

Documentation en ligne :

- <http://jquery.com/>
- <http://api.jquery.com/>
- <http://jqueryui.com/>

Exercice 1 : Les sélecteurs

Référence : <http://api.jquery.com/category/selectors/>

Pour chaque expression ci-dessous, donnez le sélecteur permettant de sélectionner le(s) élément(s) demandé(s).

Testez votre sélecteur dans une page HTML pour vérifier son exactitude. Vous pouvez, par exemple, utiliser la méthode `hide()` pour masquer le(s) élément(s) sélectionné(s) par le sélecteur.

1. Tous les paragraphes sur lesquels la classe CSS « `css_paragraphe` » est appliquée.
2. L'élément DIV dont l'identifiant (attribut `id`) est « `id_div` ».
3. Toutes les cellules appartenant au tableau ayant l'identifiant « `id_tableau` ».
4. Toutes les lignes appartenant au tableau ayant l'identifiant « `id_tableau` » et ne possédant qu'une seule cellule.
5. Tous les paragraphes sur lesquels la classe CSS « `css_paragraphe` » est appliquée et toutes les lignes paires du tableau ayant l'identifiant « `id_tableau` ».
6. Toutes les cellules vides (i.e. n'ayant pas d'élément fils) appartenant au tableau ayant l'identifiant « `id_tableau` ».
7. Tous les éléments INPUT ayant un identifiant et dont le nom (attribut « `name` ») contient la chaîne de caractères « `form_texte` ».
8. Le deuxième formulaire (élément FORM) de la page HTML.
9. Tous les éléments parents des paragraphes sur lesquels la classe CSS « `css_paragraphe` » est appliquée.
10. Tous les éléments frères DIV (i.e. ayant le même parent) dont la largeur est de 150 pixels situés après le tableau ayant l'identifiant « `id_tableau` ».

Exercice 2 : La manipulation des éléments

Référence : <http://api.jquery.com/category/manipulation/>

Dans cet exercice, vous allez créer une barre chromatique (i.e. un dégradé de couleur) par programmation. La barre sera composée de 10 cases à l'horizontale allant du bleu clair au bleu très foncé.

Le corps (élément BODY) de votre page HTML ne devra contenir que le code défini ci-dessous :

```
<html>
  <head>
    <!-- Mettre ici le code approprié //-->
  </head>
  <body>
    <table id="barre_chromatique" />
  </body>
</html>
```

Voici le résultat attendu dans le navigateur Web :



Vous construirez entièrement la barre chromatique à l'aide des fonctions de manipulation des éléments en modifiant l'élément TABLE ayant l'identifiant « barre_chromatique ».

Exercice 3 : La gestion des évènements

Référence : <http://api.jquery.com/category/events/>

Dans cet exercice, vous allez créer un champ de saisie spécifique aux montants. Vous utiliserez pour cela un élément INPUT de type « text » (attribut « type »). Vous ferez en sorte qu'il ne soit possible de saisir que des chiffres ou des virgules.

Pour identifier les éléments INPUT sur lesquels ce filtre de saisie sera appliqué, vous utiliserez une classe CSS comme critère discriminant. Par exemple, la classe « montant » :

```
<input type="text" class="montant" />
```

Pourquoi ne pas utiliser un identifiant (attribut « id ») comme critère discriminant ?

Exercice 4 : Les effets visuels

Référence : <http://api.jquery.com/category/effects/>

Dans cet exercice, vous allez créer une horloge affichant les heures, les minutes et les secondes. Votre horloge se mettra à jour en temps réel.

Voici un exemple de résultat attendu dans le navigateur Web :



Vous utiliserez trois effets de transition différents pour le passage des heures, des minutes et des secondes.

Exercice 5 : La validation de formulaire

Référence : <http://docs.jquery.com/Plugins/Validation>

Téléchargement : <http://bassistance.de/jquery-plugins/jquery-plugin-validation/>

Dans cet exercice, vous allez créer un formulaire de saisie. Votre formulaire comportera les champs suivants :

Libellé	Type	Taille maxi.	Obligatoire	Description
Civilité	Radio-bouton	-	Oui	Liste de valeurs : Monsieur, Madame, Mademoiselle
Nom	Texte	50	Oui	-
Prénom	Texte	150	Non	-
Date de naissance	Texte	-	Oui	Format à contrôler : JJ/MM/AAAA
E-mail	Texte	-	Non	Format à contrôler
Téléphone	Texte	-	Oui	Format à contrôler : numéro de téléphone français
Valider	Bouton	-	-	-

Le clic sur le bouton « Valider » doit contrôler que tous les champs sont correctement saisis.

Voici un exemple de résultat attendu dans le navigateur Web :

Formulaire de saisie

Civilité * ☐ Monsieur ☐ Madame ☐ Mademoiselle

Nom *

Prénom

Date de naissance *

E-mail

Téléphone *

Exercice 6 : L'affichage d'un carrousel

Référence : <https://github.com/jafin/jcarousellite/tree/master/src>

De plus en plus de sites marchands utilisent un carrousel pour afficher leurs produits. Le carrousel se traduit généralement par une barre horizontale d'images défilant en boucle.

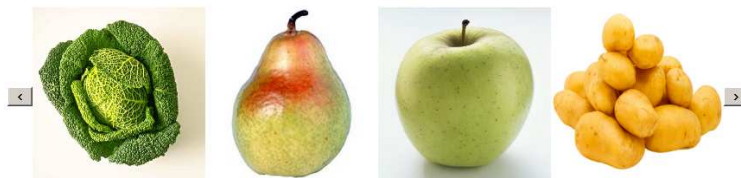
Dans cet exercice, vous allez créer un carrousel horizontal de 4 vignettes visibles.

Le défilement des vignettes dans le carrousel se fera manuellement par deux boutons « Précédent » et « Suivant ».

Le clic de souris sur une des vignettes du carrousel devra afficher le nom de l'objet sélectionné dans une zone située en dessous du carrousel.

Voici un exemple de résultat attendu dans le navigateur Web :

Fruits et légumes



Exercice 7 : La sélection de dates à l'aide d'un calendrier

Référence : <http://jqueryui.com/datepicker/>

Dans cet exercice, vous allez créer un formulaire pour saisir des tâches. Une tâche est définie par une date de début, une date de fin et une description. Le clic sur le bouton « Ajouter » ajoute la nouvelle tâche à la liste des tâches, et efface les champs du formulaire.

Vous ne contrôlerez pas la saisie du formulaire. Cet exercice a pour objectif de vous entraîner à utiliser le widget Calendrier de jQuery.

Voici un exemple de résultat attendu dans le navigateur Web :

Nouvelle tâche

Date de début

Date de fin

Description

Ajouter une nouvelle tâche

Tâches à faire :

- 07/01/2011 - 10/01/2011 : Week-end à la montagne !
- 15/01/2011 - 16/01/2011 : Cours de piano.

Vous paramètrerez le calendrier de telle sorte que les dates soient saisies au format JJ/MM/AAAA et qu'il soit possible d'accéder directement à un mois en particulier.

Enfin, vous configurerez le calendrier pour qu'il ne soit possible de saisir une date uniquement sur une période d'une année, et cela à partir de la date du jour.

Exercice 8 : L'auto-complétion

Référence : <http://jqueryui.com/autocomplete/>

Dans cet exercice, vous allez créer un champ d'auto-complétion pour des noms de villes. Un champ d'auto-complétion est un champ de saisie qui propose des résultats en fonction du texte en cours de saisie dans le champ.

Votre champ d'auto-complétion devra autoriser la sélection multiple de villes. Vous utiliserez une liste statique de villes pour l'exercice.

Voici un exemple de résultat attendu dans le navigateur Web :



The screenshot shows a text input field with the label "Villes :". Below the input field, a dropdown menu is displayed, listing several French cities: Bourges, Orléans (highlighted), Tours, Nanterre, Paris, Versailles, Amiens, Clermont, and Mouy.

Exercice 9 : Les boîtes de dialogue

Référence : <http://jqueryui.com/dialog/>

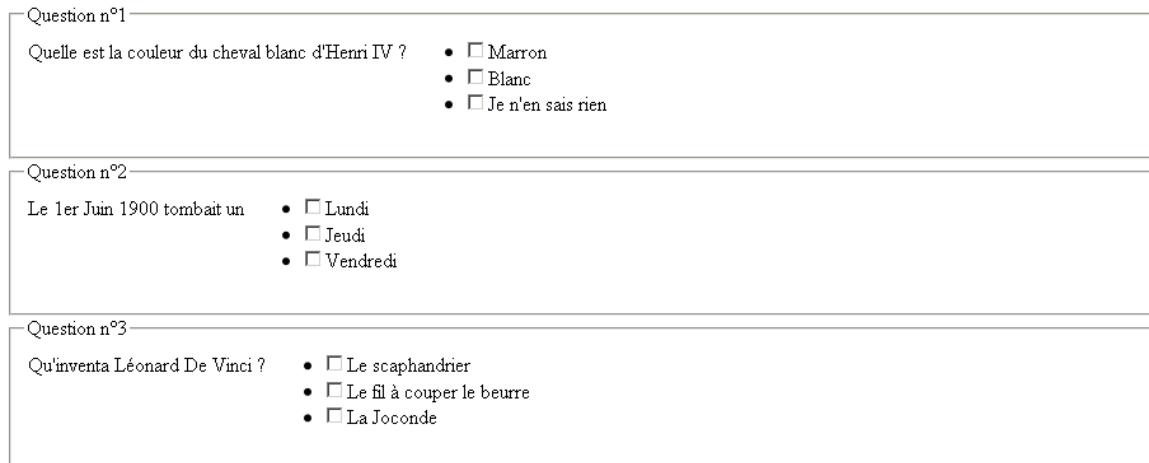
Dans cet exercice, vous allez créer un QCM de 3 questions. Vous utiliserez des cases à cocher (élément INPUT de type « checkbox ») pour définir les réponses possibles à vos questions.

Le clic sur le bouton « Valider » affichera une boîte de dialogue modale indiquant le nombre de réponses exactes.

Le clic sur le bouton « Fermer » de la boîte de dialogue effacera les champs du formulaire.

Voici un exemple de résultat attendu dans le navigateur Web :

QCM



The screenshot shows a form titled "QCM" with three questions, each in a separate box. Each question has a list of possible answers with checkboxes.

Question n°1
Quelle est la couleur du cheval blanc d'Henri IV ?

- ☐ Marron
- ☐ Blanc
- ☐ Je n'en sais rien

Question n°2
Le 1er Juin 1900 tombait un

- ☐ Lundi
- ☐ Jeudi
- ☐ Vendredi

Question n°3
Qu'inventa Léonard De Vinci ?

- ☐ Le scaphandrier
- ☐ Le fil à couper le beurre
- ☐ La Joconde

Valider

Exercice 10 : Les requêtes AJAX

Référence : <http://api.jquery.com/category/ajax/>

Dans cet exercice, vous allez créer une page utilisant une requête Ajax pour simuler l'interrogation d'un service météorologique.

Votre page présentera un champ de saisie pour un code postal. Lorsque que celui-ci est renseigné, la page est automatiquement mise à jour, sans être rechargée, et indique la météo pour le code postal saisi.

Voici un exemple de résultat attendu dans le navigateur Web :

Code postal :
Le 30/01/2012 à 17:34:23 : Il pleut comme d'habitude ! (45000)

Exercice d'application : Jeu de plateau « Evil Monkeys vs Tomates Tueuses »

Il s'agit de réaliser un **jeu de plateau** dans une **seule page HTML** à l'aide de **jQuery**.

Règle du jeu :

La grille de jeu est représentée par une grille de 7 cases en hauteur et de 10 cases en largeur. Le jeu se joue à 2 joueurs humains (aucune intelligence artificielle). Chaque joueur représente une équipe. Deux équipes s'affrontent : l'équipe des « Evil Monkeys » contre l'équipe des « Tomates Tueuses ».

Chaque équipe doit protéger le totem qui a été dérobé à l'équipe adverse. Les tomates tueuses protègent le totem « Banane Nucléaire ». Les evil monkeys protègent le totem « Engrais Hallucinogène ».

Le but du jeu est de récupérer le totem qui a été dérobé. La première équipe qui récupère son totem a gagné. Les evil monkeys doivent reprendre le totem « Banane Nucléaire ». Les tomates tueuses doivent reprendre le totem « Engrais Hallucinogène ».

L'équipe qui débute la partie est désignée de façon aléatoire par le programme.

Mode de déplacement :

Chaque équipe est composée de 5 pions et du totem qu'il protège.
A tour de rôle, chaque joueur déplace un de ses pions ou le totem qu'il protège.

Les pions et les totems peuvent être déplacés dans tous les sens (horizontal, vertical ou diagonal) sur la grille de jeu. Les pions ne peuvent se déplacer que d'une seule case à la fois. Les totems peuvent être déplacés d'une ou de deux cases à la fois.

En revanche, le déplacement des totems est limité. Un joueur ne peut déplacer que 3 fois le totem qu'il protège au cours d'une partie.

Les pions peuvent être déplacés sur une case vide ou sur une case occupée par un pion de l'adversaire. Dans ce dernier cas, le pion de l'adversaire se transforme en pion du joueur et prend l'ancienne place du pion du joueur.

Les totems ne peuvent être déplacés que sur une case vide.

Pour reprendre le totem dérobé, le joueur doit déplacer un de ses pions sur la même case que le totem. La partie est alors terminée.

Position de départ des pions et des totems :

	TT							EM	
	TT							EM	
BN	TT							EM	EH
	TT							EM	
	TT							EM	

Légende :

- TT : Pion « Tomate Tueuse »
- EM : Pion « Evil Monkey »
- BN : Totem « Banane Nucléaire »
- EH : Totem « Engrais Hallucinogène »

Déplacement d'un pion :

	TT						TT	EH	
	TT					TT	EM		
BN								EM	
	TT					EM		EM	
								EM	

Légende :

- Case « Bleue » : Pion à déplacer
- Case « Verte » : Case sur laquelle le pion à déplacer peut se positionner.
- Case « Rouge » : Case sur laquelle le pion à déplacer ne peut pas se positionner.

Déplacement du totem :

			TT				TT	EH	
			TT			TT	EM		
		BN		EM				EM	
			TT					EM	
								EM	

Conversion d'un pion adverse en un pion de son équipe :

			TT				TT	EH	
			TT			TT	EM		
		BN		EM				EM	
			TT					EM	
								EM	

			TT				TT	EH	
			TT			TT	TT		
		BN		EM				EM	
			TT					EM	
								EM	

Légende :

- Case « Violette » : Le pion convertit prend l'ancienne place du pion à déplacer. L'équipe du joueur gagne ainsi un nouveau pion. L'équipe adverse perd un pion.

Récupération du totem dérobé (fin de partie) :

			TT				TT	EH	
			TT			TT	EM		
		BN		EM				EM	
			TT					EM	
								EM	

Travail à réaliser :

Cet exercice est à réaliser par groupe de 2 ou 3 étudiants **maximum**. Il sera noté et constituera la note globale d'évaluation du TD jQuery. Le projet doit être réalisé dans **une seule page HTML** à l'aide de **jQuery**.

A l'instar des projets informatiques réalisés en entreprise, vous serez noté sur 4 critères :

- ⤴ Le respect du cahier des charges : Dans le cas présent, il s'agit d'implémenter toutes les règles du jeu. Il ne doit pas être possible, par exemple, d'effectuer un déplacement non autorisé, de faire disparaître des pions en les sortant de la grille, etc. Ce critère est noté sur **10 points**.
- ⤴ Les aspects graphique et ergonomique : Vous avez carte blanche au niveau du graphisme (couleurs, mise en forme, utilisation d'images ou de fond sonore, etc.) et de l'ergonomie (par exemple : comment débiter une partie ou une recommencer une partie, la désignation du joueur qui doit jouer, ou encore comment déplacer un pion : drag & drop ou déplacement par multi-sélection de cases (case source vers case cible) dans la grille de jeu, etc.). Ce critère est noté sur **7 points**.
- ⤴ La propreté du code : Le code doit être bien indenté et commenté. Les fichiers doivent être organisés. Ce critère est noté sur **1,5 points**.
- ⤴ La livraison : L'archive de livraison doit respecter les consignes et doit être envoyée au plus tard à la date annoncée en cours. Ce critère est noté sur **1,5 points**.

Vous enverrez votre archive de livraison à l'adresse suivante : lopez.sylvain@inbox.com.

L'archive de livraison est une archive au format ZIP. Le nom de l'archive de livraison correspond aux noms de famille des étudiants séparés par un tiret.

<NOM 1>-<NOM 2>-<NOM 3>.ZIP

L'archive de livraison est constituée d'un répertoire « src/ » dans lequel vous placerez tous fichiers nécessaires au bon fonctionnement de votre projet. Veillez à bien utiliser des chemins relatifs dans l'inclusion de vos ressources afin que votre page HTML fonctionne quelque soit le répertoire où l'archive de livraison sera décompressée.

En outre, vous ajouterez un fichier README.TXT à la racine de votre archive de livraison. Dans ce fichier, vous indiquerez :

- Les noms et prénoms des étudiants qui ont réalisés le projet
- Un commentaire pour indiquer si vous avez réussi à implémenter complètement le jeu ou les points sur lesquels vous avez butés.