

TD1-1 : JavaScript et jQuery

V4.0.0



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](#).

Document en ligne : www.mickael-martin-nevot.com

Travail : **binôme**

1 Généralités

N'oubliez pas de faire des recherches sur le Web à chaque fois que cela est nécessaire en prenant soin de vérifier que les informations trouvées soient correctes.

Vous visualiserez systématiquement votre travail dans différents navigateurs Web.

2 Formulaire HTML

Réalisez une page Web avec un formulaire HTML contenant les champs suivant :

Champ	Type
Identifiant	zone de saisie textuelle (<code><input type="text"/></code>)
Civilité (sexe)	boutons radio (<code><input type="radio"/></code>)
E-mail	zone de saisie textuelle (<code><input type="text"/></code>)
Mot de passe	mot de passe (<code><input type="password"/></code>)
Vérification de mot de passe	mot de passe (<code><input type="password"/></code>)
Téléphone	zone de saisie textuelle (<code><input type="text"/></code>)
Pays	liste de choix (<code><select></select></code>)
Conditions générales	case à cocher (<code><input type="checkbox"/></code>)
Bouton de soumission	soumission (<code><input type="submit"/></code>)

Remplissez la liste de choix du champ de pays avec quelques noms de pays.

Le bouton de soumission est pour le moment fictif (il ne fait aucune action).

N'oubliez pas d'identifier votre formulaire (par exemple : `id="MFC"`).

3 Formatage HTML

Nous souhaitons faire apparaître des messages informant de la non-conformité de la saisie des informations durant la saisie. Pour commencer, ajoutez des balises `` contenant des

éléments vides dans tous les paragraphes relatifs à des champs de saisie. Identifiez vos champs de saisie et vos balises `` :

```
<input id="input_email" type="text"/>
<span id="valid_email"></span>
<span id="valid_password"></span>
```

...

4 Interruption de traitement

Créez un répertoire `js` au même niveau que votre fichier HTML. Créez ensuite un nouveau fichier nommé `script.js`.

Liez `script.js` ainsi que jQuery à la page HTML. Cette bibliothèque permet de manipuler le DOM ou encore de configurer des gestionnaires d'évènements. Cependant, avant d'appliquer les fonctions jQuery aux éléments du document, il faut s'assurer que le DOM est complètement chargé :

```
$(document).ready(function(){
  ...
});
```

Puis, utilisez l'évènement DOM `submit` pour afficher une alerte JavaScript (`alert()`) au moment de la soumission du formulaire :

```
$("#MFC").submit(function() {
  alert("Soumission du formulaire demandée");
});
```

Il est possible d'annuler l'envoi du formulaire en retournant `false` dans l'évènement `submit`. Si les champs Identifiant, E-mail, Mot de passe et Vérification de mot de passe ne sont pas renseignés, annulez l'envoi du formulaire.

5 Messages d'information dynamiques

Il n'est pas du tout recommandé d'utiliser la méthode `alert()` (autrement que pour du débogage). Remplacez donc cet appel en affichant un message dans chaque balise `` en utilisant la fonction `text()` ou la fonction `html()` de jQuery :

```
$("#valid_email").text("Email invalide");
```

6 Validation de données « à la volée »

Affichez un message de non-conformité « à la volée » lorsqu'un champ est mal renseigné (validation de données) est particulièrement commode et évite un aller-retour client-serveur coûteux en bande passante. En outre, cela permet de cibler finement et immédiatement (sans attendre la soumission du formulaire) une erreur de saisie.

Une expression rationnelle est une chaîne de caractères (appelée motif) qui décrit un ensemble de chaînes de caractères possibles selon une syntaxe précise. C'est un outil particulièrement puissant pour la validation de données puisqu'il permet de décrire avec précision le format d'information

que doit recevoir un champ de formulaire. JavaScript prend en charge les expressions rationnelles¹.

Voici les conditions de validation proposées pour votre formulaire :

Champ	Expression rationnelle
Identifiant	<code>^[a-zA-Z0-9]{6,24}\$</code>
E-mail	<code>^[a-z0-9._-]+@[a-z0-9._-]{2,}\.[a-z]{2,4}\$</code>
Mot de passe	<code>^[a-zA-Z0-9]{1,30}\$</code>
Vérification de mot de passe	<code>^[a-zA-Z0-9]{1,30}\$</code>
Téléphone	<code>^0[1-9]([-.]?[0-9]{2}){4}\$</code>

Pour mettre en place la validation de données, utilisez l'évènement DOM de changement d'un élément de saisie d'un formulaire ([change](#)) sur l'ensemble des champs de saisie. Grâce aux expressions rationnelles, testez la valeur `val()` des champs de saisie avec la fonction `match()` qui renvoie `true` si la chaîne de caractère est conforme à l'expression rationnelle et affiche/cachez les messages d'information en conséquence :

```
$("#input_email").change(function(){
    if(!$("#input_email").val().match(/^[a-z0-9._-]+@[a-z0-9._-]{2,}\.[a-z]{2,4}$/)){
        $("#valid_email").text("Email invalide");
    } else {
        $("#valid_email").text("");
    }
});
```

Pour parfaire la validation de données des champs de votre formulaire, vérifiez également que les deux mots de passes saisis soient identiques lors de la soumission du formulaire.

¹ Pour plus d'informations sur les expressions rationnelles : http://fr.wikipedia.org/wiki/Expression_rationnelle