

JQUERY :

Sélecteur d'élément		Sélecteur d'élément	
\$('ul')	Les balises 	\$('a')	sélectionne tous les liens hypertextes ;
\$('ul.bleu')	La balise de classe bleu	\$('.rouge')	sélectionne les éléments de classe rouge ;
\$('div ul')	La balise contenue dans la balise <div>	\$('#e2')	sélectionne l'élément d'identifiant e2 ;
\$('div ul li[class="pair"]')	La balise contenue dans une balise , elle-même contenue dans une balise <div>, et dont l'attribut class vaut pair	\$('[src]')	sélectionne tous les éléments qui possèdent un attribut src ;
\$('li[class]')	Les balises qui possèdent un attribut class	\$('[width="40"]')	sélectionne tous les éléments qui ont un attribut width égal à 40.
\$('li[class="impair"]')	Les balises qui possèdent un attribut class de valeur impair	\$('*')	Toutes les balises du document

la fonction \$() est un objet jQuery		Sélecteurs d'attributs	
\$('a').length	retourne le nombre de liens hypertextes contenus dans la page. \$('a')[2] fait référence au troisième lien hypertexte contenu dans la page.	\$('[nom]')	un attribut donné, en utilisant le sélecteur
\$('ul.bleu').length	retourne le nombre de balises de classe bleu.	\$('[nom=valeur]')	un attribut donné qui a une certaine valeur
\$('li[class="impair"]').length	retourne le nombre de balises qui ont un attribut class de valeur impair.	\$('[border*="1"]')	Image dont l'attribut border contient partiellement ou totalement la valeur « 1 ».
\$('body').length	retourne « 1 » car le document contient une seule balise <body>.	\$('[title~="animal"]')	Image dont l'attribut title contient le mot « animal » délimité par une espace.
\$('a')[0]	retourne le premier lien hypertexte de la page.	\$('[src\$="e.jpg"]')	Image dont l'attribut src se termine par « e.jpg ».
\$('ul.bleu')[3]	retourne la quatrième balise de classe bleu.	\$('[border!="15"]')	Image qui ne possède pas un attribut border égal à « 15 ».
\$('body')[0]	est équivalent à document.body.	\$('[src^="ch"]')	Image dont l'attribut src commence par « ch ».

Le **chaînage** est un concept très puissant et pourtant simple à comprendre. Étant donné que les méthodes jQuery retournent un objet jQuery, il est possible de les « chaîner », c'est-à-dire de les exécuter les unes à la suite des autres. **\$('.rouge').css('background', 'red'); + \$('.rouge').css('color', 'yellow'); = \$('.rouge').css('background', 'red').css('color', 'yellow');**

sélecteurs hiérarchiques sélectionner les enfants d'un certain parent				pseudo-sélecteur		
('p > e')	Éléments e directement descendants d'éléments p	\$('li > ul')	Éléments ul directement descendants d'éléments li	:even	\$('p:even')	Éléments p pairs
('p + e')	Éléments e directement précédés d'un élément p	\$('li + li')	Éléments li directement précédés d'un élément li	:odd	\$('p:odd')	Éléments p impairs
('p ~ e')	Éléments e précédés d'un élément p			:eq()	\$('p:eq(3)')	Éléments p après le deuxième
:empty	Éléments qui n'ont pas d'enfant			:gt()	\$('p:gt(1)')	Élément p d'index 4
:first-child	Premier enfant	\$('li:first-child')	Premier élément li enfant	:lt()	\$('p:lt(3)')	Éléments p avant le quatrième
:first	Premier élément	\$('li:first')	Premier élément li	Sélecteurs d'élément particulier		
:last-child	Dernier enfant	\$('li:last-child')	Dernier élément li enfant	\$(':header').css('color', 'red');	Tous les titres<h1>à<h6>	
:last	Le dernier élément de la sélection	\$('li:last')	Dernier élément li	\$('div:hidden').show();	Éléments cachés	
:nth-child()	Élément qui est l'énème enfant de son parent	\$('li:nth-child(2)')	Deuxième enfant li	:visible	Éléments visibles	
:only-child	Éléments qui sont enfants uniques de leur parent	\$('li:only-child')	Éléments li enfants uniques de leur parent	\$(':header:not(h1)').hide();	Toutes les balises de titre sont cachées, à l'exception de	

Pseudo-sélecteurs css pour les formulaires			
:input	Tous les éléments de typeinput,textarea,selectetbutton	:submit	Éléments de typesubmit
:button	Éléments de typebutton	:text	Éléments de typetext
:checkbox	Éléments de typecheckbox	:password	Éléments de typepassword
:checked	Éléments qui sont cochés	:selected	Éléments sélectionnés
:radio	Éléments de typeradio	:focus	Sélectionne l'élément s'il a le focus
:reset	Éléments de typereset	:enabled	Éléments validés
:image	Tous les boutons de typeimage		

sélecteurs spécifiques pour les tableaux		Accéder aux attributs des balises HTML	
Pour le tableau suivant Nous allons affecter la couleur jaune à l'arrière-plan de certaines cellules du tableau en utilisant la méthode jQuerycss().		<i>getters</i> =signifie « obtenir » / <i>setters</i> signifie « définir »	
\$('td:first')	Coloration de la première cellule	\$('#plus').attr('src');	retourne l'attribut src de l'élément d'identifiant plus.
\$('tr:last')	Coloration de la dernière ligne	\$('div').attr('class');	retourne l'attribut class du premier <div>.
\$(':empty')	Coloration des cellules vides	\$('div').attr('class', 'madiv');	modifie ou crée l'attribut class dans les balises <div> du document et leur affecte la valeur « madiv ».
\$('tr:even')	Coloration des lignes paires	\$('#illustration').attr('src', 'monimage.jpg');	modifie ou crée l'attribut src dans la balise d'identifiant illustration et lui affecte la valeur « monimage.jpg ».
\$('td:even')	Coloration des cellules paires	\$('a').removeAttr('href');	supprimer l'attribut href de toutes les balises <a> du document
\$('tr:gt(1)')	Coloration des lignes d'index supérieur à 2	\$('.para').css('font-size', '40px');	affecte la valeur « 40px » à l'attribut font-size de tous les éléments de classe para
\$('tr:gt(0):lt(4)')	Coloration des lignes d'index supérieur à 1, et, dans le sous-ensemble correspondant, des lignes d'index inférieur à 5	Passe en revue les éléments sélectionnés et permet de les modifier en utilisant des instructions JavaScript	
\$('td:gt(4):lt(10)')	Coloration des cellules d'index supérieur à 5, et, dans le sous-ensemble correspondant, des cellules d'index inférieur à 11	<pre>\$('sel').each(function(index){ //Une ou plusieurs instructions JavaScript });</pre>	
\$('tr:not(tr:last)')	Coloration de toutes les lignes à l'exception de la dernière		

Tester / modifier la valeur des zones de texte Formulaire		Travailler avec les valeurs stockées dans les éléments	
\$('#nom').val()	Lit le nom de l'utilisateur.	\$('p').text();	Pour récupérer le texte stocké dans les deux paragraphes
\$('#pass').val()	Lit le mot de passe.	alert(\$('p').text());	pour afficher ce texte sur l'écran
\$(':radio#H:checked').val()	Lit l'état du bouton radio H. Renvoie true si le bouton est sélectionné, sinon false.	var premier = \$('p:first').text();	Première valeur de p
\$('#fonction').val()	Lit l'élément sélectionné dans la liste déroulante.	var dernier = \$('p:last').text();	Dernière valeur de p
\$('#nom').val('Michel')	Écrit « Michel » dans la zone de texte Nom d'utilisateur.	alert(\$('p').html());	Affiche le code HTML du premier élément (voir image suivante).
\$('#pass').val('abcde')	Écrit « abcde » dans la zone de texte Mot de passe.	alert(\$('p:first').html());	Affiche le code HTML du premier élément.
\$(':radio').val(['H']);	Sélectionne le bouton radio H.	alert(\$('p:last').html());	Affiche le code HTML du dernier élément.
\$('#fonction').val('retraite')	Sélectionne Retraité dans la liste déroulante.	<pre>\$('p').html(function(index, actuel){ { alert('Paragraphe ' + (index+1) + ' : '+actuel);} });</pre>	Affiche individuellement le code HTML de chaque élément

Attribut class			
addClass()	ajoute une classe dans les éléments sélectionnés ;	Ex:	\$('#julia').addClass('rouge');
removeClass()	supprime (si elle existe) une classe des éléments sélectionnés ;	Ex:	\$('#eric').removeClass('vert') ;
toggleClass()	accomplit deux actions : si la classe spécifiée n'existe pas dans les éléments sélectionnés, elle y est ajoutée. Si elle existe, elle est supprimée.	Ex:	\$('#paul').toggleClass('vert')
hasClass()	permet de tester si la sélection est d'une certaine classe.		
Associer des données au balises			
\$.data()	permet d'associer une ou plusieurs données textuelles à n'importe quel élément du DOM	var uneVariable = \$.data(el,'nom').nom_don;	uneVariable est une variable quelconque ;
\$.data()	en getter		El est le nom de l'élément (sans apostrophes) auquel une donnée a été associée
\$.data(el, 'nom', nom_don: don);	El est le nom de l'élément concerné, sans apostrophes ;		nom est le nom (entre apostrophes) dans lequel a été stockée la donnée ;
	Nom est le nom (entre apostrophes) de la « variable » dans laquelle sera stockée la donnée ;		nom_don est le nom de la donnée à retrouver, sans apostrophes.
	nom_don est le nom (sans apostrophes) associé à la donnée ;		
	Don est une donnée quelconque. S'il s'agit d'un nombre, les apostrophes sont inutiles. Par contre, s'il s'agit d'une chaîne, mettez-la entre apostrophes.	\$.removeData(el,'nom');	supprimer les données associées à un élément sans supprimer l'élément et sans changer de page

Insérer et remplacer des éléments dan le DOM			
Insérer du contenu dans ou en dehors de la sélection par un sélecteur jQuery		Insérer des éléments dans l'arborescence du DOM	
\$('#h2').append(' ***');	Ajout d'un espace et de trois astérisques après chaque titre<h2>	(Élément à insérer).appendTo(cible)	insère un élément à la fin de la cible ;
\$('#h2').prepend('*** ');	Ajout d'un espace et de trois astérisques avant le titre <h2>	(Élément à insérer).prependTo(cible)	insère un élément au début de la cible ;
\$('#trois').before('<hr>');	Ajout d'une ligne de séparation horizontale avant le titre <h2>	(Élément à insérer).insertBefore(cible)	insère un élément avant la cible ;
\$('#hr').after(' ');	Insertion de deux sauts de ligne apres chaque balises<hr>	(Élément à insérer).insertAfter(cible)	insère un élément après la cible.
\$('#h2').replaceWith('<h3>');	Remplacer tous les titres<h2>du document par des titres<h3>?	Dupliquer un élément existant	
Pour déplacer un élément existant dans le document		\$('#deux').clone().insertBefore(\$('#un'));	Nous allons dupliquer le paragraphe Id #deux et l'insérer avant le paragraphe Id #un
\$('#selection').append(élément a deplacer);	le déplacement se fait avant la balise de fin de l'élément sélectionné	Entourer un élément par un ou plusieurs autres éléments créés à la volée	
\$('#sel').prepend(depl);	le déplacement se fait après la balise de début de l'élément sélectionné	\$('#li').wrap('<i></i>');	la liste à puces en italique. Pour cela, il vous suffit de les entourer avec la balise <i> en utilisant la méthode wrap()
\$('#sel').before(depl);	elle se fait avant cette balise la balise de début de l'élément sélectionné	\$('#p').wrapAll('<div></div>');	Les paragraphes ont été rassemblés et entourés par une balise <div>
\$('#sel').after(depl);	elle se fait après la balise de fin de l'élément sélectionné	\$('#p').wrapInner('<div></div>');	<p><div>...</div></p>
\$('#h2').remove();	Supprimer tous les titres <h2>du document		

Les bases de la gestion événementielle : Avant tout écrire **\$(selection).methode(function() { instruction pour gérer événement }**

la souris		Le clavier	
click()	Clic gauche	keydown()	Appui sur une touche du clavier
dblclick()	double-clic	keyup()	Relâchement d'une touche du clavier préalablement enfoncée
mousedown()	Appui sur le bouton gauche ou droit de la souris alors que le pointeur est au-dessus de l'élément	keypress()	Maintien d'une touche du clavier enfoncée
mouseenter()ou mouseover()	Début de survol de l'élément	event.which	savoir quelle touche du clavier a été pressée
mouseleave()ou mouseout()	Arrêt de survol de l'élément	event.type	connaître le type d'événement qui a été levé
mousemove()	Déplacement du pointeur au-dessus de l'élément	Les pages en chargement	
mouseup()	Relâchement du bouton gauche ou droit alors que le pointeur est au-dessus de l'élément	Window.addEventListener('load',monscript) ; function.monscript{	Javascript, attendre le chargement de la page pour lancer le script
scroll()	Utilisation de la roulette alors que le pointeur se trouve, au dessus d'un élément concerné par ce type d'événement	\$('document').ready(function(){}) ; ou \$(function(){}) ;	Jquery, attendre le chargement de la page pour lancer le script
event.which	il peut être nécessaire de savoir quel bouton de la souris a été pressé		

les méthodes événementielles en rapport avec le gain et la perte de focus, la modification de la taille et du contenu, et la sélection d'un élément			
focus()	Réception de focus par l'élément	trigger()	Va déclencher l'événement spécifier en argument ainsi que le comportement par défaut de l'événement
blur()	Perte de focus par l'élément	triggerHandler()	Ne va pas déclencher le comportement par défaut de l'événement
focusin()	Réception de focus par l'élément ou un de ses enfants	Effets et animations	
focusout()	Perte de focus par l'élément ou un de ses enfants	.hide(nb)	Disparition....nb est le nombre de ms
resize()	Redimensionnement d'un élément	Slow:600ms Normal:400ms Fast:200ms	
change()	Modification d'un élément	.show(nb)	Apparition....nb est le nombre de ms
.on	Va nous permettre de gérer tous les événements \$('div1').on({ mouseenter: function(){ \$('#msg').text("Vous êtes entré dans le div");}, mouseleave: function(){ \$('#msg').text("Vous êtes sorti du div");}}	.fadeIn()	Fait disparaître progressivement
		.fadeOut()	Fait apparaître progressivement
		.fadeTo()	Gère l'opacité
.off	Va nous permettre de supprimer des gestionnaires d'événements attaché a la méthode on() \$('#b1').on('click', function(){..... \$('#b1').off('click');	.FadeToggle()	Soit l'un soit l'autre
		.slideUp()	Disparition en mouvement vers le haut
		.slideDown	Disparition en mouvement vers le Bas
		.slideToggle	Disparition en mouvement vers le Bas ou le haut