

TP - Contrôler les champs d'un formulaire en javascript

Présentation générale :

Vous disposez d'un formulaire déjà pré-stylisé. Il vous est demandé dans cet exercice de vérifier la valeur des champs du formulaire selon les règles suivantes :

Sexe :	<input type="radio"/> Homme <input type="radio"/> Femme	<div>Vous devez sélectionner votre sexe</div>
Nom :	<input type="text"/>	<div>Un nom ne peut pas faire moins de 2 caractères</div>
Prénom :	<input type="text"/>	<div>Un prénom ne peut pas faire moins de 2 caractères</div>
Âge :	<input type="text"/>	<div>L'âge doit être compris entre 5 et 140</div>
Pseudo :	<input type="text"/>	<div>Le pseudo ne peut pas faire moins de 4 caractères</div>
Mot de passe :	<input type="text"/>	<div>Le mot de passe ne doit pas faire moins de 6 caractères</div>
Mot de passe (confirmation) :	<input type="text"/>	<div>Le mot de passe de confirmation doit être identique à celui d'origine</div>
Pays :	<div>Sélectionnez votre pays de résidence ▼</div>	<div>Vous devez sélectionner votre pays de résidence</div>
<input type="checkbox"/> Je désire recevoir la newsletter chaque mois.		
<div>M'inscrire</div> <div>Réinitialiser le formulaire</div>		

Objectifs principaux du TP :

1 - Vous devrez vérifier la valeur des champs à deux moments :

- au moment de la soumission du formulaire
- au moment où vous modifierez la valeur d'un champ

2 - Afficher le message d'erreur correspondant à la saisie incorrecte

Si vous regardez la structure du fichier HTML, vous trouverez après chaque `<input>` un `` avec la classe "tooltip".

Ce span correspond au message d'erreur que vous devrez afficher en cas de saisie incorrecte pour chaque `<input>` ou `<select>`.

Que ce soit au moment de la soumission ou lorsque vous modifiez un champ, vous devrez donc récupérer ce `` d'erreur et l'afficher au bon moment, mais également le faire disparaître si la saisie est correcte.

TP - Contrôler les champs d'un formulaire en javascript

3 - Modifier la couleur de la bordure d'un champ suivant si il est valide ou non

Sexe : ☒ Homme ☐ Femme

Nom :

Prénom :

Âge :

Pseudo :

Mot de passe :

Mot de passe (confirmation) :

Pays :

☐ Je désire recevoir la newsletter chaque mois.

Vous devrez également modifier la couleur de la bordure de chaque champ modifiable en rouge si la saisie est incorrecte, et en vert si la saisie est correcte:

4 - Alerte de confirmation/erreur au moment de la soumission

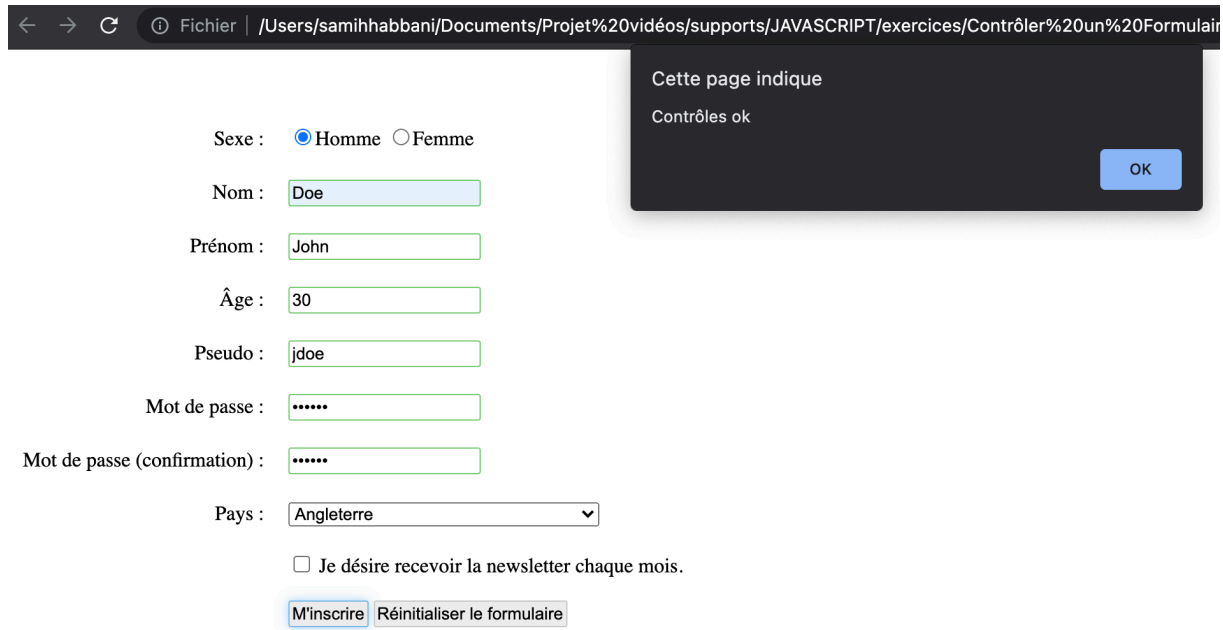
Lorsque vous soumettrez les données du formulaire vous devrez gérer deux scénarios différents :

- si toutes les saisies sont correctes, vous devrez afficher une alerte confirmant que les données saisies sont correctes

- si toutes les saisies ne sont pas correctes, vous devrez afficher une alerte indiquant à l'internaute que les données saisies sont incorrectes

A la suite de cette alerte, vous devrez évidemment afficher les différents messages d'erreur, ainsi que mettre en rouge les bordures des `<input>` dont la saisie sera incorrecte.
(idem si aucune valeur n'a été sélectionnée pour le `<select>`)

TP - Contrôler les champs d'un formulaire en javascript



The screenshot shows a web browser window with the address bar displaying the file path: `/Users/samihabbani/Documents/Projet%20vidéos/supports/JAVASCRIPT/exercices/Contrôler%20un%20Formulaire`. The page contains a registration form with the following fields and controls:

- Sexe :** Radio buttons for ☒ Homme and ☐ Femme.
- Nom :** Text input containing "Doe".
- Prénom :** Text input containing "John".
- Âge :** Text input containing "30".
- Pseudo :** Text input containing "jdoe".
- Mot de passe :** Password input (masked with dots).
- Mot de passe (confirmation) :** Password input (masked with dots).
- Pays :** Dropdown menu showing "Angleterre".
- ☐ Je désire recevoir la newsletter chaque mois.
- Buttons: and .

A dark overlay box in the top right corner displays the message: "Cette page indique Contrôles ok" with an .

5 - Gérer le reset du formulaire

Vous devrez également gérer le reset du formulaire (réinitialiser les différents champs à zéro) au moment du click sur l'input type reset.

Recommandations :

Si vous observez bien les consignes ci-dessus, nous avons des actions qui se répètent à différents moments :

- la vérification des saisies
- l'affichage des messages d'erreur
- la gestion de la couleur de la bordure des champs modifiables

Ces trois actions sont déclenchées au moment de la soumission, mais également lorsque l'on modifie la valeur d'un champ, c'est pourquoi nous conseillons de procéder aux étapes suivantes :

Organiser votre code:

Si vous observez le code html, vous avez un `<id>` pour chaque input.

Il serait peut-être intéressant de créer pour chaque `<id>` une fonction du même nom dans votre code Javascript.

Ainsi, si vous êtes en train de modifier par exemple la valeur de l'input avec `l'id="lastName"` vous n'aurez qu'à appeler la fonction `lastName` pour vérifier la valeur de l'input avec `id="lastName"`.

Il faudra donc récupérer l'id de l'input en question et appeler dynamiquement la fonction correspondant à l'id.

TP - Contrôler les champs d'un formulaire en javascript

Lorsque vous soumettrez vos données, vous n'aurez donc qu'à appeler l'ensemble des fonctions établies pour vérifier avant envoi des données les valeurs de tous les champs, y compris les `<input type="radio">` ou `<select>`.

Chaque fonction de contrôle s'occupera :

- de vérifier la valeur de l'input
- de récupérer le `` d'erreur correspond à l'input
- d'afficher/cacher le `` d'erreur en fonction de la valeur saisie
- de modifier la couleur de la bordure en fonction de la valeur saisie

Chaque fonction de contrôle devra également renvoyer `<false>` en cas de saisie incorrecte, et `<true>` si la saisie est correcte.

En effet, nous aurons besoin au moment de la soumission des données, de savoir si l'ensemble des fonctions de contrôle à renvoyer un contrôle positif ou négatif.

Une fois que vous aurez créé une fonction de contrôle pour chaque `<input>`, vous pourrez procéder aux étapes suivantes :

1 - Mettre en place une IIFE*, dans laquelle vous commencerez par faire disparaître au chargement de la page l'ensemble des messages d'erreur (``). Vous pouvez donc créer pour cela une fonction "désactiver les messages d'erreur" à l'extérieur de l'IIFE.

2 - Dans cette IIFE, vous allez capter l'événement "je modifie la valeur d'un champ" pour l'ensemble des `<input>` modifiables (`type="text"`), récupérer l'id de l'input en question, et appeler la fonction de contrôle correspondante.

3 - Toujours dans cette IIFE, vous allez capter l'événement "je soumetts le formulaire", annuler le rechargement de page par défaut, pour pouvoir vérifier l'ensemble des valeurs des différents `<input>` et `<select>` en appelant chaque fonction de contrôle :

- si toutes les fonctions de contrôle ont renvoyé un boolean à `<true>` alors cela signifiera que toutes les saisies sont correctes, vous pourrez donc afficher une alerte de confirmation avec un message "Contrôles ok"

- Par contre, si parmi ces fonctions de contrôle, une seule vous a renvoyé un boolean à `<false>` alors cela signifiera que toutes les saisies ne sont pas correctes, vous pourrez donc afficher une alerte indiquant un message à l'internaute : "Saisie(s) incorrecte(s)".

Dans ce deuxième scénario, évidemment, les fonctions de contrôle afficheront les messages d'erreurs correspondant aux mauvaises saisies, et mettront une bordure sur les `<input>` correspondants à ces erreurs de saisie.

TP - Contrôler les champs d'un formulaire en javascript

4 - Toujours dans cette IIFE, vous devrez capter l'événement "je fais un reset de mon formulaire", et vous devrez réinitialiser à du vide tous les `<input type="text">` et faire disparaître l'ensemble des messages d'erreur ainsi que les bordures rouges.

Les fonctions de contrôle sont à définir à l'extérieur de l'IIFE.

Organiser son code en objet?

Vous pouvez choisir de créer de simples fonctions, ou organiser votre code dans un objet.

L'objet représentera votre petite application javascript de contrôle des champs.

L'avantage d'organiser son code en objet, c'est qu'au moment de la soumission vous n'aurez pas besoin d'appeler toutes les fonctions à la suite :

```
lastName();  
firstName();  
country();  
....
```

Vous pourrez directement parcourir l'ensemble des méthodes de votre objet via une boucle for par exemple.

Dernières informations pratiques:

Seul le code javascript est à développer pour cet exercice, vous n'avez donc aucune modification à faire côté HTML et CSS.

Les consignes ci-dessus sont faites pour vous aider à mettre en place votre code autour d'une certaine logique. Evidemment il existe plusieurs codes différents pour une même solution, n'oubliez pas que le meilleur code est celui qui se répète le moins et qui est le plus simple à comprendre.

* les IIFE sont des fonctions qui s'auto exécutent et permettent de créer un scope local à vos variables, empêchant donc tout conflit avec un autre code javascript qui serait chargé.
Votre fonction IIFE sera donc exécutée au chargement de la page