Développement Web

Manipulation de l'arbre DOM

LI328
Laure Soulier
slides de Sylvain Lamprier

UPMC

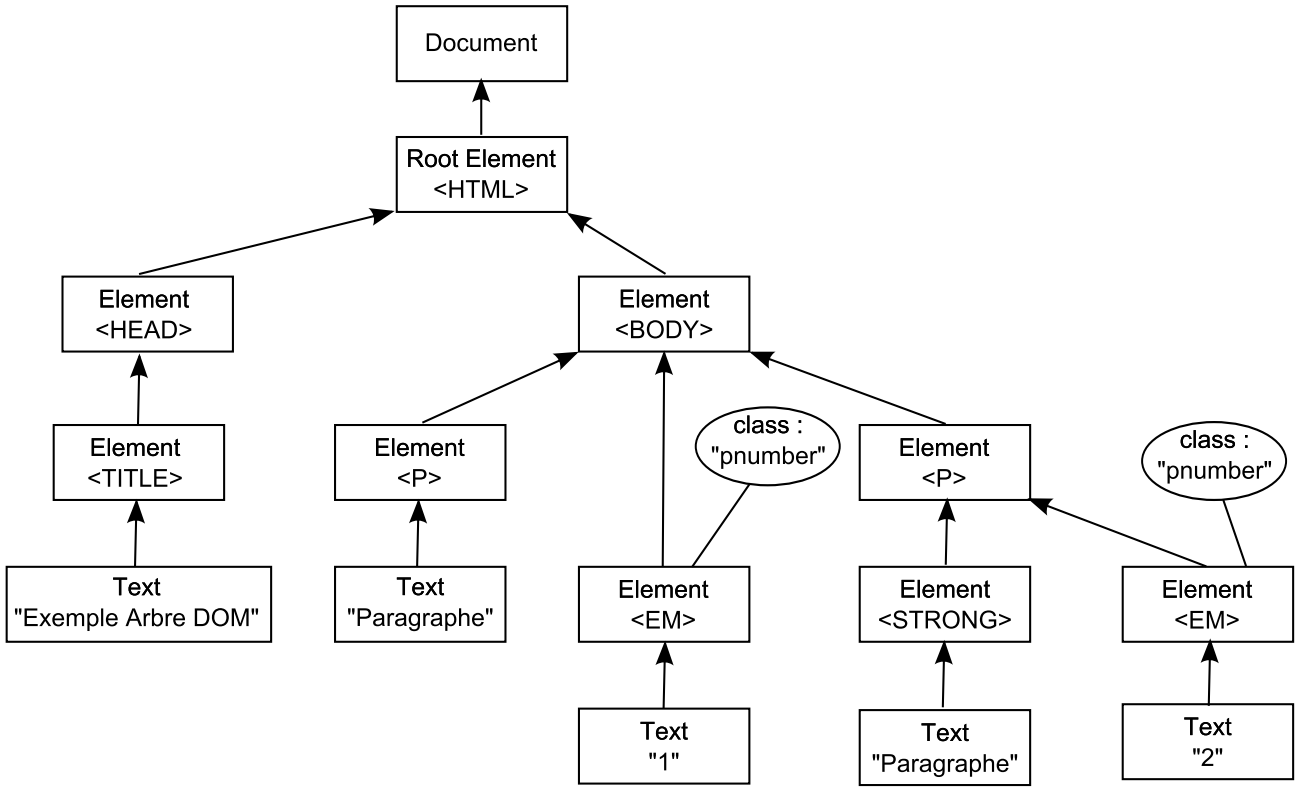
Arborescences Web

- Arbre DOM (Document Object Model)
  - Standard du W3C
  - Décrit une interface indépendante de tout langage de programmation

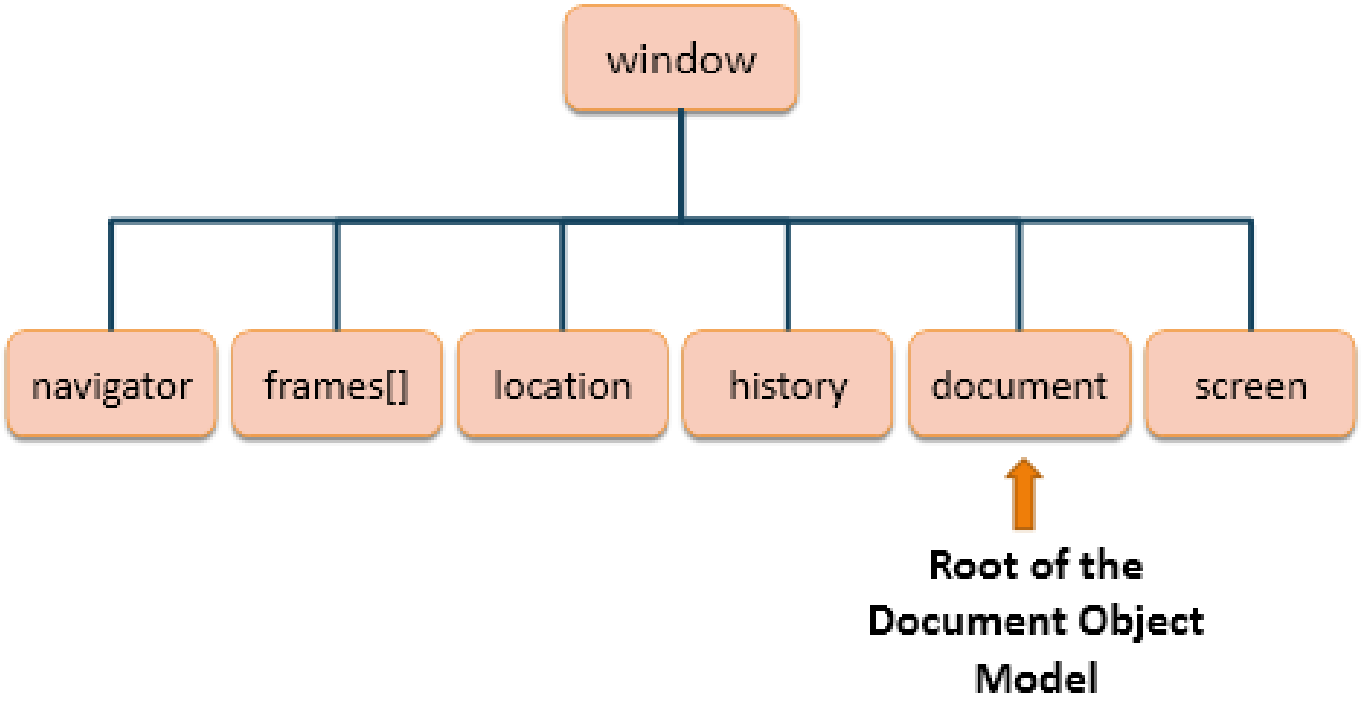
  ⇒ Accéder ou de mettre à jour le contenu, la structure ou le style de documents XML et HTML
- Arbre BOM (Browser Object Model)
  - Pas de standard sur la manière de gérer les éléments du navigateur
  - Nécessité de produire du code javascript adapté à tout navigateur / système d'exploitation

  ⇒ BOM sert d'interface standard entre le navigateur et javascript

Document Object Model (DOM)



Browser Object Model (BOM)



Javascript : Objets du Navigateur

- Lors de l'ouverture d'une page Web, le navigateur crée différents objets. Les principaux sont :
  - window : fenêtre d'affichage de la page (contient des propriétés sur la fenêtre mais aussi les objets de la page chargée)
  - navigator : qui contient des informations sur le navigateur
  - location : contient des informations relatives à l'adresse de la page à l'écran
  - history: historique de navigation (liste de liens visités précédemment)
  - document : contient les propriétés sur le contenu de la page Web chargée

Javascript : Objet Window

- Objet Window : Fenêtre active du navigateur
- Parent de tous les autres objets
  - window.navigator
  - window.location
  - window.document
  - window.history
  - window.frames
  - ...
- Méthodes relatives à window
  - méthodes pour l'ouverture de boîtes de dialogue alert(), confirm() et prompt()
  - méthodes pour l'ouverture ou la fermeture de fenêtres open() et close()

    ⇒ *open(x,[y]) permet de charger dans une fenêtre une page d'url x.*
    Si y absent, on charge dans une nouvelle fenêtre, sinon on charge dans la fenêtre / onglet y (une nouvelle si pas encore de fenêtre nommée y)

Javascript : Objet Navigator

- Objet Navigator : Permet d'avoir des informations sur le navigateur utilisé
  - Infos générales sur le navigateur : *navigator.userAgent;*
  - Nom du navigateur : *navigator.appName;*
  - Version du navigateur : *navigator.appVersion;*
  - Méthodes de test sur le nom du navigateur : *isIE(), isFirefox, isSafari(), ...*
  - Système d'exploitation du poste client : *navigator.platform;*
  - Langue utilisée par le navigateur : *naviigator.language;*
  - Types de données supportées par le navigateur : *navigator.mimeTypes;*
  - Liste de plugins installés : *navigator.plugins;*
  - Méthode pour savoir si le Java est autorisé : *navigator.javaEnabled();*
  - Pour savoir si les cookies sont autorisés : *navigator.cookiesEnabled;*

Javascript : Objet History

- Objet History : Contient des infos sur l'historique
  - La propriété *history.length* permet de connaître le nombre d'objets dans l'historique
  - La méthode *history.back()* permet d'aller à l'URL précédent dans l'historique
  - La méthode *history.forward()* permet d'aller à l'URL suivant dans l'historique
  - La méthode *history.go(variable)* permet d'aller à un des URL de l'historique (variable peut être un index de page ou une chaîne proche de la page désirée).

Javascript : Objet Location

- Objet Location : Contient les éléments de l'URL du document
  - *location.protocol* : contient la partie protocole de l'url (le plus souvent "http:")
  - *location.port* : Numéro du port utilisé (serveur web : port 80)
  - *location.hostname* : Domaine de la page
  - *location.host* : "domaine:port"
  - *location.pathname* : Répertoire et nom de fichier de la page
  - *location.hash* : Ancre de l'url ("presentation" dans "www.lip6.fr#presentation")
  - *location.search* : Liste des variables et leurs valeurs (par exemple *"?nom=valeur&nom2=valeur2"*)
  - *location.href* : url complète (on peut changer cette propriété pour changer de page)
  - *location.reload()* : pour recharger la page courante

Javascript : Objet Document

- Objet Document : Contenu de la page
- Propriétés
  - alinkColor : couleur des liens lorsqu'ils sont cliqués
  - bgColor : couleur d'arrière plan
  - charset : jeu de caractères utilisés
  - cookie : chaîne de caractères pouvant être sauvegardée chez l'utilisateur
  - fgColor : couleur du texte
  - lastModified : date de dernière modification
  - linkColor : couleur des liens
  - referrer : pages déjà visitées
  - title : titre de la page

Javascript: Cookies

- Un cookie est une paire attribut/valeur
  - Stocké dans le navigateur du client
  - Possède éventuellement une date péremption
  - Contient des informations envoyées par un serveur à un client HTTP

  → Envoyé par le client en entête de chaque communication HTTP avec le serveur qui l'a créé
- Sert à enregistrer chez le client
  - Des données d'authentification
  - L'identifiant de la session en cours sur le serveur
  - Des préférences de l'utilisateur
  - Le contenu d'un panier d'achat électronique
  - ...

Gestion de cookies

```
function createCookie(name,value,days) {
        if (days) {
           var date = new Date();
           date.setTime(date.getTime()+(days*24*60*60*1000));
           var expires = "; expires="+date.toGMTString();
        }
        else var expires = "";
        document.cookie = name+"="+value+expires+"; path=/";
}

function readCookie(name) {
        var nameEQ = name + "=";
        var ca = document.cookie.split(';');
        for(var i=0;i < ca.length;i++) {
           var c = ca[i];
           while (c.charAt(0)==' ') c = c.substring(1,c.length);
           if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
        }
        return null;
}

function eraseCookie(name) {
        createCookie(name,"",-1);
}

createCookie("a",1,10); // crée un cookie a=1 expirant dans 10 jours
var val=readCookie("a");  // lit un cookie nommé a
```

Javascript : Objet Document

- Méthodes
  - createAttribute(nom_attribut) : Crée un noeud attribut
  - createElement(nom_balise) : Crée un noeud élément
  - createTextNode(texte) : Crée un noeud de texte
  - getElementById(id) : Retourne le noeud correspondant à l'identifiant passé en paramètre
  - getElementsByName(nom) : Retourne un tableau contenant les éléments possédant le nom passé en paramètre
  - getElementsByTagName(nom_balise) : Retourne un tableau contenant les éléments du type passé en paramètre

⇒ Attention : manipuler arbre DOM suppose d'attendre la fin du chargement de la page. Les traitements devant être faits lors de de l'affichage doivent être appelés par le gestionnaire d'évènement document.onload.

- Node = Noeud de l'arbre DOM
  - Différents types de noeud

Numéro	Type de nœuds
1	Nœud élément
2	Nœud attribut
3	Nœud texte
4	Nœud pour passage CDATA
5	Nœud pour référence d'entité
6	Nœud pour entité
7	Nœud pour instruction de traitement
8	Nœud pour commentaire
9	Nœud document
10	Nœud type de document
11	Nœud de fragment de document
12	Nœud pour notation

Javascript : Node

- Node = Noeud de l'arbre DOM
- Propriétés
  - nodeType : numéro du type du noeud
  - nodeValue : contenu du noeud
  - nodeName : nom du noeud
  - attributes : tableau de noeuds attribut associé
  - firstChild : premier noeud enfant
  - lastChild : dernier noeud enfant
  - nextSibling : noeud suivant dans l'arborescence
  - parentNode : noeud parent
  - previousSibling : noeud précédent dans l'arborescence

- Méthodes
  - appendData(texte) : concatène du texte en fin de la valeur d'un noeud texte ou attribut
  - insertData(pos,texte) : insère du texte à la position pos dans la valeur d'un noeud texte ou attribut
  - replaceData(pos,nb,texte) : remplace du texte d'un noeud texte ou d'un noeud attribut (pos correspond à la position dans la chaîne et nb le nombre de caractères à supprimer)
  - deleteData(pos,nb) : efface du texte d'un noeud texte ou d'un noeud attribut

- Méthodes
  - getAttribute(nom) : retourne la valeur d'un noeud attribut
  - getAttributeNode(nom) : retourne un noeud attribut
  - setAttribute(nom,valeur) : fixe la valeur d'un noeud attribut
  - setAttributeNode(node) : ajoute un noeud attribut
  - removeAttribute(nom) : efface la valeur d'un noeud attribut
  - removeAttributeNode(node) : supprime un noeud attribut

- Méthodes
  - appendChild(node) : ajoute un noeud enfant (en tant que dernier enfant)
  - removeChild(node) : retire un noeud passé en paramètre de la liste des fils du noeud
  - replaceChild(new,old) : remplace un noeud enfant old par un noeud new dans la liste des fils du noeud
  - hasChildNodes() : vérifie l'existence de noeuds enfants
  - insertBefore(new,avant) : insère un noeud enfant avant un autre noeud enfant dans la liste de fils du noeud (new = nouveau noeud, avant = noeud enfant avant lequel insérer)
  - cloneNode(booleen) : copie un noeud (si le booléen passé en paramètre est true, alors on copie aussi les fils du noeud)

Manipulation arbre DOM

```
// Crée un noeud titre et insertion sur la page
var titre=document.createElement("h1");
document.body.appendChild(titre);
titre.appendChild(document.createTextNode("Titre"));
titre.setAttribute("style","text-align:center; color:blue");

// Deplacement du titre en haut de la page
document.body.insertBefore(titre,document.body.firstChild);

// Crée des noeuds de texte
var horodateur = document.createTextNode(document.lastModified);
 var texteprecedent = document.createTextNode("Date de la dernière mise à jour: ");

// Ajoute les nouveaux noeuds en fin du paragraphe d'identifiant "paragraphe"
 document.getElementsByTagName("p")[0].replaceChild(
        texteprecedent,document.getElementsByTagName("p")[0].firstChild);
 document.getElementById("horo").replaceChild(
        horodateur,documentgetElementById("horo").firstChild);
```

La librairie JQuery

Javascript : JQuery

- jQuery
  - Bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript et HTML
  - ⇒ Simplifier les commandes communes de JavaScript.
- La librairie JQuery inclus
  - Parcours et modification du DOM
  - Gestion d'évènements
  - Effets et animations
  - Manipulations des feuilles de style
  - ...

Chargement librairie JQuery

```
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
</html>
```

- La bibliothèque jQuery peut être appelée de trois manières différentes :

Via la fonction $(expression)

```
$("div.test").add("p.quote").addClass("blue").slideDown("slow");
```

Via le préfixe jQuery(expression)

```
jQuery("#div_a_effacer").slideUp("fast");
```

Via le préfixe de fonction $.

```
$.each([1,2,3], function() {
 document.write(this + 1);
});
```

- Fonction $() permet différentes choses selon les paramètres
  - Une fonction
    - ⇒ Lancer cette fonction lorsque la page est chargée par le navigateur (comme window.onload mais n'attend pas que tout soit complètement téléchargé)

Processus de chargement

1. Le fichier html est téléchargé.
2. Il est lu par le navigateur.
3. Le navigateur commence à télécharger les éléments auxquels le fichier html fait référence (images, css...).
4. Le code situé dans $(function(){}) est exécuté dès que possible.
5. Lorsque tout est téléchargé, le code situé dans window.onload = function(){} est exécuté.

- Fonction $() permet différentes choses selon les paramètres
  - Un élément, ou un tableau d'éléments DOM
    - ⇒ Pour créer un objet JQuery auquel on peut appliquer diverses fonctions

Création d'objet JQuery

```
var mon_element = document.getElementById("id_de_mon_element_html");
var mon_objet_jquery = $(mon_element);
```

- Fonction $() permet différentes choses selon les paramètres
  - Le mot-clé this
    - ⇒ Pour créer un objet JQuery contenant l'objet sur lequel la fonction dans laquelle on se trouve a été appelée

Création d'objet JQuery

```
var mon_element = document.getElementById("main");
var jquery1=$(mon_element);

mon_element.getJQuery=function(){return($(this));}
var jquery2 = mon_element.getJQuery();

// => jquery1 et jquery2 équivalents
```

- Fonction $() permet différentes choses selon les paramètres
  - Une expression en CSS
    - ⇒ Permet de sélectionner facilement des éléments
    - ⇒ Forme un objet JQuery regroupant les noeuds correspondant à l'expressions CSS

Sélection d'objets avec sélecteurs CSS

```
// Sélection des éléments ayant pour classe "ique",
// dans l'élément ayant pour id "fixe"
var mon_objet_jquery = $("#fixe .ique");

// Sélection du premier élément <p> de l'élément <div>.
var mon_objet_jquery = $("div p:first-child");

// Sélection des éléments <p>, enfants directs d'un élement <div>
// et ayant pour classe "ique".
var mon_objet_jquery = $("div > p.ique");
```

- Fonction $() permet différentes choses selon les paramètres
  - Un chemin XPath
    - ⇒ Permet de sélectionner facilement des éléments
    - ⇒ Forme un objet JQuery regroupant les noeuds correspondant aux chemins XPath

Sélection d'objets avec chemin XPath

```
// Sélection des éléments p enfants de body, lui-même enfant de html
$("/html/body//p");
```

Documentation XPath en JQuery :

http://docs.jquery.com/DOM/Traversing/Selectors#XPath_Selectors

- Fonction $() permet différentes choses selon les paramètres
  - Une expression avec sélecteurs spécifiques à JQuery
    - ⇒ Permet de sélectionner facilement des éléments
    - ⇒ Forme un objet JQuery regroupant les noeuds correspondant à l'expression JQuery

Sélection d'objets avec sélecteurs JQuery

```
// Sélection du deuxième élément p de la page
$("p:nth(1)");

// Sélection des div cachés
 $("div:hidden");

// Sélection des div contenant la chaine de caractères "test"
 $("div:contains('test')");
```

Documentation sélecteurs JQuery :

http://docs.jquery.com/DOM/Traversing/Selectors#Custom_Selectors

Javascript : JQuery

- Fonction $() permet de regrouper des noeuds correspondant à un sélecteur passé en paramètre
  - ⇒ Objet JQuery contenant des noeuds DOM
  - ⇒ Possibilités d'y appliquer des méthodes JQuery
- Attention :
  - Fonctions Node pas applicables directement
  - ⇒ Accès aux éléments de l'objet par la méthode JQuery get
    - get() retourne un tableau contenant les noeuds sélectionnés
    - get(index) retourne le noeud à l'index index

Exemple utilisation JQuery

```
// Mauvaise utilisation JQuery (génère une erreur)
$("#comment > p").firstChild.appendData("Posté par Joe le 14/02/2012");

// Accès au premier élément de l'objet et application méthode au noeud
$("#comment > p").get(0).firstChild.appendData("Posté par Joe le 14/02/2012");

// Équivalent à :
$("#comment > p").get()[0].firstChild.appendData("Posté par Joe le 14/02/2012");
```

- Objet JQuery
  - ⇒ Propriété length : nombre d'éléments contenus
  - ⇒ Éléments rangés entre index 0 et length-1
  - ⇒ Méthode slice(debut,fin) : retourne l'objet jquery contenant uniquement les objets d'un jquery initial situés entre l'index debut et fin-1

Exemple utilisation JQuery

```
// Application méthode à tous les éléments sélectionnés
var jquery=$("p");
for (i=0;i=jquery.length;i++){
        jquery.get(i).firstChild.appendData("Posté par Joe le 14/02/2012");
}
```

- Objet JQuery
  - ⇒ Méthode each(func) : applique la function func à tous les éléments de l'objet

Exemple utilisation JQuery

```
<span> Premier paragraphe </span>
<span> Deuxième paragraphe </span>
<span> Troisième paragraphe </span>

// Remplacement du texte des span par "Paragraphe 1", "Paragraphe 2", ...
$("span").each(function(index) {
  if (this.firstChild!=undefined){this.firstChild.nodeValue="Paragraphe "+index;}
}).get().join(',');


// Modification de la couleur des éléments "div"
$( "div" ).each(function( i ) {
    if ( this.style.color !== "blue" ) {
      this.style.color = "blue";
    } else {
      this.style.color = "";
    }
});
```

- Méthodes d'accès / modification de contenu
  - html() : retourne le code html du le premier élément de l'objet JQuery
  - html(code) : donne un code html à tous les éléments de l'objet
  - text() : retourne le texte contenu par le premier élément de l'objet (et ses descendants)
  - text(texte) : donne un texte à tous les éléments de l'objet
  - val() : retourne la valeur contenue par le premier élément de l'objet (et ses descendants)
  - val(valeur) : donne une valeur à tous les éléments de l'objet

- Méthodes de manipulation de l'arbre
  - before(x), after(x) : insère x avant ou après tous les éléments de l'objet (x peut être du code html, un élément DOM ou un objet JQuery)
  - insertBefore(x), insertAfter(x) : insère tous les éléments de l'objet JQuery avant ou après l'objet x (x peut être un objet DOM ou une expression JQuery permettant de sélectionner des objets)
  - append(x), prepend(x) : insère x à la fin ou au début de tous les éléments de l'objet
  - appendTo(x), prependTo(x) : insère tous les éléments de l'objet JQuery à la fin ou au début de l'objet x
  - replaceWith(x) : remplace tous les éléments de l'objet JQuery par x
  - replaceAll(x) : remplace x par les éléments de l'objet

Exemple utilisation JQuery

```
<div class='a'>
  <div class='b'>b</div>
</div>

$('.a').after($('.c'));
<div class='a'>
  <div class='b'>b</div>
</div>
<div class='c'>c</div>

$('.a').before($('.c'));
<div class='c'>c</div>
<div class='a'>
  <div class='b'>b</div>
</div>

$('.a').append($('.c'));
<div class='a'>
  <div class='b'>b</div>
  <div class='c'>c</div>
</div>
```

- Méthodes pour le CSS
  - css(nom) : retourne la valeur de la propriété css nom pour le premier élément de l'objet JQuery
  - css(nom,valeur) : donne une valeur à une propriété css pour tous les éléments de l'objet
  - height(), width() : retourne la hauteur et largeur calculées pour le premier élément de l'objet
  - height(hauteur), width(largeur) : donnent une hauteur et une largeur à tous les éléments de l'objet
  - offset() : retourne un objet contenant les coordonnées absolues du premier élément de l'objet
  - offset(coordinates) : donne les coordonnées de l'objet coordinates à tous les éléments de l'objet

Javascript : JQuery

- Events
  - click(fonc) : spécifie fonc comme fonction à lancer lors du clic d'un des éléments de l'objet JQuery
  - keypress(fonc) : spécifie fonc comme fonction à lancer lors de la pression d'une touche (avec le focus sur un des éléments de l'objet)
  - hover(fonc) : spécifie fonc comme fonction à lancer lors du passage au dessus d'un des éléments de l'objet
  - focus(fonc) : spécifie fonc comme fonction à lancer lors de l'obtention du focus par un des éléments de l'objet
  - resize(fonc) : spécifie fonc comme fonction à lancer lors du redimensionnement d'un des éléments de l'objet
  - ...

Exemple utilisation JQuery

```
$( "#target" ).click(function() {
alert( "Handler for .click() called." );
});
```

- Effets
  - show(duree,[fonc]), hide(duree,[fonc]) : affiche / cache petit à petit un élément en jouant sur la propriété display
    - Le paramètre duree permet de spécifier une vitesse à l'animation
    - Le paramètre fonc correspond à une fonction à appeler lorsque l'action est terminée
  - fadeIn(duree,[fonc]), fadeOut(duree,[fonc]) : affiche / cache petit à petit un élément en jouant sur la propriété opacity
    - Le paramètre duree permet de spécifier une vitesse à l'animation
    - Le paramètre fonc correspond à une fonction à appeler lorsque l'action est terminée
  - slideDown(duree,[fonc]), slideUp(duree,[fonc]) : affiche / cache petit à petit un élément en jouant sur la propriété height
    - Le paramètre duree permet de spécifier une vitesse à l'animation
    - Le paramètre fonc correspond à une fonction à appeler lorsque l'action est terminée

- animate(objectif, duree, [fonc]) : Produit une animation de transfert entre l'état courant et un état visé
  - Le paramètre objectif est un ensemble de {propriété:valeur}
    - ⇒ Uniquement sur les propriétés numériques
    - ⇒ Replacer tirets par majuscules (exemple margin-left ⇒ marginLeft)
  - Le paramètre duree donne la durée de l'action
  - Le paramètre fonc est une fonction à lancer une fois l'action terminée

Animation JQuery

```
<!DOCTYPE html>
<html>
<head>
  <style>div { margin:3px; width:40px; height:40px;
        position:absolute; left:0px; top:60px;
        background:green; border:solid; }  </style>
  <script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
<div id></div>
<script>
var div = $("div");
function runIt() {
  div.hide("fast"); div.show("slow");
  div.fadeOut(2000); div.fadeIn(2000);
  div.slideUp(1000); div.slideDown(1000);
  div.animate({left:'+=200'},2000); div.animate({marginTop:'+=200'},2000);
  div.animate({borderLeftWidth:'+=200'},2000);
  div.animate({height:'+=200',width:'+=200'},2000);
  div.animate({left:'-=200'},2000); div.animate({marginTop:'-=200'},2000);
  div.animate({borderLeftWidth:'-=200'},2000);
  div.animate({height:'-=200',width:'-=200'},2000,runIt);
}
runIt();
</script>
</body>
</html>
```

Documentation Javascript :
http://fr.selfhtml.org/javascript/index.htm

Documentation JQuery :
http://api.jquery.com/