jQuery : les plugins

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille Chercheur en programmation par contrainte (IA) Ingénieur en génie logiciel

elmouelhi.achref@gmail.com

Plan



Exemples de plugin

- Fancybox
- Websanova Color Picker
- FlexSlider
- GMap
- Autres plugins



- Drag and drop
- Resizable
- Dialog
- Onglet



Plugin jQuery

- jQuery est extensible en installant de nouveaux modules (plugins)
- plugin \Rightarrow : extension de jQuery \equiv ensemble de fichiers contenant :
 - code CSS
 - code JavaScript (ou jQuery)
 - images et ressources web
- proposant des fonctionnalités prêtes à l'emploi
- utilisable comme une méthode appelée à partir d'un objet jQuery : \$ (selecteur).plugin (paramètres)

Où trouver des plugins jQuery?

- En cherchant sur Google (ou autre moteur de recherche)
- En consultant certaines plusieurs plateformes
 - https://plugins.jquery.com/
 - http://www.bestjquery.com/
 - ...

→ ∃ →

Où trouver des plugins jQuery?

- En cherchant sur Google (ou autre moteur de recherche)
- En consultant certaines plusieurs plateformes
 - https://plugins.jquery.com/
 - http://www.bestjquery.com/
 - ...

Remarque

Il est possible de créer son propre plugin

イロト イポト イヨト イヨ



Comment utiliser un plugin? (très simple)

- Chaque plugin dispose d'une page web
- Chaque page web contient :
 - un lien pour télécharger les fichiers de ce plugin
 - une description (+ un exemple)
 - la compatibilité avec les navigateurs et les versions de jQuery



Fancybox

- https://fancyapps.com/fancybox/3/
- Permettant de
 - afficher des fenêtres type modale pour simuler l'ouverture de popups
 - afficher une image en taille réelle au clic sur une miniature.

→ ∃ →

Pour utiliser Fancybox, il faut référencer un fichier CSS et un deuxième JS

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery plugin</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/qh/</pre>
    fancyapps/fancybox@3.5.7/dist/jquery.fancybox.min.css" />
  <style>
    img {
      width: 200px;
      height:200px;
  </style>
</head>
<body>
  <script src="//chemin/vers/jquery.js"></script>
  <script src="https://cdn.jsdelivr.net/gh/fancyapps/fancybox@3</pre>
     .5.7/dist/jquery.fancybox.min.js"></script>
</body>
</html>
```

Plaçons nos images cliquables dans une balise ${\tt div}$ ayant la classe ${\tt imglist}$

```
<div class="imglist">
```


</<mark>a</mark>>

-
 -



Websanova Color Picker

- http://wcolorpicker.websanova.com/
- disposant d'un panneau de couleur
- offrant un sélecteur de couleur

< 47 ▶

Pour utiliser Websanova Color Picker, il faut télécharger puis référencer un fichier CSS et un deuxième JS. La zone de saisie permettra de récupérer le code hexadécimal de la couleur sélectionnée

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Plugin</title>
  <link rel="Stylesheet" type="text/css" href="./wColorPicker</pre>
    -2.1.7/wColorPicker.min.css" />
</head>
<body>
  <div id="wcp1">
    <input id="wcp-input" type="text"/><br/>
  </div>
  <script src="//chemin/vers/jquery.js"></script>
  <script src="./wColorPicker-2.1.7/wColorPicker.min.js"><///>
    script>
  <script src="file.js"></script>
</body>
</html>
```

Ce qu'il faut mettre dans file.js pour pouvoir modifier la couleur de fond et la valeur de la zone texte en fonction de la couleur choisie

```
$(function() {
  $("#wcp1").wColorPicker({
      initColor: '#ccf',
      onSelect: function(color) {
        $('body').css('background', color);
      },
      onMouseover: function(color) {
        $('#wcp-input').css('background', color).val
           (color);
      }
  });
});
```

< 口 > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >



FlexSlider

- http://www.woothemes.com/flexslider
- Permettant d'afficher un slider responsive pour faire défiler des images avec différentes options

Pour utiliser FlexSlider, il faut télécharger puis référencer un fichier CSS et un deuxième JS et enfin préparer la liste d'images

```
<!DOCTYPE html>
<html>
<head>
  <title>iOuerv Plugin</title>
  <link rel="Stylesheet" href="./FlexSlider-master/flexslider.css" />
</head>
<body>
  <div class="flexslider">
    <img src="http://www.lsis.org/elmouelhia/images/SanFrancisco"</pre>
        .ipg" /> 
      <img src="http://www.lsis.org/elmouelhia/images/Bruxelles.</pre>
        ipq" /> 
      <img src="http://www.lsis.org/elmouelhia/images/Seattle.jpg"</pre>
         /> 
    </div>
  <script src="//chemin/vers/jquery.js"></script>
  <script src="./FlexSlider-master/jquery.flexslider.js"></script></script></script></script></script>
  <script src="file.js"></script>
</body>
</html>
```

ヘロン 人間 とくほ とくほ とう



Ce qu'il faut mettre dans ${\tt file.js}$ pour activer le slider

```
$(window).load(function() {
   $('.flexslider').flexslider();
});
```

jQuery

GMap

- https://hpneo.dev/gmaps/
- permettant d'afficher
 - une carte géographique selon des coordonnées GPS
 - un pointeur sur une carte
 - un itinéraire

A B F A B F

A D M A A A M M

jQuery

GMap

- https://hpneo.dev/gmaps/
- permettant d'afficher
 - une carte géographique selon des coordonnées GPS
 - un pointeur sur une carte
 - un itinéraire

Remarque

Depuis 2017, les cartes **Google** fonctionnent avec une clé (API KEY). Cette clé doit être associée à un compte de facturation (et donc une carte bancaire).

jQuery

Si on veut afficher la position sur une carte Google Maps?

- Créer un projet Google Developers Console dans https://console.developers.google.com/flows /enableapi?apiid=maps_backend&keyType=CLIENT_SIDE &reusekey=true&hl=fr&pli=1
- Utiliser la même page pour créer une clé (sans aucune restriction) sur les protocoles)
- Activer la clé en cherchant (dans la même interface en utilisant la zone de recherche) Geocoding et Geolocating et en cliquant chaque fois sur Activer

Image: A matrix and a matrix

Copier la clé pour l'utiliser

jQuery

Pour utiliser Gmaps, il faut le télécharger puis le référencer avec API Maps de google contenant la clé générée précédemment

```
<!DOCTYPE html>
<html>
<head>
  <title>iOuerv Plugin</title>
  <style>
    div {
      width:500px;
      height:500px;
    ł
  </style>
</head>
<body>
  <div id="map" ></div>
  <script src="//chemin/vers/jquery.js"></script>
  <script src="https://maps.googleapis.com/maps/api/js?key=coller-la-cl</pre>
    é-ici"></script>
  <script src="./qmaps-master/qmaps.js"></script>
  <script src="file.js"></script>
</body>
</html>
```

Pour afficher la carte dans div#map, il faut préciser les longitude et latitude dans un objet GMaps dans file.js

```
$ (document).ready(function() {
   var map = new GMaps({
      div: '#map',
      lat: 51.5073346,
      lng: -0.1276831,
   });
});
```

< ロ > < 同 > < 回 > < 回 >

Pour afficher la carte dans div#map, il faut préciser les longitude et latitude dans un objet GMaps dans file.js

```
$ (document).ready(function() {
   var map = new GMaps({
      div: '#map',
      lat: 51.5073346,
      lng: -0.1276831,
   });
});
```

Remarque

On peut aussi ajouter autres champs comme zoom, title... dans l'objet GMaps.

iQuery

Pour marguer un pointeur sur la carte, on utilise la méthode addMark ()

```
map.addMarker({
  lat: 51.518856,
  lng: -0.1263371,
  title: 'British Museum',
  infoWindow: {
    content:
      'The British Museum is a famous museum in London.'
    }
});
```

< ロ > < 同 > < 回 > < 回 >

iQuery

Pour marguer un pointeur sur la carte, on utilise la méthode addMark ()

```
map.addMarker({
  lat: 51.518856,
  lng: -0.1263371,
  title: 'British Museum',
  infoWindow: {
    content:
      'The British Museum is a famous museum in London.'
    }
});
```

Remarque

On peut évidemment définir plusieurs pointeurs.

(I) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1)) < ((1))

jQuery

Pour dessiner un itinéraire, on utilise la méthode drawRoute ()

```
map.drawRoute({
  origin: [latitude1, longitude1],
  destination: [latitude2, longitude2],
  travelMode: 'driving',
  strokeColor: 'red',
  strokeWeight: 6
});
```

æ

jQuery

Pour dessiner un itinéraire, on utilise la méthode drawRoute ()

```
map.drawRoute({
  origin: [latitude1, longitude1],
 destination: [latitude2, longitude2],
 travelMode: 'driving',
  strokeColor: 'red',
  strokeWeight: 6
});
```

Pour afficher la position du visiteur (géo-localisation)

```
GMaps.geolocate({
  success: function(position) {
    map.setCenter(position.coords.latitude, position.coords.longitude);
    map.addMarker({
      lat: position.coords.latitude,
      lng: position.coords.longitude,
        title: 'You are here.',
    });
  }
});
```

Autres plugins

- **jCarousel** (http://sorgalla.com/jcarousel/): Galerie d'images
- Uploadify (http://www.uploadify.com): Formulaire d'upload avec indicateur de progression
- Noty (http://ned.im/noty/#/about) : Popup de notifications
- jQuery Validator (https://jqueryvalidation.org/): Validation de formulaires web
- DataTables (https://datatables.net/) : Affichage et tri de données complexes
- Isotope (http://isotope.metafizzy.co) : Pour filtrer et trier des éléments HTML
- Masonry (http://masonry.desandro.com) : Pour réorganiser et d'imbriquer des éléments HTML de différentes hauteurs

jQuery UI (User Interface)

- https://jqueryui.com/
- un plugin jQuery très développé regroupant plusieurs plugins accessibles via un seul fichier JS
- composé de plusieurs fichiers CSS et JS
- permettant de
 - simplifier certaines tâches : drag and drop, redimensionnement...
 - améliorer certains visuels
 - gérer le DOM

A (10) > A (10) > A (10)



Comment intégrer jQuery UI dans notre fichier?

Comme pour jQuery

- soit en téléchargeant puis en faisant référence aux fichiers nécessaires de jQuery UI
- soit en utilisant une CDN



En se rendant sur (https://jqueryui.com/download/)

on remarque que

- on peut choisir un thème (il y en a plusieurs et on peut créer un thème personnalisé)
- on peut sélectionner quelques fonctionnalités selon le besoin



En se rendant sur (https://jqueryui.com/download/)

on remarque que

- on peut choisir un thème (il y en a plusieurs et on peut créer un thème personnalisé)
- on peut sélectionner quelques fonctionnalités selon le besoin

Remarque

Dans les deux cas, il faut référencer jQuery UI après jQuery

jQuery UI

jQuery

Contenu de la page index.html qui fait référence à jQuery et jQuery UI

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery UI</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui</pre>
    /1.12.1/themes/smoothness/jquery-ui.css" />
</head>
<body>
  <script src="//chemin/vers/jquery.js"></script>
  <script src="http://code.jquery.com/ui/1.12.1/jquery-ui.js"><</pre>
    /script>
  <script src="file.js"></script>
</body>
</html>
```

э.

A B A B A B A
 A B A
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 B
 A
 A
 A

jQuery UI

jQuery

Contenu de la page index.html qui fait référence à jQuery et jQuery UI

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery UI</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui</pre>
    /1.12.1/themes/smoothness/jquery-ui.css" />
</head>
<body>
  <script src="//chemin/vers/jquery.js"></script>
  <script src="http://code.jquery.com/ui/1.12.1/jquery-ui.js"><</pre>
    /script>
  <script src="file.js"></script>
</body>
</html>
```

Remarque

Ici on utilise le thème smoothness

Drag and drop (glisser-déposer)

- un concept permettant de déplacer des éléments d'un conteneur vers un autre
- existant depuis quelques temps dans les OS (déplacer un fichier en le glissant avec la souris d'un dossier vers un autre)
- existant dans le web bien avant le HTML5 mais très compliqué
- simplifié avec API HTML5
- encore plus simple avec jQuery UI
- constitué de deux plugins : draggable et droppable

• • • • • • • • • • • • •

Considérons les deux blocs element1 et element2 situés dans le bloc conteneur

< 口 > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

Considérons les deux blocs element1 et element2 situés dans le bloc conteneur

Le CSS correspondant

```
#conteneur {
   width:500px;
   height:500px;
}
.element {
   width:100px;
   height:100px;
}
```

< 口 > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

Considérons les deux blocs element1 et element2 situés dans le bloc conteneur

Le CSS correspondant

```
#conteneur {
   width:500px;
   height:500px;
}
.element {
   width:100px;
   height:100px;
}
```

Objectif

Déplacer les éléments dans le conteneur
Pour rendre les éléments déplaçables, il faut appeler le plugin

```
draggable dans file.js
```

```
$ (document) .ready(function() {
   $('.element').draggable();
});
```

< ロ > < 同 > < 回 > < 回 >

Pour rendre les éléments déplaçables, il faut appeler le plugin

```
draggable dans file.js
```

```
$ (document) .ready(function() {
    $('.element').draggable();
});
```

Pour rendre les éléments déplaçables selon un seul axe

```
$ (document).ready(function() {
    $('.element:first').draggable({
        axis : 'x'
    });
    $('.element:last').draggable({
        axis : 'y'
    });
});
```

э

ヘロト 人間 ト イヨト イヨト

Pour rendre les éléments déplaçables seulement dans le conteneur (en utilisant un sélecteur CSS, ici #conteneur)

```
$(document).ready(function(){
   $('.element:first').draggable({
    axis : 'x',
    containment : '#conteneur'
  });
  $('.element:last').draggable({
    axis : 'y',
    containment : '#conteneur'
  });
});
```

On peut aussi utiliser la constante parent pour désigner le conteneur

```
$(document).ready(function(){
   $('.element:first').draggable({
    axis : 'x',
    containment : 'parent'
   });
   $('.element:last').draggable({
    axis : 'y',
    containment : 'parent'
   });
});
```

Pour ajouter un effet de magnétisme à un élément (dès qu'il approche un deuxième élément, il se met sur le même niveau), on ajoute la propriété snap

```
$(document).ready(function(){
    $('.element:first').draggable({
        containment : '#conteneur',
        snap : true
    });
    $('.element:last').draggable({
        containment : '#conteneur',
        snap : true
    });
});
```

Pour remettre un élément à sa place initial dès qu'il est lâché

```
$(document).ready(function() {
  $('.element:first').draggable({
    containment : '#conteneur',
    snap : true
  });
  $('.element:last').draggable({
    containment : '#conteneur',
    snap : true,
    revert : true
  });
});
```



Il est possible de définir un pas de déplacement horizontal et vertical

```
$ (document).ready(function() {
   $('.element').draggable({
      grid : [50 , 50]
   });
});
```

A B b 4 B b



Quatre évènements liés aux déplacements

- create : se déclenche lorsque l'élément est créé
- start : se déclenche lorsqu'on démarre le déplacement
- drag : se déclenche pendant le déplacement
- stop : se déclenche lorsqu'on arrête le déplacement

Exemple

```
$(document).ready(function(){
  $(".element").draggable({
    create: function(event,ui) {
      console.log("create");
    },
    start: function(event,ui) {
      console.log("start");
    },
    stop: function(event,ui) {
      console.log("stop");
    },
    drag: function(event,ui) {
      console.log("drag");
    },
  });
});
```

Pour désactiver le déplacement, on considère l'exemple suivant

Pour désactiver le déplacement, on considère l'exemple suivant

Le code jQuery permettant d'activer et désactiver ce plugin

```
$ (document) .ready (function () {
    $ (' #start').click (function () {
        $ ('div').draggable ();
    });
    $ (' #destroy').click (function () {
        $ ('div').draggable ('destroy');
    });
});
```

Considérons les trois blocs drag1, drag2 et drop

```
<div id=drag1 style="background-color:teal" class=element>drag1</div>
<div id=drag2 style="background-color:gold" class=element>drag2</div>
<div id="drop" style="background-color:pink">drop</div></div></div>
```

Considérons les trois blocs drag1, drag2 et drop

```
<div id=drag1 style="background-color:teal" class=element>drag1</div>
<div id=drag2 style="background-color:gold" class=element>drag2</div>
<div id="drop" style="background-color:pink">drop</div></div></div></div>
```

Le CSS correspondant

```
#drag1, #drag2 {
   width:100px;
   height:100px;
}
#drop {
   width:200px;
   height:200px;
}
```

Considérons les trois blocs drag1, drag2 et drop

```
<div id=drag1 style="background-color:teal" class=element>drag1</div>
<div id=drag2 style="background-color:gold" class=element>drag2</div>
<div id="drop" style="background-color:pink">drop</div></div></div>
```

Le CSS correspondant

```
#drag1, #drag2 {
   width:100px;
   height:100px;
}
#drop {
   width:200px;
   height:200px;
}
```

Objectif

Déclencher un traitement en déposant un élément déplaçable dans une zone de dépôt



Pour déclarer une zone de dépôt, il suffit d'appeler le plugin droppable

```
$(document).ready(function() {
    $('#drag1,#drag2').draggable();
    $('#drop').droppable({
        drop : function() {
            alert('élément déposé');
        }
    });
});
```

- A B M A B M



Pour déclarer une zone de dépôt, on ajoute le paramètre accept

```
$ (document).ready(function() {
   $('#drag1,#drag2').draggable();
   $('#drop').droppable({
      accept : '#drag1',
      drop : function() {
        alert('élément déposé');
      }
   });
});
```

- 3 →

Le paramètre revert

- la valeur valid permet à l'élément d'être renvoyé à sa place s'il est déposé dans une zone de dépôt après avoir exécutée la fonction drop
- la valeur invalid permet de garder l'élément s'il est déposé dans une zone de dépôt

イロト イヨト イヨト イヨト

Le paramètre revert

- la valeur valid permet à l'élément d'être renvoyé à sa place s'il est déposé dans une zone de dépôt après avoir exécutée la fonction drop
- Ia valeur invalid permet de garder l'élément s'il est déposé dans une zone de dépôt

Exemple

```
$(document).ready(function(){
    $('#drag1').draggable({
        revert : 'valid'
    });
    $('#drag2').draggable({
        revert : 'invalid'
    });
    $('#drop').droppable({
        drop : function(){
            alert('élément déposé');
        }
    });
});
```

Six évènements liés aux déplacements

- create : se déclenche lorsqu'une zone de dépôt est créée
- out : se déclenche lorsqu'un élément quitte la zone de dépôt
- over : se déclenche lorsqu'un élément survole la zone de dépôt
- drop : se déclenche lorsqu'un élément est déposé la zone de dépôt
- activate : se déclenche lorsqu'un élément commence à se déplacer
- deactivate : se déclenche lorsqu'un élément arrête de se déplacer

Exemple

```
$(document).ready(function(){
  $('#drag1,#drag2').draggable();
  $('#drop').droppable({
    drop : function() {
      console.log("drop");
    ł,
    create: function(event,ui) {
      console.log("create");
    ł,
    out: function(event,ui) {
      console.log("out");
    ł.
    over: function(event,ui) {
      console.log("over");
    },
    activate: function(event,ui) {
      console.log("activate");
    },
    deactivate: function(event,ui) {
      console.log("deactivate");
    ł
  });
});
```

э

イロト イポト イヨト イヨト



Resizable (redimensionnement)

- un concept permettant de modifier la taille d'un élément HTML
- réalisable avec le plugin Resizable

< 47 ▶

→ ∃ →

Considérons les deux blocs element1 et element2 situés dans le bloc conteneur

Considérons les deux blocs element1 et element2 situés dans le bloc conteneur

Le CSS correspondant

```
#conteneur {
   width: 500px;
   height: 500px;
}
.element {
   width: 100px;
   height: 100px;
}
```

Considérons les deux blocs element1 et element2 situés dans le bloc conteneur

Le CSS correspondant

```
#conteneur {
   width: 500px;
   height: 500px;
}
.element {
   width: 100px;
   height: 100px;
}
```

Objectif

Redimensionner des éléments HTML

Pour pouvoir modifier la dimension d'un élément, il faut appeler le plugin resizable dans file.js

```
$ (document).ready(function() {
   $("div").resizable();
});
```

< ロ > < 同 > < 回 > < 回 >

Pour pouvoir modifier la dimension d'un élément, il faut appeler le plugin resizable dans file.js

```
$ (document).ready(function() {
    $("div").resizable();
});
```

Pour préserver le ratio hauteur/largeur

```
$(document).ready(function(){
   $("div").resizable({
      aspectRatio : true
   });
});
```

э

イロト イポト イヨト イヨト

Pour définir une zone max de redimensionnement (qu'on ne peut dépasser)

```
$(document).ready(function(){
    $("#conteneur > div").resizable({
        containment: "#conteneur"
    });
});
```

Pour définir une zone max de redimensionnement (qu'on ne peut dépasser)

```
$(document).ready(function(){
    $("#conteneur > div").resizable({
        containment: "#conteneur"
    });
});
```

Pour définir un ratio hauteur/largeur (ici hauteur = largeur)

```
$(document).ready(function(){
    $("div").resizable({
        aspectRatio : true
    });
});
```

Pour définir une zone max de redimensionnement (qu'on ne peut dépasser)

```
$(document).ready(function(){
    $("#conteneur > div").resizable({
        containment: "#conteneur"
    });
});
```

Pour définir un ratio hauteur/largeur (ici hauteur = largeur)

```
$(document).ready(function(){
    $("div").resizable({
        aspectRatio : true
    });
});
```

Dans cet exemple : hauteur = 2 * largeur

```
$ (document).ready(function() {
    $("div").resizable({
        aspectRatio : 1/2
    });
});
```

On peut aussi définir les largeur et longueur max et min

```
$ (document).ready(function() {
    $("#conteneur > div").resizable({
        maxWidth: 500,
        maxHeight: 400,
        minHeight: 50,
        minWidth: 50
    });
});
```

< 回 > < 三 > < 三 >

Il est possible de définir une classe CSS

```
<style>
.maClasse {
    border : 1px solid black;
    background : #eee;
    opacity : 0.5;
  }
</style>
```

æ

<ロ><日><日><日</th>

Il est possible de définir une classe CSS

```
<style>
.maClasse {
    border : 1px solid black;
    background : #eee;
    opacity : 0.5;
  }
</style>
```

Et d'appeler cette classe CSS pendant le redimensionnement avec le paramètre helper

```
$(document).ready(function(){
   $("div").resizable({
       helper : 'maClasse'
   });
});
```



Il est possible d'ajouter un effet animation (À ne pas confondre avec la méthode animate () de jQuery)

```
$(document).ready(function(){
   $("div").resizable({
        helper : 'maClasse',
        animate : true
   });
});
```



Dialog (Boîte de dialogue), comment ça marche?

- on définit notre boîte avec son contenu
- on gère l'affichage avec le plugin dialog de jQuery UI

On définit notre boîte avec son contenu

```
<div id="boite" title="Ma première boîte de dialogue">
    Ma boîte de dialogue est, par définition, redimensionnable et
    déplaçable.
</div>
```

æ

・ロト ・ 四ト ・ ヨト ・ ヨト

On définit notre boîte avec son contenu

```
<div id="boite" title="Ma première boîte de dialogue">
    Ma boîte de dialogue est, par définition, redimensionnable et
    déplaçable.
</div>
```

Pour l'afficher

```
$ (document).ready(function() {
    $("#boite").dialog();
});
```

э
On définit notre boîte avec son contenu

```
<div id="boite" title="Ma première boîte de dialogue">
    Ma boîte de dialogue est, par définition, redimensionnable et
    déplaçable.
</div>
```

Pour l'afficher

```
$ (document).ready(function() {
    $("#boite").dialog();
});
```

Constats

- Le titre de la boîte n'est affiché qu'en partie
- La boîte de dialogue s'ouvre au démarrage de la page

Pour modifier la dimension de la boîte

```
$(document).ready(function(){
    $("#boite").dialog({
        width: 400,
        height: 120
    });
});
```

크

< 口 > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

iQuery

Pour modifier la dimension de la boîte

```
$(document).ready(function(){
  $("#boite").dialog({
    width: 400,
    height: 120
  });
});
```

Pour ne pas ouvrir la boîte au chargement de la page

```
$(document).ready(function(){
  $("#boite").dialog({
    width: 400,
    height: 120,
    autoOpen: false
  });
});
```

On ajoute un bouton pour afficher la boîte

<button id=showDialog> afficher la boîte </button>

★ ∃ > < ∃ >

On ajoute un bouton pour afficher la boîte

<button id=showDialog> afficher la boîte </button>

Le code associé au bouton pour afficher la boîte

```
$(document).ready(function(){
    $("#boite").dialog({
        width: 400,
        height: 120,
        autoOpen: false
    });
    $('#showDialog').click(function(){
        $('#boite').dialog('open');
    });
});
```

On peut associer un ensemble de boutons à notre boîte de dialogue

```
$(document).ready(function(){
  $("#boite").dialog({
    width: 400,
    height: 120,
    autoOpen: false,
    buttons: {
      "Oui": function() {
        $('body').css('background', 'blue');
        $(this).dialog("close");
      },
      "Non": function() {
        $('body').css('background', 'red');
        $(this).dialog("close");
      }
    }
  });
  $('#showDialog').click(function() {
    $('#boite').dialog('open');
  });
});
```

ヘロマ ヘビマ ヘビマ イロマ

On peut bloquer l'accès au reste de la page tant que la boîte est affichée

```
$(document).ready(function(){
  $("#boite").dialog({
    width: 400,
    height: 120,
    autoOpen: false,
    modal: true
    buttons: {
      "Oui": function() {
        $('body').css('background', 'blue');
        $(this).dialog("close");
      },
      "Non": function() {
        $('body').css('background', 'red');
        $(this).dialog("close");
      ł
    }
  });
  $('#showDialog').click(function() {
    $('#boite').dialog('open');
  });
});
```



Tabs (Onglets), comment ça marche?

- on définit une liste de liens
- on gère l'affichage avec le plugin tabs de jQuery UI

On définit notre boîte avec son contenu

```
<div id="onglets">

        <a href="#onglet1">Titre onglet 1</a>
        <a href="#onglet2">Titre onglet 2</a>
        <a href="#onglet3">Titre onglet 3</a>
```

```
<div id="onglet1">
    Contenu onglet1
</div>
<div id="onglet2">
    Contenu onglet1
</div>
<div id="onglet3">
    Contenu onglet3
</div>
</div>
</div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></ti>
```

Le CSS correspondant

```
div > div {
    height: 300px;
}
#onglet1{
    background-color: teal;
}
#onglet2{
    background-color: gold;
}
#onglet3{
    background-color: red;
}
```

크

イロト イヨト イヨト イヨト

Le CSS correspondant

```
div > div {
    height: 300px;
}
#onglet1{
    background-color: teal;
}
#onglet2{
    background-color: gold;
}
#onglet3{
    background-color: red;
}
```

Pour afficher sous forme d'onglets

```
$ (document) . ready (function() {
    $ ('#onglets').tabs();
});
```

Pour créer un plugin

- déclarer une fonction
- associer cette fonction à l'objet jQuery (jQuery.fn)

< ロ > < 同 > < 回 > < 回 >

Pour créer un plugin

- déclarer une fonction
- associer cette fonction à l'objet jQuery (jQuery.fn)

Illustration

```
jQuery.fn.nomPlugin = function(param1, param2...)
{
    // Les instructions
}
```

イロト イポト イヨト イヨ

Dans un fichier mesPlugins.js, on définit le plugin suivant qui permettra d'afficher un texte comme un warning (en gras, couleur de fond rouge et couleur blanche pour le texte)

```
jQuery.fn.warningText = function(size) {
   this.each(function() {
     $(this).wrap('<b></b>')
        .css("background-color","red")
        .css("font-size", size+"px")
        .css("color","white");
   });
   return this;
};
```

< ロ > < 同 > < 回 > < 回 >

Dans un fichier mesPlugins.js, on définit le plugin suivant qui permettra d'afficher un texte comme un warning (en gras, couleur de fond rouge et couleur blanche pour le texte)

```
jQuery.fn.warningText = function(size) {
   this.each(function() {
     $(this).wrap('<b></b>')
        .css("background-color","red")
        .css("font-size", size+"px")
        .css("color","white");
   });
   return this;
};
```

Remarques

- this.each() permet de parcourir tous les éléments de la sélection
- return this retourne l'objet jQuery après modification pour permettre un éventuel chaînage

Préparons notre code HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery UI</title>
  k rel="stylesheet" href="http://code.jquery.com/ui
    /1.12.1/themes/smoothness/jquery-ui.css" />
</head>
<body>
  >
    Bonjour, ce n'est pas un <span class=warning> avertissement
       </span>.
    C'est juste pour vous montrer comment fonctionne mon
      nouveau plugin <<u>span</u> class=warning> avertissement </<u>span</u>
      >.
  <script src="//chemin/vers/jquery.js"></script>
  <script src="mesPlugins.js"></script>
  <script src="file.js"></script>
</body>
</html>
```



Utilisons notre plugin pour afficher quelques avertissements en rouge

```
$ (document).ready(function() {
    $ ('.warning').warningText(20);
});
```

.