton de réinitialisation du formulaire f

`$( 'div *:header' )` ..... titres dans une division

`$( 'body *:hidden' )` ..... les éléments invisibles du corps de la page

`$( ':contains("cf.")' )` ..... éléments contenant le fragment de texte cf.

`$( 'p:animated' )` ..... paragraphes avec une animation en cours

`$( 'img:not([alt])' )` ..... les images sans l'attribut alt

`$( 'td:parent' )` ..... cellules de données avec un contenu

`$( 'tr:not(:has(th))' )` ..... lignes d'un tableau, qui ne contiennent pas de cellules d'entête

## EXERCICES DE SÉLECTION

---

1. Tous les éléments de citation
2. Les titres possédant un identificateur
3. Les scripts internes à la page
4. Les divisions vides
5. Les divisions ne contenant pas d'images
6. Les composants de saisie sur plusieurs lignes avec un texte proposé
7. Tous les composants de formulaire sélectionnés
8. Les cases cochées dont le nom débute par `z`
9. Le premier élément au niveau d'une sous-liste, qui ne contient pas lui-même de sous-liste
10. Les images précédées de 2 autres images

# INDEX

---

#, 34

\$( ), 12, 13, 28

\$.ajax(), 23, 26

\$.get(), 23

\$.post(), 23

\*, 34

,, 34

., 34

:animated, 37

:button, 37

:checkbox, 37

:checked, 37

:disabled, 37

:empty, 37

:enabled, 37

:eq(), 36

:even, 36

:file, 37

:first, 36

:first-child, 36

:gt(), 36

:has(), 34

:header, 37

:hidden, 37

:image, 37

:input, 37

:last, 36

:last-child, 36

:lt(), 36

:not(), 37

:nth-child(), 36

:odd, 36

:only-child, 36

:parent, 37

:password, 37

:radio, 37

:reset, 37

:selected, 37

:submit, 37

:text, 37

:visible, 37

[!=], 34

[=\$], 34

[\*=], 34

[], 34

[^=], 34

[=], 34

~, 34

+, 34

<button>, 37

<h1>, 37

<iframe>, 18, 32

<input>, 37

<option>, 37

<textarea>, 37

>, 34

abort(), 4

ActiveXObject, 4

add(), 32

addClass(), 15, 30

Ajax, 3

animation, 16

Animations, 33

append(), 29

appendTo(), 29

attr(), 29

attribut, 29

blur, 31

button, 37

cadre, 32

callback, 17

change, 31

checkbox, 31

children(), 32

click, 31

click(), 21

compatibilité, 11

contents(), 18

Contenu, 29

css(), 12, 30

cursor:, 21

document, 13, 28

DOM, 28

durée, 33

each(), 18

enchaînement, 16, 28

eq(), 32

even, 36

événement, 31

event, 14

event handler, 14

fadeIn(), 16  
fadeOut(), 16, 33  
fast, 16, 33  
filter(), 32  
filtre, 34, 36, 37  
find(), 18, 32  
focus, 31  
fonction en Javascript, 13  
formulaire, 37  
function literal, 13  
  
gérant, 31  
gestionnaire d'événement, 14  
  
hasClass(), 30  
height(), 30  
hide(), 21, 33  
html(), 29  
HTTP, 3, 23  
  
image, 37  
  
Javascript non intrusif, 11  
jQuery, 11  
jQuery(), 12, 28  
  
listener, 14  
load, 18, 31  
load(), 23, 26  
  
mouseout, 31  
mouseover, 31  
  
next(), 32  
nextAll(), 32  
normal, 16, 33  
not(), 32  
  
odd, 36  
offset(), 30  
onreadystatechange, 4  
open(), 4  
  
parent(), 32  
parents(), 32  
prepend(), 29  
prependTo(), 29  
prev(), 32  
prevAll(), 32  
preventDefault(), 31  
  
radio, 31  
ready, 31  
ready(), 13, 28  
readyState, 4  
removeAttr(), 29  
removeClass(), 17, 30  
reset, 37  
responseText, 4  
responseXML, 4  
RIA, 11  
  
same origin policy, 3  
select, 31  
sélection, 32, 34  
  
send(), 4  
show(), 21  
siblings(), 32  
slideDown(), 33  
slideUp(), 33  
slow, 16, 33  
status, 4  
statusText, 4  
stop(), 33  
stopPropagation(), 31  
submit, 31, 37  
  
target, 21, 31  
text(), 29  
this, 18  
toggle(), 21  
toggleClass(), 30  
trigger(), 31  
type, 31  
  
unbind(), 31  
unload, 31  
unobtrusive, 11  
URL, 23  
  
val(), 14, 31  
  
width(), 30  
wrapper set, 12  
  
XHTML, 28, 34  
XMLHttpRequest, 4, 23

## SOMMAIRE

|  |    |
|--|----|
| AJAX .....   | 2  |
| Présentation de Ajax .....                           | 3  |
| Spécifications de XMLHttpRequest .....               | 4  |
| Exemple d'utilisation de Ajax .....                  | 5  |
| Exemple d'utilisation de Ajax : code XHTML .....     | 6  |
| Exemple d'utilisation de Ajax : script .....         | 7  |
| Exemple d'utilisation de Ajax : script (suite) ..... | 8  |
| Exemple d'utilisation de Ajax : script (fin) .....   | 9  |
| JQUERY .....   | 10 |
| Présentation de jQuery .....                         | 11 |
| Exemple du coloriage (version n° 1) .....            | 12 |
| Exemple du coloriage (version n° 2) .....            | 13 |
| Exemple du coloriage (version n° 3) .....            | 14 |
| Exemple du marquage (version n° 1) Texte .....       | 15 |
| Exemple du marquage (version n° 2) .....             | 16 |
| Exemple du marquage (version n° 3) .....             | 17 |
| Exemple du marquage (version n° 4) .....             | 18 |
| Exemple du marquage (version n° 4) : HTML .....      | 19 |
| Exemple du marquage (version n° 4) : script .....    | 20 |
| Exemple de la liste avec jQuery .....                | 21 |
| Exemple de la liste avec jQuery .....                | 22 |
| Exemple d'Ajax avec jQuery .....                     | 23 |
| Exemple d'Ajax avec jQuery : XHTML .....             | 24 |
| Exemple d'Ajax avec jQuery : script .....            | 25 |
| Exemple d'Ajax avec jQuery (suite) .....             | 26 |
| Exemple d'Ajax avec jQuery : script (bis) .....      | 27 |
| jQuery : appels .....                                | 28 |
| jQuery : contenu et attributs d'élément .....        | 29 |
| jQuery : styles d'élément .....                      | 30 |
| jQuery : événements et zone de formulaire .....      | 31 |
| jQuery : manipulations de sélection .....            | 32 |
| jQuery : animations .....                            | 33 |
| jQuery : sélections à balisage .....                 | 34 |
| jQuery : sélections à balisage (suite) .....         | 35 |
| jQuery : sélections selon DOM .....                  | 36 |
| jQuery : sélections spécifiques .....                | 37 |
| jQuery : sélections spécifiques (suite) .....        | 38 |
| Exercices de sélection .....                         | 39 |
| Index .....  | 40 |