Université Montpellier-II
IUT de Béziers
Département SRC

UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

Programmation Web avancée
jQuery et Ajax

Responsable : Chouki TIBERMACINE
Bureau : Direction des études
Tél. : 04 67 11 18 21
Mél. : Chouki.Tibermacine@iutbeziers.fr
Web : http://www.lirmm.fr/~tibermacin/ens/pwa/

2011-2012

Cours 2

Gestion des événements et animations avec jQuery

Plan du cours

- Gestion des évènements avec jQuery
- Animations avec jQuery

Plan du cours

- Gestion des évènements avec jQuery
- Animations avec jQuery

Associer un gestionnaire d'événements

- L'une des difficultés de l'utilisation de JavaScript est que IE propose une API pour la gestion des événements qui est différente des autres navigateurs :
  - Écrire un code différent selon le navigateur
- jQuery propose une API unifiée
- Pour associer un gestionnaire d'événements, il suffit d'invoquer la fonction jQuery appropriée en lui passant en argument la fonction qui jouera le rôle de gestionnaire d'événement

Exemple d'association d'un gestionnaire d'événements

- Changer la couleur de fond de tous les paragraphes lors du click :

```
$("p").click(function() {
    $(this).css("backgound-color","gray");
});
```

- De cette façon, ce gestionnaire d'événement est associé à toutes les balises <p> du document (mieux que de l'associer aux balises une par une)

Fonctions pour associer des gestionnaires d'événements

- Voilà la liste complète de ces fonctions :

blur()	focusin()	mousedown()	mouseup()
change()	focusout()	mouseenter()	resize()
click()	keydown()	mouseleave()	scroll()
dblclick()	keypress()	mousemove()	select()
error()	keyup()	mouseout()	submit()
focus()	load()	mouseover()	unload()

- mouseover et mouseout vs. mouseenter et mouseleave : mouseover et mouseout associent un handler à un élément mais aussi à tous les éléments qui se trouvent à l'intérieur de celui-ci (event bubbling)

Particularités de ces fonctions

- Les fonctions resize() et unload() s'appliquent à l'objet « window »
  - Exemple : `$(window).resize(...)`
- La fonction scroll() s'applique principalement à l'objet « window », mais elle peut s'appliquer à n'importe quel objet ayant des barres de défilement (ex, propriété CSS overflow = scroll ou auto)
- La fonction load() peut s'appliquer à « window », mais aussi à un objet image
- La fonction error() peut s'appliquer à un objet image pour gérer l'échec de chargement d'une image

Fonctions supplémentaires : hover() et toggle()

- La fonction hover() permet d'associer un gestionnaire d'événement pour les mouseenter et mouseleave
- Appeler hover(f,g) où f et g sont deux fonctions :
  - Appeler mouseenter(f) et mouseleave(g)
- Appler hover avec une seule fonction, c'est celle-ci qui est utilisée pour gérer les deux événements (mouseenter et mouseleave)
- La fonction toggle() permet d'associer un gestionnaire d'événement pour le click
- Appeler toggle(f,g,h) où f, g et h sont trois fonctions :
  - Appeler f lors du premier click, g lors du second, h lors du 3ème, ...

Associer un gestionnaire d'événement comme propriété

- La création d'objets jQuery peut se faire avec la fonction $() en lui passant en premier argument une chaine représentant la balise à créer, et en second argument un objet de propriétés

- Dans ce deuxième argument, on peut préciser comme propriété une fonction d'association de gestionnaire d'événement

- Exemple :

```
$("<img/>",{
   src : url_image,
   alt : desc_image,
   className : "image_transp",
   click : function(){$(this).css("opacity","50%);"}
});
```

Gestionnaires d'événements jQuery

- La fonction jouant le rôle de gestionnaire d'événement est définie généralement sans paramètres
- En réalité jQuery ajoute au moins un paramètre à celle-ci :
  - L'objet représentant l'événement (détaillé après)
- Cette fonction peut retourner une valeur
- Si elle retourne false, le comportement par défaut associé à l'événement n'est pas effectué (pas d'envoi d'un formulaire au serveur, par exemple)

L'objet événement jQuery

- Cet objet possède les mêmes propriétés sur tous les navigateurs
- Cet objet se rapproche beaucoup de la norme W3C
- jQuery ne fait pas la différence entre les objets liés aux événements de la souris des objets liés aux événements du clavier
- jQuery copie toutes les propriétés dans chaque objet. Si une propriété ne s'applique pas, sa valeur est égale à « undefined »

Propriétés de l'objet événement jQuery

- Voilà la liste complète de ces propriétés :

altKey	ctrlKey	newValue	screenX
attrChange	currentTarget	offsetX	screenY
attrName	detail	offsetY	shiftKey
bubbles	eventPhase	originalTarget	srcElement
button	fromElement	pageX	target
cancelable	keyCode	pageY	toElement
charCode	layerX	prevValue	view
clientX	layerY	relatedNode	wheelDelta
clientY	metaKey	relatedTarget	which

Fonctions de l'objet événement jQuery

- En plus des propriétés précédentes, l'objet d'événement jQuery fournit les fonctions suivantes :

```
preventDefault()
stopPropagation()
stopImmediatePropagation()
isDefaultPrevented()
isPropagationStopped()
isImmediatePropagationStopped()
```

Propriétés de cet objet propres à jQuery

- La majorité des propriétés et fonctions de cet objet est normalisée par le W3C
- La propriété currentTarget est normalement équivalente à this (objet auquel le gestionnaire d'événement a été associé)
- La propriété relatedTarget est l'autre élément impliqué dans un événement. Par exemple, lorsque le curseur de la souris entre dans la zone d'un élément (événement mouseover), l'élément qui vient d'être quitté par le curseur de la souris est disponible dans relatedTarget
- La propriété which précise quel bouton de la souris (0, 1, 2 ou 3) ou quelle touche du clavier a provoqué l'événement

Fonctions avancées pour l'association de gestionnaires d'événements

- La fonction bind() permet d'associer un gestionnaire d'événement à un élément HTML
- Cette fonction est invoquée implicitement par les autres fonctions
- Elle permet de mettre en place des dispositifs avancés pour la gestion des événements
- La forme la plus simple de l'appel à cette fonction :
  - Deux arguments : le type de l'événement et une fonction (gestionnaire de l'événement)
- Exemple :

```
$("p").bind("click",f);
```

La fonction bind()

- La fonction bind() peut recevoir 3 arguments :
  - Le premier argument est le type d'événement et le 3ème est la fonction
  - Le 2ème argument sert dans ce cas a précisé une valeur pour la propriété data de l'objet événement qui sera généré

- La fonction bind() peut recevoir comme premier argument une liste de types d'événement

- Exemple :

la fonction hover() suivante :

```
$("a").hover(f);
```

est équivalente à :

```
$("a").bind("mouseenter mouseleave",f);
```

La fonction bind() -suite-

- La fonction bind() peut recevoir comme premier argument un objet qui fait correspondre à certains types d'événements des fonctions gestionnaires d'événements

- Exemple :

la fonction hover() suivante :

```
$("a").hover(f,g);
```

est équivalente à :

```
$("a").bind({mouseenter:f , mouseleave:g});
```

- La fonction one() fait la même chose que bind(), sauf qu'elle dés-associe le gestionnaire d'événement lorsqu'il est appelé la première fois (gestionnaire d'événement utilisé une fois)

Dés-associer un gestionnaire d'événement

- La fonction unbind() permet de dés-associer un gestionnaire d'événement
- Cette méthode dés-associe les gestionnaires associés avec jQuery (bind() entre autres)
- Sans arguments, cette fonction dés-associe tous les gestionnaires (tous les types d'événements) sur l'objet jQuery sélectionné
- Exemple : Dés-associer tous les gestionnaires d'événements de tous les éléments

```
$("*").unbind();
```

La fonction unbind()

- Le type d'événement peut être précisé en argument

```
$("a").unbind("mouseover mouseout");
```

- Il est possible d'appeler unbind() avec deux arguments
  - type(s) d'événements
  - référence vers la fonction qui a été utilisée comme gestionnaire d'événement
- Il est possible de passer en argument un objet contenant comme propriétés des types d'événements ayant comme valeurs des réf. vers les fonctions gestionnaires d'év.
- Il est possible de passer en argument à cette fonction un objet év.

```
unbind(ev);
est équivalent à:
unbind(ev.type,ev.handler);
```

Déclencher des événements

- Les gestionnaires d'événements associés aux éléments HTML sont invoqués lorsque des événements sont déclenchés automatiquement par le navigateur (à partir d'actions effectuées par l'utilisateur)

- A l'intérieur d'un script, il est possible de déclencher manuellement des événements

- La solution la plus simple : appeler les fonctions qui servent à associer des gestionnaires d'événements, mais sans arguments : `click() ou mouseover()`

Déclencher un événement -suite-

- Exemple :

```
$("#monForm").submit();
```

- Dans ce cas, les gestionnaires d'événements submit associés à l'objet jQuery sélectionné sont appelés. Si au moins un renvoie false ou appelle preventDefault(), le formulaire n'est pas envoyé sinon, le formulaire est envoyé au serveur

- L'appel des gestionnaires d'événements se fait de façon synchrone : appels bloquants

- La fonction trigger() permet de déclencher n'importe quelle sorte d'événement

La fonction trigger()

- L'utilisation la plus simple de cette fonction :
  - Un seul argument : le type d'événement

```
$("#monForm").trigger("submit");
```

- Il est possible de passer un objet événement en paramètre de cette fonction (le seul argument)

- Il est possible de passer n'importe quel objet qui a une propriété type qui doit désigner un type d'événement

- Exemple : le gestionnaire du clique sur le bouton 1 déclenche le même événement sur le bouton 2

```
$("#bouton1").click(function(e) {
    $("#bouton2")trigger(e);
});
```

Événements personnalisés

- Il est possible d'associer aux éléments d'un document HTML des gestionnaires de n'importe quel sorte de type d'événements personnalisés
- Il s'agit de désigner ce nouveau type d'événements personnalisé par une chaîne de caractères
- Ensuite, nous pouvons associer des gestionnaires de ce type d'événement à n'importe quel élément grâce à la fonction bind()
- Il est également possible de déclencher ces événements avec la fonction trigger()

Événements dynamiques

- Dans jQuery, nous générons parfois du contenu dynamiquement
- Si des gestionnaires d'événements ont été associés à certains types d'éléments HTML avant que d'autres éléments du même type n'aient été générés, les gestionnaires d'événements ne sont pas pris en compte pour les nouveaux éléments
- Solution : utiliser la fonction delegate

```
$(document).delegate("a","mouseover",
                     "gestionnaireLien");
```

Plan du cours

- Gestion des évènements avec jQuery
- Animations avec jQuery

Animations avec JavaScript et apport de jQuery

- Afficher et masquer des éléments :
  - Modifier la valeur de la propriété CSS visibility (visible et hidden)
- Produire un effet d'affichage/masquage progressif :
  - Modifier progressivement (½ seconde) la valeur de la propriété CSS opacity
  - Programmer une exécution répétée et avec timeout
- jQuery définit des fonctions assez simples pour faire cela
- Des fonctions d'animation avancées sont fournies par des plugins jQuery (des modules externes qu'on peut ajouter à jQuery)

Animations jQuery et leur durée

- jQuery fournit un certain nombre de fonctions pour animer les éléments d'un document HTML : show(), slideUp(), fadeIn(), ...
- jQuery fournit aussi une fonction animate() pour créer des animations personnalisées plus complexes
- Ces fonctions reçoivent en paramètre la durée de l'animation : nombres en ms (200, 400 -par défaut-, ...) ou strings ("fast" : 200, "slow" : 600, ...)

```
$("#message").fadeIn("fast");
```

- Il est possible de définir de nouvelles durées d'animation :

```
jQuery.fx.speeds["medium-fast"] = 300;
jQuery.fx.speeds["medium-slow"] = 500;
```

Désactiver les animations

- Parfois, on veut désactiver les animations (lourdeur de l'interface graphique, ...)

- jQuery fournit le moyen de désactiver toutes les animations en fixant la valeur de la propriété : `jQuery.fx.off`

- Exemple : Désactiver les animations à la demande

```
$("#stopmoving").click(function() {
    jQuery.fx.off = true;
});
```

Propriétés des animations jQuery

- Les fonctions d'animation s'exécutent de façon asynchrone. L'exécution se poursuit après l'appel de la fonction en même temps que l'animation se produit

- Dans une fonction callback précisée comme second paramètre de la fonction d'animation, this référence l'élément sur lequel l'animation se fait :

```
// Lors du fadein un message s'affiche sur l'élément
$("#message").fadeIn("fast",function() {
    $(this).text("Hello World");
});
```

la fonction callback s'exécute à la fin de l'animation

Propriétés des animations jQuery -suite-

- Une file d'attente gère les animations parallèles sur un élément

- Exemple : un élément clignotant

```
$("#clignotant").fadeIn(200).fadeOut(200)
                .fadeIn(200).fadeOut(200)
                .fadeIn();
```

- Il est possible de passer en paramètre des fonctions d'animation un objet possédant les propriétés *duration* et *complete* :

```
$("#message").fadeIn({
   duration : "fast",
   complete : function() {$(this).text("Hello World");}
});
```

Animations fadeIn, fadeOut et fadeTo

- Les fonctions correspondantes permettent d'afficher ou masquer des éléments HTML en manipulant la propriété CSS opacity
- Les fonction fadeIn() et fadeOut() reçoivent en paramètre la durée (vitesse) de l'animation et éventuellement une fonction callback
- La fonction fadeTo() reçoit en paramètre la durée (vitesse) de l'animation et la valeur de l'opacité finale de l'élément animé. Une fonction callback peut être passé en (3ème) paramètre aussi
- La fonction fadeOut() cache un élément, mais laisse vide l'espace occupé par celui-ci

Animations show, hide et toggle

- Les fonctions correspondantes permettent d'afficher/masquer des éléments, mais en décalant les autres éléments
- Lorsque ces fonctions sont invoquées sans paramètres, elles affichent/masquent les éléments sans animation
- Ces fonctions peuvent recevoir en argument la durée (vitesse) de l'animation
- La fonction hide(), appelée avec une vitesse d'animation, réduit la taille (width et height) de l'élément jusqu'à 0 et en même temps réduit l'opacité jusqu'à 0 (la fonction show() fait l'inverse)
- La fonction toggle() fait hide() ou show() selon l'état de l'élément

Animations slideDown, slideUp et slideToggle

- Les fonctions correspondantes permettent d'afficher/masquer des éléments en augmentant/réduisant leur hauteur
- La fonction slideUp() masque un élément (ou plusieurs) en animant sa hauteur (height) vers 0 et en mettant la valeur de la propriété CSS display à none
- La fonction slideDown() fait l'inverse
- La fonction slideToggle() fait l'une ou l'autre des deux premières fonctions selon l'état de l'élément
- Chaque fonction peut recevoir une durée d'animation et une fonction callback

Animations personnalisées

- Il est possible de créer ses propres animations en utilisant la fonction animate()
- Il faudra lui préciser comme premier argument un objet contenant les propriétés CSS et leurs valeurs finales (objet des propriétés de l'animation)
- Exemple : slideUp

```
$("img").animate({height:0});
```

Animations personnalisées -suite-

- Le second argument optionnel de cette fonction est un objet appelé l'objet des options de l'animation
- Il contient la vitesse d'animation et une fonction callback entre autres
- Exemple :

```
$("#monElem").animate({
    opacity:.25,
    fontSize:10
},{
    duration:500,
    complete:function() {$(this).text("Au revoir");}
});
```

L'objet des propriétés de l'animation

- Seules les propriétés CSS numériques peuvent être définies ici
- Impossible d'animer la couleur, la font, ...
- L'unité par défaut est le pixel (sinon, utiliser une chaîne à la place d'un nombre et ajouter l'unité)
- Pour préciser des valeurs relatives, il faudra utiliser += et -=
- Exemple :

```
$("p").animate({"margin-left":"+=.5in",opacity:"-=.1"});
```

- Possibilité d'utiliser les valeurs `"hide"`, `"show"` et `"toggle"` pour reprendre les valeurs initiales

L'objet des options de l'animation

- Il est possible de définir dans cet objet les propriétés duration et complete, vues précédemment
- Une autre propriété possible, c'est step. Elle peut être initialisée avec une fonction qui sera appelée à chaque étape de l'animation
- Il est possible aussi d'utiliser la propriété queue. Cette propriété prend les valeurs true et false. Si elle est initialisée à false, l'animation est lancée sans attendre les autres animations
- Une dernière propriété possible, c'est easing. Elle permet de préciser le style de la transition (linear, swing -par défaut-, ...)

Arrêter ou retarder des animations

- La fonction stop() arrête les animations démarrées sur les objets sélectionnés

- Cette fonction peut recevoir deux paramètres :
  - Le premier est un booléen, qui, initialisé à true, efface les animations en attente
  - Le second paramètre précise quelles sont les propriétés CSS qu'il faudra laisser telles qu'elles ou à mettre à l'état final

- La fonction delay() permet de retarder une animation d'une certaine durée exprimée en millisec. ou une chaine passée en argument

- Exemple : Faire disparaître à moitié, attendre et glisser vers le haut

```
$("img").fadeTo(100,0.5).delay(200).slideUp();
```

Arrêter ou retarder des animations -suite-

- jQuery associe à chaque élément dans un document HTML (ou même les objets document et window) des files d'attente de fonctions d'animation à invoquer

- Pour ajouter une fonction à cette file, on peut utiliser queue() :

```
$("#message").queue(function(next) {
    // ...
    next(); // obligatoire : appeler la prochaine fct
});
```

- La fonction clearQueue() efface la file d'attente

Références bibliographiques

- Bear Bibeault et Yehuda Katz. « **jQuery in Action** ». Second Edition. Manning Publications Co. Juin 2010.
- David Flanagan. « ***jQuery – Pocket Reference*** ». First Edition. Éditions O'Reilly. Décembre 2010.
- Anthony T. Holdener III. « **Ajax: The Definitive Guide – Interactive Applications for the Web** ». Éditions O'Reilly. Janvier 2008.
- Documentation en ligne (très pratique) : **http://api.jquery.com/**

Questions