



22 Beautiful
 **jQuery** **PLUGINS**
FOR WEB DESIGNERS



Table des matières

Chapitre 1 – PLUGINS.....	4
1.1 - Présentation des plug-ins.....	5
1.1.1 - Généralités.....	5
1.1.2 – Un exemple : le carrousel jcarouselite.....	6
1.1.3 - Généralités sur l'utilisation d'un plug-in.....	11
1.2 – Trier une <table>.....	12
1.2.1 - Présentation.....	12
1.2.2 - Téléchargement et installation.....	12
1.2.3 - Démarche.....	13
1.2.4 - Trier une table statique.....	14
1.2.5 - Exercice : trier une table remplie avec des données JSON ou XML	17
1.3 - DataTables.....	18
1.3.1 - Présentation.....	18
1.3.2 - Téléchargement et installation.....	20
1.3.3 - Code.....	21
1.4 – Valider un formulaire.....	22
1.4.1 - Objectif.....	22
1.4.2 - Téléchargement et installation.....	23
1.4.3 - Démarche.....	25
1.4.4 - Validation de base.....	26
1.4.5 - Validation avancée.....	30
1.4.6 - Exercice : validation de formulaire.....	35
1.5 - PlugIns pour l'animation (Carrousel, transitions, zoom, ...)......	36
1.5.1 – Le carrousel jQueryCycle.....	36
1.5.2 - Le carrousel waterwheelCarousel.....	40
1.5.3 - Le diaporama slideshow.....	41
1.5.4 – Le zoomer jQZoom.....	47
1.5.5 - Le zoomer ElevateZoom.....	51
1.6 – Présenter des données dans une arborescence (Treeview).....	55
1.6.1 - Objectif.....	55
1.6.2 - Téléchargement et installation.....	55
1.6.3 - Principes et démarche.....	56
1.6.4 - Script.....	57
1.6.5 – Exercice : remplir un treeview avec des données XML.....	59
1.7 - Gérer les cookies.....	60
1.8 - Gérer le storage.....	62
1.9 - TimePickers.....	65
Chapitre 2 - ANNEXES.....	68

2.1 - Fichiers ressources.....	69
2.1.1 - cours.ini.....	69
2.1.2 - Connexion.php.....	70
2.1.3 - VillesDAO.php.....	71
2.1.4 - VillesDAOTests.php.....	72
2.2 - Corrigés des exercices.....	73

CHAPITRE 1 – PLUGINS

1.1 - PRÉSENTATION DES PLUG-INS

1.1.1 - Généralités

Les plugins – de l'anglais plug (prise) et donc se brancher - sont des bibliothèques supplémentaires ou plus exactement des **modules d'extension**, dépendantes de jQuery. Au 1^{er} février 2020 il en existe plus de 2800.

cf <https://plugins.jquery.com/>

Exemples de plug-ins :

Fonctionnalité	Plugin
Tri de <table>	Tablesorter, DataTables, ...
Validation de formulaire	Validate, ...
Création de menus (contextuel, radial, ...)	http://medialize.github.io/jQuery-contextMenu/
Gestion d'animations (Carrousel, Zoom, ...)	jcarousel, jquery.cycle, jqzoom, ...

1.1.2 – Un exemple : le carrousel jcarouselite

- **Objectif**

Un carrousel. Par défaut 3 images sont affichées, donc pour voir le défilement il faudra avoir plus de 3 images dans le code HTML.



- **Téléchargement et installation**

Bibliothèque très légère (2 ko) permettant de créer un carrousel de base.
Tested and compatible with jQuery 1.11.x, Mousewheel 3.1.x and Easing 1.3.

<https://kswedberg.github.io/jquery-carousel-lite/>

Téléchargez jcarouselite.js.

Copiez le fichier dans l'arborescence de votre site dans /jquery/jcarouselite/.

Cf aussi <http://sorgalla.com/jcarousel/> qui est plus riche.

- **Démarche et syntaxes**

La démarche et les syntaxes génériques sont explicitées au paragraphe suivant.

- **Syntaxe**

Avec les options minimum :

```
$(".diapo").jCarouselLite  
({  
    btnPrev: "classe du bouton précédent",  
    btnNext: "classe du bouton suivant" // et autres options  
});
```

Les options sont entre parenthèses - () - puis accolades - { } - . Chaque option est un nom suivi de deux points - : - suivi d'une valeur. Les options sont séparées par des virgules. La dernière option n'est pas suivie d'une virgule.

Toute <div> de classe .diapo sera gérée par jcarouselite.
Par défaut cela affichera 3 images à l'horizontal et le cycle est continu.

- **Quelques options**

Option	Description
btnPrev	Sélecteur du bouton précédent (obligatoire et en dehors de la <div> des images).
btnNext	Sélecteur du bouton suivant (obligatoire et en dehors de la <div> des images).
btnGo	Sélecteur du bouton d'accès direct (facultatif en dehors de la <div> des images).
vertical	Mouvement : true ou false .
circular	Circulaire : true ou false.
visible	Une valeur entre 1 et n ; par défaut à 3 .
auto	Défilement automatique : true ou false
speed	En millisecondes : par défaut rapide
scroll	Nombre de nouvelles images affichées lors du clic sur Next ou Previous. Par défaut : 1 .
start	Image de démarrage. Par défaut : 0 .
beforeStart	Fonction de callback à exécuter avant le défilement d'une image.
afterEnd	Fonction de callback à exécuter après le défilement d'une image.

Cf la documentation pour plus d'options (easing, mouseWheel, externalControls, ...).

- **Démarche**

Créez une `<div>` avec les `` dans des ``. Attribuez-lui, à la `<div>`, la classe **diapo** (le mot est réservé). Ne dimensionnez pas la `<div>`.

Créez des boutons en dehors de la `<div>`. Attribuez-leur une classe.

Codez la fonction `init()`.

Stylez éventuellement.

- Scripts

JQPIjCarouselLite.html

```
<style type="text/css">
    img{width:100px; height:150px;}
    .precedent{float:left; width:33px; height:150px; background-
color:black; color:white;}
    div.diapo{float:left; padding:0; overflow: hidden;}
    .suivant{float:left; width:33px; height:150px; background-
color:black; color:white;}
    .nettoyeur{clear:both;}
</style>
```

```
<script src="../jquery/jquery.js"></script>
<script src="../jquery/jcarousellite/jcarousellite.js"></script>
<script>
// -----
function init() {
    // --- Toute classe .diapo (une div) est gérée par jCarouselLite
    // --- Toute classe .suivant est gérée par la méthode btnNext()
    $(".diapo").jCarouselLite
    ({
        btnPrev: ".precedent",
        btnNext: ".suivant"
    });
}
$(document).ready(init);
</script>
```

```
<div id="centre">
    <h3>JQPIjCarouselLite</h3>

    <button class="precedent">&lt;</button>
    <div class="diapo">
        <ul>
            <li></li>
            <li></li>
            <li></li>
            <li></li>
            <li></li>
        </ul>
    </div>
    <button class="suivant">&gt;</button>
    <br class="nettoyeur" />
    <label id="lblMessage">Message</label>
</div>
```

- **Exercice : modifiez les paramètres d'affichage**

Affichez une seule image à la fois,
Le défilement est vertical,
Le défilement n'est pas circulaire.

jqPIjcarouseliteExo



Vous devez consulter la documentation officielle !

1.1.3 - Généralités sur l'utilisation d'un plug-in

Ce que l'on peut déduire du paragraphe précédent.

Un plug-in jQuery est une extension jQuery. C'est une classe qui enrichit jQuery.

Il faut télécharger le plug-in !!!

Un plug-in jQuery s'appuie sur jQuery !!!

Il faut donc référencer en premier, dans la page HTML, le fichier jquery.x.x.x.js.

Ensuite il faut référencer, dans la page HTML, le(s) .js du plug-in.

Il faut aussi référencer en premier dans le <head> le(s) fichier(s) .css du plug-in.

Tout plug-in jQuery, fondamentalement basé sur HTML 4, va utiliser deux attributs génériquement fonctionnels et communs à tous les éléments HTML : l'attribut id et l'attribut class.

Ensuite il faut lire la documentation du plug-in, et la relire, pour savoir comment structurer le code HTML, comment « classer » certains éléments, comment identifier certains éléments pivots utilisés par le plug-in et connaître les attributs à passer en paramètres de la fonction du plug-in.

Note : avec HTML5 et la possibilité de créer de nouveaux attributs personnalisés, pas des éléments – cela est réservé à XML - la donne pourrait changer. Bootstrap ne s'en est pas privé, bien au contraire.

1.1.4 - Exercice de veille technologique

Recherchez un plugin jQuery pour mettre en place un carrousel. Vous l'installerez et vous le testerez.

1.2 – TRIER UNE <TABLE>

1.2.1 - Présentation

- **Objectif**

Trier des données d'une <table>.
Le tri est réalisé dans le navigateur.

Cliquez sur un nom de colonne pour trier

Prénom	Nom
Reporter	Tintin
Le chien	Milou
Bianca	Castafiore
Laetitia	Casta

1.2.2 - Téléchargement et installation

Le plugin « tablesorter » permet de trier une table HTML.

<https://mottie.github.io/tablesorter/>

Téléchargez le PlugIn.

Décompactez-le.

Créez un dossier **tablesorter** dans le dossier jquery.

Copiez au minimum **jquery.tablesorter.js** dans ce dossier et éventuellement **tablesorter.css** (Dans certaines versions il est nommé style.css) et les boutons asc, desc, bg. Ce sont des .gif ou des png selon le thème.

1.2.3 - Démarche

Inclure les bibliothèques suivantes :

```
<script src="../../jquery/jquery.js"></script>  
<script src="../../jquery/tablesorter/jquery.tablesorter.js"></script>
```

Activer le tri sur une table :

```
$(document).ready(function()  
{  
    $("#tableClients").tablesorter();  
});
```

La table HTML doit être identifiée et comporter des en-têtes et un corps (thead et tbody).

1.2.4 - Trier une table statique

La table est triée en ordre croissant/décroissant en cliquant sur l'en-tête de la colonne de tri.

jqTableSorter

Cliquez sur un nom de colonne pour trier	
Prénom	Nom
Reporter	Tintin
Le chien	Milou
Bianca	Castafiore
Laetitia	Casta

Ou ça (Theme green)

Cliquez sur un nom de colonne pour trier	
↕ Prénom	↕ Nom
Reporter	Tintin
Le chien	Milou
Bianca	Castafiore
Laetitia	Casta

Ou ça (Theme blue)

Cliquez sur un nom de colonne pour trier	
Prénom ↕	Nom ↕
Reporter	Tintin
Le chien	Milou
Bianca	Castafiore
Laetitia	Casta

JQPITableSorter.html

- Basique

```
<!DOCTYPE html>
<html>
<head>
  <title>jqTableSorter0.html</title>
  <meta charset="utf-8" />
</head>

<body>
  <div id="centre">
    <h3>JQPITableSorter</h3>

    <table id="tableClients">
      <caption>Cliquez sur un nom de colonne pour trier</caption>
      <thead>
        <tr><th scope="col">Pr&eacute;nom</th><th
scope="col">Nom</th></tr>
      </thead>

      <tbody>
        <tr><td>Reporter</td><td>Tintin</td></tr>
        <tr><td>Le chien</td><td>Milou</td></tr>
        <tr><td>Bianca</td><td>Castafiore</td></tr>
        <tr><td>Laetitia</td><td>Casta</td></tr>
      </tbody>
    </table>

  </div>

  <script src="../jquery/jquery.js"></script>
  <script src="../jquery/tablesorter/jquery.tablesorter.js"></script>

  <script>
    $(document).ready(function()
    {
      $("#tableClients").tablesorter();
    });
  </script>

</body>
</html>
```


- La même table avec un peu style

jqTableSorter1

Cliquez sur un nom de colonne pour trier	
Prénom ▲	Nom ◆
Bianca	Castafiore
Laetitia	Casta
Le chien	Milou
Reporter	Tintin

Ajoutez le lien vers le CSS de tablesorter :

```
<link href="../../jquery/tablesorter/theme.green.css" rel="stylesheet"
type="text/css" media="screen" />
```

Et ajoutez class="tablesorter" comme attribut de la table et la classe.

```
<table id="tableClients" class="tablesorter-green">
```

Si vous voulez modifier la présentation, modifiez le contenu du fichier tablesorter.css :

Par exemple :

width: 75%; au lieu de 100% dans table.tablesorter

background-color: silver; au lieu de background-color: #FFFFCC; dans table.tablesorter
thead tr th, table.tablesorter tfoot tr th



background-color: yellow; au lieu de background-color: #999966; dans table.tablesorter
thead tr .headerSortDown, table.tablesorter thead tr .headerSortUp

Cliquez sur un nom de colonne pour trier	
Prénom ◆	Nom ▼
Reporter	Tintin
Le chien	Milou
Bianca	Castafiore
Laetitia	Casta

1.2.5 - Exercice : trier une table remplie avec des données JSON ou XML

Afficher les données de villes.json dans une table triable.
AJAX est un pré-requis, en l'occurrence \$.get().

jqTableSorterExoJSON

 Code postal	 Ville
75000	Paris
69000	Lyon
13000	Marseille
59000	Lille
33000	Bordeaux
38000	Grenoble
24200	Sarlat

Afficher les données de villesData.xml ou du JSON dans une table triable.
AJAX est un pré-requis, en l'occurrence \$.get().

1.3 - DATATABLES

1.3.1 - Présentation

Plugin très complet sur l'affichage et la gestion de données dans une <table>.
Lourd aussi (pas loin de 15 000 lignes de code !!! et donc 78 ko en version min et 426 ko en version open).
Il y a de très nombreux exemples et de nombreuses extensions.

<http://www.datatables.net/>

VillesLister Fichier Historique Outils Personnes Aide

localhost / localhost x VillesLister x

localhost/jqueryCours2015/php/VillesLister.php

Voir 10 enregistrements Filtre:

CP	Nom de la ville	Site
13000	Marseille	www.ma.net
24200	Sarlat	
24300	Carsac	
24400	Aillac	
59000	Lille	www.lille.fr
69000	Lyon	www.lyon.fr
75011	Paris 11	www.paris.fr
75012	Paris 12	www.paris.fr
75019	Paris XIX	www.paris.fr
78000	Versailles	

Affichage de 1 à 10 sur 14 enregistrements

Précédent 1 2 Suivant

Fichier Modifier Afficher Historique Outils Personnes Aide

localhost / localhost x VillesLister x

localhost/jqueryCours2015/php/VillesLister.php

Voir 10 enregistrements Filtre: par

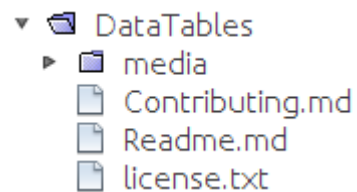
CP	Nom de la ville	Site
75011	Paris 11	www.paris.fr
75012	Paris 12	www.paris.fr
75019	Paris XIX	www.paris.fr

Affichage de 1 à 3 sur 3 enregistrements (filtered from 14 total entries)

Précédent 1 Suivant

1.3.2 - Téléchargement et installation

Téléchargez et dézippez.
Seul le dossier media suffit.



Ne gardez que les versions min.js que vous renommez.

Voilà les lignes de codes css et js à insérer dans votre page.

```
| <link rel="stylesheet" href="../jquery/DataTables/dataTables.css">
```

```
| <script src="../jquery/jquery.js"></script>
| <script src="../jquery/DataTables/dataTables.min.js"></script>
| <script>
|     $(document).ready(function() {
|         $('#tableVilles').dataTable();
|     });
| </script>
```

Quelques options

1.3.3 - Code

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="../../jquery/DataTables/dataTables.css">
    <title>jqDataTable</title>
  </head>

  <body>
    <?php
      require_once '../daos/Connexion.php';
      require_once '../daos/VillesDAO.php';

      $lcnx = Connexion::getConnexion("../daos/cours.ini");
      $tVilles = VillesDAO::selectAll($lcnx);
      Connexion::seDeconnecter($lcnx);
      ?>
    <table id="tableDesVilles">
      <thead>
        <tr>
          <th>CP</th>
          <th>Nom de la ville</th>
          <th>Site</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach ($tVilles as $key => $ville) { ?>
          <tr>
            <td><?php echo $ville["cp"]; ?></td>
            <td><?php echo $ville["nomVille"]; ?></td>
            <td><?php echo $ville["site"]; ?></td>
          </tr>
        <?php } ?>
      </tbody>
    </table>
    <script src="../../jquery/jquery.js"></script>
    <script src="../../jquery/DataTables/dataTables.js"></script>
    <script>
      $(document).ready(function () {
        $('#tableDesVilles').dataTable();
      });
    </script>
  </body>
</html>
```

1.4 – VALIDER UN FORMULAIRE

1.4.1 - Objectif

Valider les saisies d'un formulaire.

jqValidation1

Nom	<input type="text" value="Albert"/>	
Prénom	<input type="text" value="Milou"/>	
Age	<input type="text" value="5"/>	Veuillez entrer une valeur supérieure ou égale à 7.
CP	<input type="text" value="75011"/>	
E-Mail	<input type="text" value="p.b@free.fr"/>	
Site	<input type="text" value="http://www.pb.fr"/>	
Mot de passe	<input type="text" value="mdp"/>	Veuillez entrer au moins 5 caractères.
Ressaisir le mot de passe	<input type="text" value="mdp"/>	
<input type="button" value="Valider"/>		

Pour la suite les mots de passe passeront en type='password'.

1.4.2 - Téléchargement et installation

Le plugin testé est le suivant : `.validate`.

La version testée est la version 1.13.1 (1^{er} mars 2015).



[HOME](#)[CONTRIBUTE](#)[DOCUMENTATION](#)[REFERENCE DOCS](#)

jQuery Validation Plugin

This jQuery plugin makes simple clientside form validation easy, whilst still offering plenty of customization options. It makes a good choice if you're building something new from scratch, but also when you're trying to integrate something into an existing application with lots of existing markup. The plugin comes bundled with a useful set of validation methods, including URL and email validation, while providing an API to write your own methods. All bundled methods come with default error messages in english and translations into 37 other languages.

Ce plugin permet de valider un formulaire soit de façon basique soit en personnalisant la validation.

La validation est réalisée sur la perte de focus et sur la soumission du formulaire.

Téléchargement : <http://jqueryvalidation.org/>

Documentation : <http://jqueryvalidation.org/documentation/>

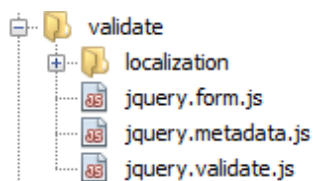
Téléchargez le plugIn.

Décompactez-le.

Créez un dossier **validate** dans le dossier jquery de votre site.

Copiez au minimum **jquery.validate.js** dans ce dossier.

Dans ce même dossier copiez éventuellement les fichiers **jquery.form.js** et **jquery.metadata.js** ainsi que le dossier **localization** pour paramétrer la langue des messages d'erreurs.



1.4.3 - Démarche

Inclure les bibliothèques :

```
<script src="../../jquery/jquery.js"></script>  
<script src="../../jquery/validate/jquery.validate.js"></script>  
<script src="../../jquery/validate/localization/messages_fr.js"></script>
```

et éventuellement

```
<script src="../../jquery/validate/jquery.form.js"></script>  
<script src="../../jquery/validate/jquery.metadata.js"></script>
```

Pour coloriser les messages d'erreur :

```
label.error{color:red;}
```

1.4.4 - Validation de base

- **Syntaxes**

La validation du formulaire

```
$("#idDuFormulaire").validate(); // --- Dans une fonction init()
```

Le bouton submit doit être de class="submit".

Les éléments à valider

Pour la validation de base il suffit de préciser les règles :

- ✓ Soit comme valeur d'un attribut **class** de l'élément du formulaire. Les règles peuvent être cumulées pour un même élément, le séparateur étant l'espace.
- ✓ Soit avec un attribut spécial.

Class	Description
required	Obligatoire
email	Format e-mail
url	Format url (http:// ...)
number	Un nombre décimal (Avec un point décimal)
digits	Un nombre entier
date	Date au format jj/mm/aaaa ou jj/mm/aa ou aaaa/mm/jj
dateISO	Date au format aaaa-mm-jj

Attribut spécial	Description
minlength	Longueur mini
maxlength	Longueur maxi
min	Valeur mini
max	Valeur maxi
equalTo("#id")	Egale à la valeur d'un autre élément

```
| <input it="itNom" type="text" value="" class="required" minlength="2" />
```

```
| <input id="itEMail" name="itEMail" type="text" value="p.b@free"  
| class="required email" />
```

- **Exemple**

Les nom et prénom sont requis et sont d'au moins 2 caractères.

L'âge est compris entre 7 et 77 ans.

Le CP est formé des 5 chiffres.

L'e-mail est requis et doit avoir un format correct.

Le site est facultatif. Il doit avoir un format correct s'il est saisi.

Le premier mot de passe est requis et comporte de 5 à 10 caractères.

Le deuxième mot de passe est requis et doit être égal au premier.

jqValidation1

Nom	<input type="text" value="Albert"/>	
Prénom	<input type="text" value="Milou"/>	
Age	<input type="text" value="5"/>	Veuillez entrer une valeur supérieure ou égale à 7.
CP	<input type="text" value="75011"/>	
E-Mail	<input type="text" value="p.b@free.fr"/>	
Site	<input type="text" value="http://www.pb.fr"/>	
Mot de passe	<input type="text" value="mdp"/>	Veuillez entrer au moins 5 caractères.
Ressaisir le mot de passe	<input type="text" value="mdp"/>	
<input type="button" value="Valider"/>		

Pour franciser les messages :

```
<script src="../../jquery/validate/localization/messages_fr.js"></script>
```

```

<!DOCTYPE html>
<html>
<head>
  <title>jqValidate1</title>

  <meta charset="utf-8" />

  <style type="text/css">
    *{font-size:x-small;}
    label.error{color:red; width:300px;}
    form label{float:left; display:block; width:150px; margin:3px;}
    form input{float:left; display:block; width:150px; margin:3px;}
    .nettoyeur{clear:both;}
  </style>
</head>

<body>
  <div id="centre">
    <h3>jqValidation1</h3>
    <form action="" method="get" id="formSaisie">
      <label>Nom</label>
      <input id="itNom" name="itNom" type="text" value="Albert"
class="required" minlength="2" />
      <br class="nettoyeur"/>

      <label>Pr&eacute;nom</label>
      <input id="itPrenom" name="itPrenom" type="text" value="Milou"
class="required" minlength="2" />
      <br class="nettoyeur"/>

      <label>Age</label>
      <input id="itAge" name="itAge" type="text" value="7"
class="required" min="7" max="77" />
      <br class="nettoyeur"/>

      <label>CP</label>
      <input id="itCP" name="itCP" type="text" value="75011"
class="required digits" minlength="5" maxlength="5" />
      <br class="nettoyeur"/>

      <label>E-Mail</label>
      <input id="itEMail" name="itEMail" type="text"
value="p.b@free.fr" class="required email" />
      <br class="nettoyeur"/>

      <label>Site</label>
      <input id="itSite" name="itSite" type="text"
value="http://www.pb.fr" class="url" />
      <br class="nettoyeur"/>

      <label>Mot de passe</label>
      <input id="itMdp1" name="itMdp1" type="text" value="mdp"
class="required" minlength="5" maxlength="10" />
      <br class="nettoyeur"/>

      <label>Ressaisir le mot de passe</label>

```

```
        <input id="itMdp2" name="itMdp2" type="text" value="mdp"
class="required" equalTo="#itMdp1" />
        <br class="nettoyeur"/>

        <label>&nbsp;</label>
        <input id="btValider" type="submit" class="submit" />
    </form>
</div>

<script src="../../jquery/jquery.js"></script>
<script src="../../jquery/validate/jquery.validate.js"></script>
<script src="../../jquery/validate/localization/messages_fr.js"></script>

<script>
    // -----
    function init() {
        // Valide le formulaire quand il est soumis
        // et lors des pertes de focus
        $("#formSaisie").validate();
    }

    $(document).ready(init);
</script>

</body>
</html>
```

1.4.5 - Validation avancée

En plus des règles précédentes il est possible d'en utiliser d'autres ainsi que d'utiliser des règles personnalisées.

Règle/Attribut	Description
range(mini, maxi)	Une plage de valeurs.
rangelength(mini, maxi)	Une plage pour la taille de la valeur.
required(expression)	Une valeur est requise pour un champ si l'expression renvoie vrai. L'expression dépend d'une case à cocher, d'une sélection dans une balise <select>.
required(callback)	Une valeur renvoyée par une fonction.
remote(option)	Validation en fonction d'un retour d'un script PHP ou autre (JSP, ASP, ASPX).

Une méthode personnalisée :

On ajoute une méthode de contrôle à l'objet \$.validator via `addMethod("nomDeMéthode", fonction anonyme qui renvoie un booléen)`.

```
// -----  
$.validator.addMethod("validationSexe", function(value)  
// -----  
{  
    return (value == "M" || value == "F");  
}, "Entrez le bon sexe!");
```

Il est aussi possible de personnaliser les messages.

```
messages :  
{  
    itNom: "Le nom est obligatoire et d'au moins 2 lettres",  
    ...  
}
```

Exemple

jqValidation2

Nom	<input type="text" value="A"/>	Le nom est obligatoire et d'au moins 2 lettres
Prénom	<input type="text" value="Milou"/>	
Age	<input type="text" value="5"/>	L'age est obligatoire et doit être compris entre 7 et 77
Date de naissance	<input type="text" value="03/05/1980"/>	
Sexe	<input type="text" value="Z"/>	Entrez le bon sexe
CP	<input type="text" value="7501"/>	Un CP est composé de 5 chiffres
NewsLetter ?	<input type="checkbox"/>	
E-Mail	<input type="text" value="milou@free"/>	Saisissez correctement votre e-mail
Site	<input type="text" value="http://www.pb.fr"/>	
Mot de passe	<input type="text" value="mdp"/>	Veuillez entrer entre 5 et 10 caractères.
Ressaisir le mot de passe	<input type="text" value="mdp12"/>	Les 2 mots de passe doivent être identiques
<input type="button" value="Valider"/>		

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="Expires" content="mon, 8 jun 2009 14:30:00 GMT">
<meta http-equiv="cache-control" content="no-cache, must-revalidate">

<meta charset="utf-8" />
<title>jqValidation2</title>

<style type="text/css">
    *{font-size:x-small;}
    label.error{color:red; width:300px;}
    form label{float:left; display:block; width:150px; margin:3px;}
    form input{float:left; display:block; width:150px; margin:3px;}
    .nettoyeur{clear:both;}
</style>

<script src="../../../jquery/jquery.js"></script>

<script src="../../../jquery/jquery.validate.js"></script>
<!--
<script src="../../../jquery/validate/jquery.form.js"></script>
<script src="../../../jquery/validate/jquery.metadata.js"></script>
-->

<script src="../../../jquery/validate/localization/messages_fr.js"></script>

<script>
    // -----
    $.validator.addMethod("validationSexe", function(value) {
        return (value == "M" || value == "F");
    }, "Entrez le bon sexe!");
```



```

// -----
function init() {
    $("#formSaisie").validate
    (
    {
        rules:
        {
            itNom:
            {
                // --- Nom obligatoire et >= 2
                required: true,
                minlength: 2
            },
            itAge:
            {
                required: true, // --- Obligatoire
                digits: true, // --- Des nombres
                range: [7,77] // --- Age entre 7 et 77
            },
            itDN:
            {
                date: true // jj/mm/aaaa
                //dateISO: true // aaaa-mm-jj
            },
            itSexe: "validationSexe", // OK - Nom d'une méthode
            itEmail:
            {
                // --- Email obligatoire si la case a cocher

                required: "#cbxNewsletter:checked",
                // --- Format email
                email: true
            },
            itCP:
            {
                digits: true,
                rangelength:[5,5]
            },
            itMdp1:
            {
                // --- Le mot de passe entre 5 et 10 lettres
                rangelength:[5,10],
            },
            itMdp2:
            {
                // --- itMdp2 = itMdp1
                equalTo: "#itMdp1"
            }
        },
        messages:
        {
            itNom: "Le nom est obligatoire et d'au moins 2
lettres",
            itAge: "L'age est obligatoire et doit être compris
entre 7 et 77",
            //itDN: "Date au format yyyy-mm-dd", // dateISO

```

```

        itDN: "Date au format jj/mm/aaaa", // date
        itSexe: "Entrez le bon sexe",
        itEMail:
        {
            required: "Si vous voulez recevoir la
NewsLetter vous devez saisir un e-mail",
            email: "Saisissez correctement votre e-mail"
        },
        itCP:
        {
            digits: "Que des chiffres !!!",
            ranglength: "Un CP est composé de 5 chiffres"
        },
        itMdp2: "Les 2 mots de passe doivent etre
identiques"
    }
}

);
}

$(document).ready(init);
</script>
</head>

<body>
    <div id="centre">
        <h3>jqValidation2</h3>

        <form action="" method="get" id="formSaisie">
            <label>Nom</label>
            <input it="itNom" name="itNom" type="text" value="Albert" />
            <br class="nettoyeur"/>

            <label>Pr&eacute;nom</label>
            <input id="itPrenom" name="itPrenom" type="text" value="Milou"
/>

            <br class="nettoyeur"/>

            <label>Age</label>
            <input id="itAge" name="itAge" type="text" value="7" />
            <br class="nettoyeur"/>

            <label>Date de naissance</label>
            <input id="itDN" name="itDN" type="text" value="03/05/1960"/>
            <br class="nettoyeur"/>

            <label>Sexe</label>
            <input id="itSexe" name="itSexe" type="text" value="M"/>
            <br class="nettoyeur"/>

            <label>CP</label>
            <input id="itCP" name="itCP" type="text" value="75011" />
            <br class="nettoyeur"/>

            <label>NewsLetter ?</label>
            <input id="cbxNewsLetter" name="cbxNewsLetter"
type="checkbox"/>
            <br class="nettoyeur"/>

```

```
        <label>E-Mail</label>
        <input id="itEMail" name="itEMail" type="text"
value="milou@free.fr" />
        <br class="nettoyeur"/>

        <label>Site</label>
        <input id="itSite" name="itSite" type="text"
value="http://www.pb.fr" />
        <br class="nettoyeur"/>

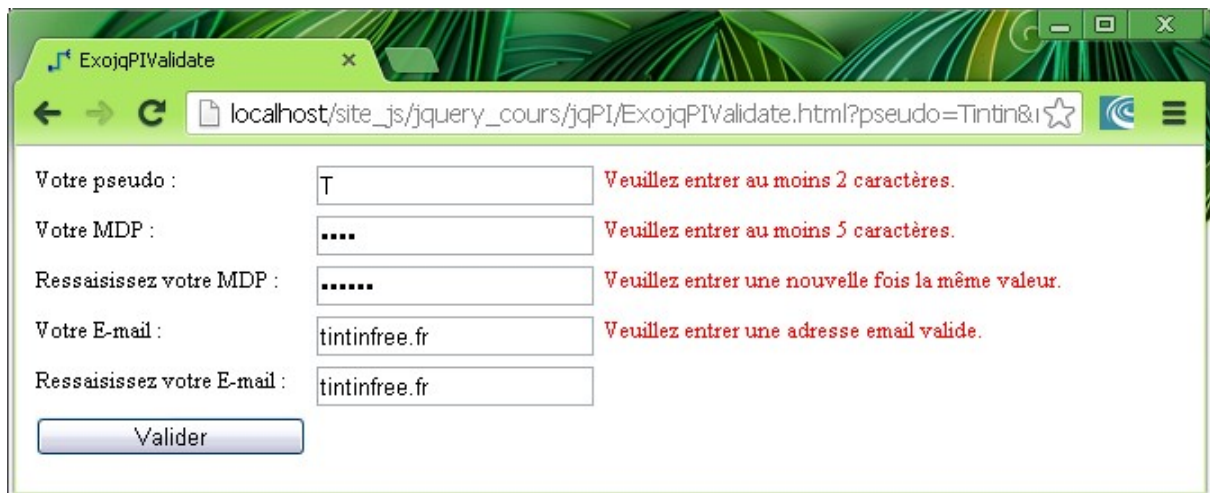
        <label>Mot de passe</label>
        <input id="itMdp1" name="itMdp1" type="text" value="mdp12" />
        <br class="nettoyeur"/>

        <label>Ressaisir le mot de passe</label>
        <input id="itMdp2" name="itMdp2" type="text" value="mdp12" />
        <br class="nettoyeur"/>

        <label>&nbsp;</label>
        <input id="btValider" type="submit" class="submit" />
    </form>
</div>
</body>
</html>
```

1.4.6 - Exercice : validation de formulaire

Le pseudo est d'au moins 2 caractères,
Le mot de passe est au minimum de 5 caractères,
L'e-mail est obligatoire et doit être conforme.
Le mot de passe et l'e-mail doivent être saisis 2 fois pour contrôle.



The screenshot shows a web browser window titled 'ExojqPIValdate'. The address bar shows the URL: `localhost/site_js/jquery_cours/jqPI/ExojqPIValdate.html?pseudo=Tintin&...`. The form contains five input fields with corresponding labels and validation messages:

- Votre pseudo :** Input field contains 'T'. Validation message: 'Veuillez entrer au moins 2 caractères.'
- Votre MDP :** Input field contains '....'. Validation message: 'Veuillez entrer au moins 5 caractères.'
- Ressaisissez votre MDP :** Input field contains '.....'. Validation message: 'Veuillez entrer une nouvelle fois la même valeur.'
- Votre E-mail :** Input field contains 'tintinfree.fr'. Validation message: 'Veuillez entrer une adresse email valide.'
- Ressaisissez votre E-mail :** Input field contains 'tintinfree.fr'.

At the bottom of the form is a button labeled 'Valider'.

Votre point de départ :

```
<form>
  <label>Votre pseudo : </label>
  <input type="text" name="pseudo" value="Tintin" />
  <br class="nettoyeur"/>

  <label>Votre MDP : </label>
  <input type="password" name="mdp" value="Tintin" />
  <br class="nettoyeur"/>

  <label>Ressaisissez votre MDP : </label>
  <input type="password" name="mdpBis" value="Tintin" />
  <br class="nettoyeur"/>

  <label>Votre E-mail : </label>
  <input type="text" name="email" value="tintin@free.fr" />
  <br class="nettoyeur"/>

  <label>Ressaisissez votre E-mail : </label>
  <input type="text" name="emailBis" value="tintin@free.fr" />

  <br class="nettoyeur"/>
  <input type="submit" name="btSubmit" />
</form>
```

1.5 - PLUGINS POUR L'ANIMATION (CARROUSEL, TRANSITIONS, ZOOM, ...)

1.5.1 – Le carrousel jQueryCycle

Note : KO avec jQuery >= 1.9 parce que le plugin utilise \$.browser.

- **Objectif**

Afficher des images ou des textes avec de nombreuses transitions (un diaporama).

jqImagePICycle



- **Téléchargement et installation**

Téléchargez à partir de <http://www.malsup.com/jquery/cycle/>

Copiez le fichier **jquery.cycle.all.js** dans un dossier de votre site (jquerycycle par exemple).

- **Syntaxe**

```
$(".classe de la div contenant les images").cycle(  
{  
    fx: "effet",  
    speed: transition en millisecondes // --- Et autres options  
});
```

Les effets (fx) : blindX, blindY, blindZ, cover, curtainX, curtainY, fade, fadeZoom, growX, growY, scrollUp, scrollDown, scrollLeft, scrollRight, scrollHorz, scrollVert, shuffle, slideX, slideY, toss, turnUp, turnDown, turnLeft, turnRight, uncover, wipe, zoom.

Pour les autres options cf la liste : <http://www.malsup.com/jquery/cycle/options.html>

- **Démarche**

Créez une <div>, avec une série de balises à l'intérieur, attribuez-lui une classe, codez.

- Exemple

jqImagePICycle.html

```
<style type="text/css">
    .diapo{height:192px; width:152px; border:0px solid black; padding:0}
    .diapo img{padding:15px; border:1px solid silver; background-
color:white;}
</style>

<script src="../../jquery/jquery-1.8.2.js"></script>
<script src="../../jquery/jquerycycle/jquery.cycle.all.js"></script>

<script>
// -----
function init() {
    $('<b>.diapo</b>').cycle(
    {
        fx: 'shuffle', // Effet similaire au passage d'une carte
derrière une autre
        speed: 500
    });
}

$(document).ready(init);
</script>

<div class="diapo">
    
    
    
    <a href="../../html/immortel.html"></a>
    
</div>
```

Autres possibilités ...

```
| $('#shuffle').cycle({  
|   fx:      'shuffle',  
|   easing:  'easeOutBack',  
|   delay:   -4000  
| });
```

```
| $('#zoom').cycle({  
|   fx:      'zoom',  
|   sync:    false,  
|   delay:   -2000  
| });
```

```
| $('#right').cycle({  
|   fx:      'scrollRight',  
|   next:    '#right',  
|   timeout: 0,  
|   easing:  'easeInOutBack'  
| });
```

```
| $('#up').cycle({  
|   fx:      'curtainX',  
|   sync:    false,  
|   delay:   -2000  
| });
```


1.5.2 - Le carrousel waterwheelCarousel

Présentation

<http://www.bkosborne.com/jquery-waterwheel-carousel>

Intéressant pour son extrême simplicité de mise en place.

MAIS UN VRAI BINZ !

Syntaxes

<http://www.bkosborne.com/jquery-waterwheel-carousel/options>

1.5.3 - Le diaporama slideshow

<http://i-like-robots.github.io/jQuery-Slideshow/>

Objectif

Un carrousel avec éventuellement les boutons précédent et suivant et les boutons pour accéder directement à une image.
Possibilité de défilement automatique.
Possibilité de boucler ...



Notes :

Facile à mettre en œuvre.
Documentation médiocre.

Site

<http://i-like-robots.github.io/jQuery-Slideshow/>

Téléchargement et installation

Téléchargez le zip.
Dézippez-le.
Copiez la production (les fichiers slideshow.js et slideshow.css) dans un dossier de votre site.

Syntaxes

cf <http://i-like-robots.github.io/jQuery-Slideshow/>

Les inclusions :

```
<link href="../../jquery/pi-slideshow/slideshow.css" rel="stylesheet"
type="text/css" />

<script src="../../jquery/jquery-2.0.0.js"></script>
<script src="../../jquery/pi-slideshow/slideshow.js"></script>
```

Les éléments :

La <div> container doit être "classée" **slideshow**

La liste doit être "classée" **carousel**

Les éléments de la liste, les , doivent être "classés" **slide**

Vous devez dimensionner les images puis le slideshow par un multiple de la largeur des images.

```
img{width: 200px; height: 300px;}
.slideshow{width: 200px; height: 300px;}
```

Le code jQuery :

```
$('.slideshow').slides(  
{  
  option1: valeur1,  
  option2: valeur2,  
  option3: valeur3  
}  
);
```

remarquez l'absence de , pour la dernière option!

Quelques options :

Attribut	Valeur	Description
carousel	carousel	La classe du conteneur
offset	1	Rang de l'image lors du démarrage
pagination	true false	Liens pour chaque image
skip	true false	Présence des boutons précédent et suivant
previousText	Previous texte	Texte du bouton
nextText	Next texte	Texte du bouton
auto	6000 durée en millisecondes	Défilé automatique et délai de passage d'une image à l'autre
loop	true false	Boucle, ie revient à la première une fois le défilé terminé

Code exemple

```
<!DOCTYPE html>
<html>
  <head>
    <title>jsSlideShow</title>
    <meta charset="UTF-8">

    <script src="../../jquery/jquery-2.0.0.js"></script>
    <script src="../../jquery/pi-slideshow/slideshow.js"></script>

    <link href="../../jquery/pi-slideshow/slideshow.css" rel="stylesheet"
type="text/css" />

    <style>
      img{width: 200px; height: 300px;}
      .slideshow{width: 200px; height: 300px;}
    </style>

  </head>

  <body>

    <div class="slideshow">
      <ul class="carousel">
        <li class="slide"></li>
        <li class="slide"></li>
        <li class="slide"></li>
        <li class="slide"></li>
        <li class="slide"></li>
        <li class="slide"></li>
        <li class="slide"></li>
        <li class="slide"></li>
        <li class="slide"></li>
      </ul>
    </div>

    <script>
      var slideshow =
        $($('.slideshow').slides(
          {
            visible: 1, /* Inoperant */
            skip: true, /* Boutons previous et next */
            pagination: false, /* Boutons 1, 2, etc */
            slideWidth: true, /* Meme largeur pour les images a voir*/
            auto: false, /* Scroll auto en millisecondes ou false */
            loop: false,
            previousText: "<",
```

```
        nextText: ">",  
        autostop: true  
    }  
);  
</script>  
  
</body>  
</html>
```

Exercice : cf le TP (scroll automatique, ...)

Note : KO avec jQuery >= 1.9 parce que le plugin utilise \$.browser.

- **Objectif**

Zoomer sur une partie d'une image.



- **Téléchargement et installation**

jQzoom Evolution Library v2.3 du 3 mai 2011.

Téléchargez la bibliothèque à partir de : <http://www.mind-projects.it/projects/jqzoom/>
Décompactez-la.

Copiez les fichiers **jquery.jqzoom-core.js** et **jquery.jqzoom.css** ainsi que le dossier **images** dans un dossier (/jquery/jqzoom/) de votre site.

Cf pour d'autres plugins de zoom compatible jQuery 2.x

<http://www.tripwiremagazine.com/2012/12/jquery-image-zoom.html>

- **Syntaxes**

```
$(".classe de l'image").jqzoom(); // --- Syntaxe de base
```

```
$(".classe de l'image").jqzoom(parametres); // --- Syntaxe améliorée
```

Les paramètres sont :

Paramètre	Description
zoomType	Type de zoom : 'standard' 'reverse'
zoomWidth	Largeur de la popup : 200
zoomHeight	Hauteur de la popup : 200
xOffset	Position x de la popup : 10
yOffset	Position y de la popup : 0
position	Positionnement de la popup : 'right' 'left' 'top' 'bottom'
lens	Loupe : True false
imageOpacity	Opacité : 0.2
title	Titre : true false
showEffect	L'effet d'apparition de la popup : 'show' 'fadein'
hideEffect	L'effet de masquage de la popup : 'hide' 'fadeout'.
fadeinSpeed	Vitesse du fondu si showEffect est à 'fadein' : 'fast' 'slow' 'medium'
fadeoutSpeed	Vitesse du fondu si hideEffect est à 'fadeout' : 'fast' 'slow' 'medium'
showPreload	Affichage du message de chargement : True false
preloadText	Message lors du chargement : 'Loading zoom'
preloadPosition	Position du message du chargement : 'center'

- **Exemple**



- **Démarche**

Inclusion des fichiers .css et .js.

```
<link rel="stylesheet" type="text/css"
href="../../jquery/jqzoom/jquery.jqzoom.css" />

<script src="../../jquery/jquery.js"></script>
<script type='text/javascript' src='../jquery/jqzoom/jquery.jqzoom-
core.js'></script>
```

Les images

Pour chaque image vous devez avoir un petit format et un grand format (pour le lien). Si vous voulez tester avec une seule image attribuez une petite "width" à la balise . L'image du zoom est dans un lien <a>.

```
<a href="../../images/cinema/0.jpg" class="imageZoom" title="Titre Film">
  
</a>
```

Code

Codez la méthode jqzoom([paramètres]).

- **Script**

```
<link rel="stylesheet" type="text/css"
href="../jquery/jqzoom/jquery.jqzoom.css" />
```

```
<div id="centre">
    <h3>jqImagePIjqZoom</h3>

    <a href="../images/cinema/0.jpg" class="imageZoom" title="Titre Film">
        
    </a>
</div>
```

```
<script src="../jquery/jquery.js"></script>
<script type='text/javascript'
src='../jquery/jqzoom/jquery.jqzoom.js'></script>

<script>
// -----
function init() {
    var parametres =
    {
        zoomWidth: 200,
        zoomHeight: 180,
        xOffset: 30,
        yOffset: 0,
        position: "right",
        title: false
    };

    $(".imageZoom").jqzoom(parametres);
}

$(document).ready(init);
</script>
```

1.5.5 - Le zoomer ElevateZoom

Objectif

Zoomer sur une image.



Téléchargement et installation

<http://www.elevateweb.co.uk/image-zoom>

Téléchargez le zip et dézipper dans un dossier de votre site.

Syntaxes

<http://www.elevateweb.co.uk/image-zoom/configuration>

Attribut	Valeur par défaut	Description
easing	false	Set to true to activate easing. Possible Values: "True", "False"
easingType	zoomdefault	default easing type is easeOutExpo, $(t==d) ? b+c : c * (-Math.pow(2, -10 * t/d) + 1) + b$
easingDuration	2000	
lensSize	200	used when zoomType set to lens, when zoom type is set to window, then the lens size is auto calculated
zoomWindowWidth	400	Width of the zoomWindow (Note: zoomType must be "window")
zoomWindowHeight	400	Height of the zoomWindow (Note: zoomType must be "window")
zoomWindowOffetx	0	x-axis offset of the zoom window
zoomWindowOffety	0	y-axis offset of the zoom window
zoomWindowPosition	1	Once positioned, use zoomWindowOffsetx and zoomWindowOffety to adjust. Possible values: 1-16
lensFadeIn	false	Set as a number e.g 200 for speed of Lens fadeIn
lensFadeOut	false	Set as a number e.g 200 for speed of Lens fadeOut
zoomWindowFadeIn	false	Set as a number e.g 200 for speed of Window fadeIn
zoomWindowFadeOut	false	Set as a number e.g 200 for speed of Window fadeOut
zoomTintFadeIn	false	Set as a number e.g 200 for speed of Tint fadeIn
zoomTintFadeOut	false	Set as a number e.g 200 for speed of Tint fadeOut
borderSize	4	Border Size of the ZoomBox - Must be set here as border taken into account for plugin calculations
zoomLens	true	set to false to hide the Lens
borderColor	#888	Border Colour
lensBorder	1	Width in pixels of the lens border
lensShape	square	can also be round (note that only modern browsers support round, will default to square in older browsers)

zoomType	window	Possible Values: Lens, Window, Inner
containLensZoom	false	for use with the Lens Zoom Type. This makes sure the lens does not fall outside the outside of the image
lensColour	white	colour of the lens background
lensOpacity	0.4	used in combination with lensColour to make the lens see through. When using tint, this is overrided to 0
lenszoom	false	
tint	false	enable a tint overlay, other options: true
tintColour	#333	colour of the tint, can be #hex, word (red, blue), or rgb(x, x, x)
tintOpacity	0.4	opacity of the tint
gallery	null	This assigns a set of gallery links to the zoom image
cursor	default	The default cursor is usually the arrow, if using a lightbox, then set the cursor to pointer so it looks clickable - Options are default, cursor, crosshair

Script

```
<!--
jqElevateZoom.html
-->
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="user-scalable=no, width=device-
width, initial-scale=1.0, maximum-scale=1.0" />
    <meta charset="utf-8" />

    <script src="../../jquery/jquery-2.0.0.js"></script>
    <script src='../jquery/elevateZoom/jquery.elevateZoom-
2.5.5.min.js'></script>

    <title>jqElevateZoom.html</title>
  </head>

  <body>
    <div id="centre">
      <article id="article">
        
      </article>
    </div>

    <script>
      $('#imageZoom').elevateZoom(
      {
        borderSize:0,
        zoomLevel:0.5,
        zoomWindowWidth:600,
        zoomWindowHeight:300,
        zoomWindowOffetx:100,
        zoomWindowOffety:10
      });
    </script>
  </body>
</html>
```

1.6 – PRÉSENTER DES DONNÉES DANS UNE ARBORESCENCE (TREEVIEW)

1.6.1 - Objectif

Afficher dans un TreeView une structure hiérarchique.



1.6.2 - Téléchargement et installation

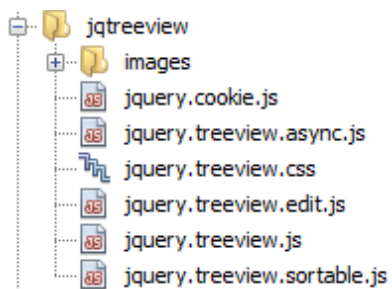
<http://bassistance.de/jquery-plugins/jquery-plugin-treeview/>

Téléchargez le .zip.

Décompactez-le.

Créez un dossier nommé **jqtreeview** dans le dossier jquery de votre site.

Dans ce dossier copiez les fichiers .js et .css ainsi que le dossier images.



1.6.3 - Principes et démarche

L'arborescence est formée de imbriquées.

Une arborescence est formée d'une racine, de branches et de feuilles.

Le niveau racine de l'arbre est une balise de class "filetree treeview-famfamfam".

```
| <ul id="arbre" class="filetree treeview-famfamfam">
```

Le texte de la racine est dans une balise dans

```
| <li><span class="folder">Livres</span>
```

Une branche est de class "folder".

Une branche fermée à l'affichage est de class "closed".

Une feuille est de class "file".

Les textes sont dans des .

1.6.4 - Script

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8"/>
  <title>jqTreeView_1</title>

  <link rel="stylesheet" href="../jquery/jqtreetview/jquery.treeview.css"
/>
  <link rel="stylesheet" href="../css/treeview.css" />

  <script src="../jquery/jquery.js"></script>

  <script src="../jquery/jqtreetview/jquery.treeview.js"></script>

  <script>
    $(document).ready(function(){
      // --- Pour que la liste soit un treeview
      $("#arbre").treeview();
    });
  </script>
</head>

<body>

<div>
  <!-- Racine -->
  <ul id="arbre" class="filetree treeview-famfamfam racine">
    <li> <!-- Racine -->
      <!-- Texte Racine -->
      <span>&nbsp;Livres</span>

      <ul> <!-- Liste des rubriques -->

        <!-- Une rubrique -->
        <li class="closed">
          <span class="folder">Romans</span>
          <ul>
            <!-- Feuille -->
            <li><span class="file">Délicatesse</span></li>
            <li><span class="file">La curée</span></li>
            <li><span class="file">La conquête de
Plassans</span></li>
          </ul>
        </li>

        <!-- Rubrique -->
        <li class="closed"> <!-- Fermée au premier affichage -->
          <span class="folder">Essais</span> <!-- Avec l'icone
dossier -->
          <ul> <!-- Sous-Rubriques -->
            <!-- Sous-Rubrique -->
            <li>
              <span class="closed">Philo</span>
            <ul> <!-- Liste de feuilles -->
```

```

                                <!-- Feuille -->
                                <li><span class="file">La
morale</span></li>
                                <!-- Feuille -->
                                <li><span
class="file">L'éthique</span></li>
                                </ul> <!-- Fin Liste de feuilles -->
                                </li>
                                <!-- Sous-Rubrique -->
                                <li>
                                <span class="closed">Politique</span>
                                <ul>
                                <!-- Feuilles -->
                                <li><span
class="file">Sénatoriales</span></li>
                                <li><span
class="file">Présidentielles</span></li>
                                </ul>
                                </li>
                                </ul> <!-- Fin Sous-Rubriques -->
                                </li>

                                <!-- Rubrique -->
                                <li>
                                <span class="folder">Arts</span>
                                <ul>
                                <li><span class="file">Renoir</span></li>
                                <li><span class="file">Caravage</span></li>
                                <li><span class="file">Renoir</span></li>
                                </ul>
                                </li>

                                <!-- Rubrique -->
                                <li>
                                <span class="folder">Dictionnaires</span>
                                </li>

                                </ul> <!-- Fin Liste des rubriques -->

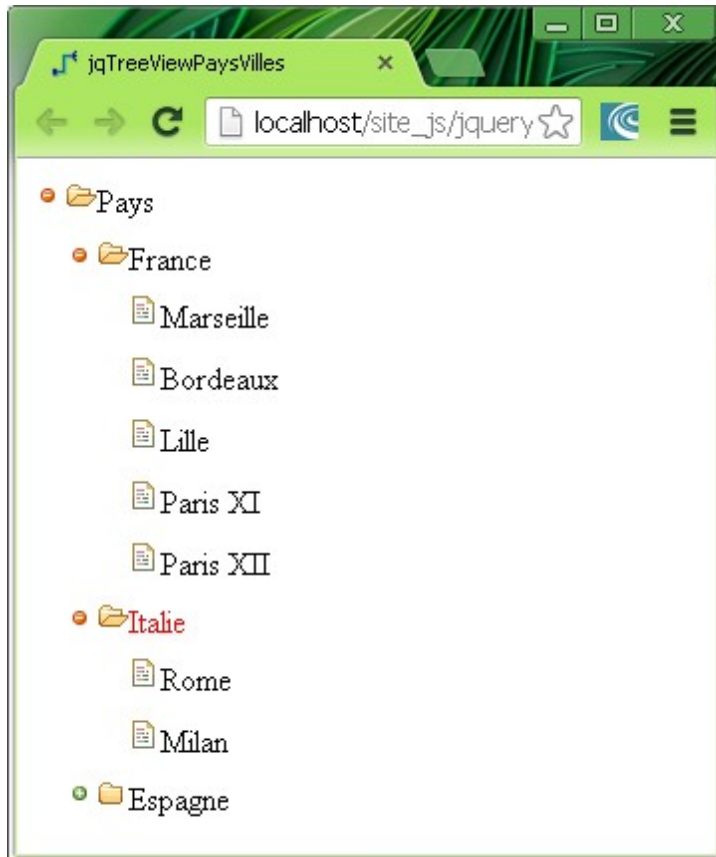
                                </li> <!-- Fin Livres -->
                                </ul> <!-- Fin arbre -->

</div>
</body>
</html>

```

1.6.5 – Exercice : remplir un treeview avec des données XML

Créez cet arbre à partir du fichier villesData.xml.



1.7 - GÉRER LES COOKIES

Téléchargez le plugin :

<https://github.com/carhartl/jquery-cookie>

Documentation : <https://github.com/carhartl/jquery-cookie/blob/master/README.md>

Usage :

Copiez le fichier jquery.cookie.js dans l'arborescence du site.

Incluez jQuery : `<script src="../../jquery/jquery.js"></script>`

Incluez le plugin : `<script src="../../jquery/PICookies/jquery.cookie.js"></script>`

Codez!

Pour créer un cookie

```
$.cookie('nomDuCookie', 'valeur' [, {options}])
```

Les options : date d'expiration, nom de domaine.

```
/*
 * Creation des cookies
 */
$.cookie('cookEmail', 'tintin@free.fr');
```

Autres exemples :

```
$.cookie('cookEmail', 'tintin@free.fr', {expires : 7});
```

```
$.cookie('cookEmail', 'tintin@free.fr', {expires : 7, path : '/'});
```

Pour récupérer la valeur d'un cookie

```
$.cookie('nomDuCookie')
```

```
console.log($.cookie('cookEmail'));
```

Pour supprimer un cookie

```
$.cookie('nomDuCookie', '', {expires: -1});
```

ou

```
$.removeCookie('nomDuCookie');
```

```
/*  
 * Suppression des cookies  
 * -1 = -1 jour  
 */  
$.cookie('cookEmail', '', {expires: -1});
```

ou

```
$.removeCookie('cookEmail');
```

La tentative d'affichage du contenu du cookie supprimé renvoie undefined.

Remarque : quoique ce soit souvent écrit sur internet `$.cookie('nomDuCookie', null);` ne supprime pas la cookie mais lui affecte la valeur null et donc la tentative d'affichage du contenu du cookie renvoie null.

1.8 - GÉRER LE STORAGE

Téléchargez le plugin : <https://github.com/andris9/jStorage/blob/master/jstorage.js>

Documentation : <https://github.com/andris9/jStorage/blob/master/README.md>

jStorage permet de stocker des chaînes, des nombres, des objets JavaScript, tableaux et même des nœuds XML ce qui est comparable à un stockage JSON.
Les valeurs sont stockées dans le localStorage.

	Elements	Network	Sources	Timeline	Profiles	Resources	Audits	Console
▶	Frames		Key		Value			
	Web SQL		jStorage		{"__jstorage_meta":{"CRC32":{},"TTL":{}}}			
	IndexedDB		jStorage_update		1415163963428			
▼	Local Storage							
	http://localhost:8383							
▶	Session Storage							
▶	Cookies							

Usage :

Copiez le fichier jstorage.js dans l'arborescence du site.

Incluez jQuery : `<script src="../../jquery/jquery.js"></script>`

Incluez le plugin : `<script src="../../jquery/PIStorage/jstorage.js"></script>`

Codez!

Pour créer une variable

```
$.jStorage.set('nomDeLaVariable', 'valeur');
```

```
| $.jStorage.set('email', 'tintin@gmail.com');
```

Pour récupérer le contenu d'une variable

```
console.log($.jStorage.get('nomDeLaVariable'));
```

```
| console.log($.jStorage.get('email'));
```

Pour supprimer une variable

```
$.jStorage.deleteKey('nomDeLaVariable');
```

```
| $.jStorage.deleteKey('email');
```


Divers

Cette méthode permet de savoir si l'API Storage est implémentée par le navigateur.

```
$.jStorage.storageAvailable();
```

Cette méthode permet de connaître la taille du storage utilisé (en octets).

```
$.jStorage.storageSize();
```

Voir aussi simpleStorage (du même auteur) si vous n'avez pas à gérer IE ≤ 7 .

<https://github.com/andris9/simpleStorage/blob/master/simpleStorage.js>

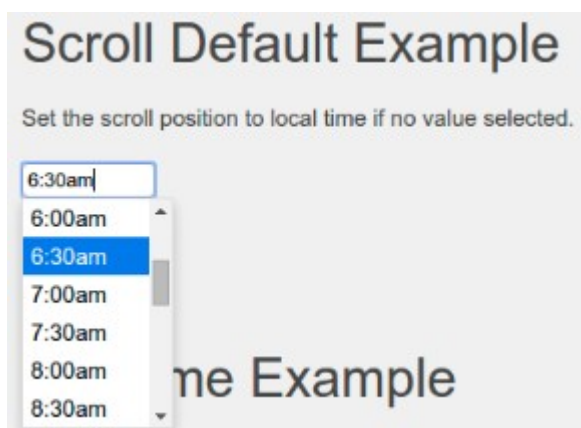
<https://github.com/andris9/simpleStorage/blob/master/README.md>

1.9 - TIMEPICKERS

jQuery UI propose un DatePicker mais pas de TimePicker. Il faut donc se tourner vers un Plug-In.
En voici 2.

jquery.timepicker

<http://jonthornton.github.io/jquery-timepicker/>



Exemples de code :

```
| $('#scrollDefaultExample').timepicker({ 'scrollDefault': 'now' });
```

```
| $('#setTimeExample').timepicker();  
| $('#setTimeButton').on('click', function () {  
|     $('#setTimeExample').timepicker('setTime', new Date());  
| });
```

```
| $('#disableTimeRangesExample').timepicker({  
|     'disableTimeRanges': [  
|         ['1am', '2am'],  
|         ['3am', '4:01am']  
|     ]  
| });
```

jQuery UI Timepicker

<https://fgelinas.com/code/timepicker/>

	Hour						Minute		
AM	00	01	02	03	04	05	00	05	10
	06	07	08	09	10	11	15	20	25
PM	12	13	14	15	16	17	30	35	40
	18	19	20	21	22	23	45	50	55

Exemples de code :

```
| $('#timepicker').timepicker();
```

```
| $('#timepicker').timepicker({  
|     showPeriod: true,  
|     showLeadingZero: true  
| });
```

```
| $('#timepicke').timepicker({  
|     hours: { starts: 6, ends: 19 },  
|     minutes: { interval: 15 },  
|     rows: 3  
|     showPeriodLabels: true,  
|     minuteText: 'Min'  
| });
```

Voir aussi jQuery.ptTimeSelect à l'URL suivante :

<http://pttimeselect.sourceforge.net/example/index.html>

Default call with no params defined. [\[view source\]](#)

Start:


End:

Option:

Example

Call with
Start Time

End Time

1 : 00 AM

Hour			Minutes		
AM	PM		00	05	10
1	2	3	15	20	25
4	5	6	30	35	40
7	8	9	45	50	55
10	11	12			

Set

CHAPITRE 2 - ANNEXES

2.1 - FICHIERS RESSOURCES

2.1.1 - *cours.ini*

```
; cours.ini
[section_connexion]
protocole=mysql
serveur=127.0.0.1
port=3306
bd=cours
ut=root
mdp=
```

2.1.2 - Connexion.php

```
<?php

class Connexion {

    public static function getConnection($psCheminParametresConnexion) {

        $tProprietes = parse_ini_file($psCheminParametresConnexion);

        $lsProtocole = $tProprietes["protocole"];
        $lsServeur = $tProprietes["serveur"];
        $lsPort = $tProprietes["port"];
        $lsUT = $tProprietes["ut"];
        $lsMDP = $tProprietes["mdp"];
        $lsBD = $tProprietes["bd"];

        $lcnx = null;
        try {
            $lcnx = new
PDO("$lsProtocole:host=$lsServeur;port=$lsPort;dbname=$lsBD;", $lsUT,
$lsMDP);
            $lcnx->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
            $lcnx->setAttribute(PDO::ATTR_AUTOCOMMIT, FALSE);
            $lcnx->exec("SET NAMES 'UTF8'");
        } catch (Exception $ex) {
            $lcnx = null;
            echo $ex->getMessage();
        }
        return $lcnx;
    }

    public static function seDeconnecter(PDO &$pcnx) {
        $pcnx = null;
    }
}
```

2.1.3 - VillesDAO.php

```
<?php

class VillesDAO {

    public static function selectAll(PDO $pcnx) {

        $lboK = true;
        $tVilles = array();

        try {
            $lsqlSelect = "SELECT * FROM villes";
            $lrs = $pcnx->prepare($lsqlSelect);

            $lrs->execute();
            $lrs->setFetchMode(PDO::FETCH_NUM);
            while ($enr = $lrs->fetch()) {

                $tVille["cp"] = $enr[0];
                $tVille["nomVille"] = $enr[1];
                $tVille["site"] = $enr[2];
                $tVille["photo"] = $enr[3];
                $tVille["idPays"] = $enr[4];

                $tVilles[] = $tVille;
            }

            $lrs->closeCursor();
        } catch (PDOException $e) {
            //echo "<br>" . $e->getMessage() . "<br>";
            $lboK = false;

            $tVille["cp"] = "0";
            $tVille["nomVille"] = $e->getMessage();
            $tVille["site"] = "";
            $tVille["photo"] = "";
            $tVille["idPays"] = "";

            $tVilles[] = $tVille;
        }
        return $tVilles;
    }
}

?>
```


2.1.4 - VillesDAOTests.php

```
<?php

/*
 * VillesDAOTests.php
 */
require_once 'Connexion.php';
require_once 'VillesDAO.php';

$lcnx = Connexion::getConnexion("cours.ini");
$tVilles = VillesDAO::selectAll($lcnx);

foreach ($tVilles as $key => $ville) {
    echo $ville["cp"] . "-";
    echo $ville["nomVille"] . "<br>";
}

Connexion::seDeconnecter($lcnx);
?>
```

2.2 - CORRIGÉS DES EXERCICES

Cf jquery_exercices_corriges.odt.