Réalisation d'un formulaire avec vérifications instantanées

Trois étapes seront nécessaires :

1. Création du formulaire en HTML5 et CSS3
2. Validation en PHP
3. Validation en JS (Ajax)

Ressources :

- Le CM
- Documentation JQuery
- https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Comment_structurer_un_formulaire_HTML
- https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation_donnees_formulaire
- https://www.w3schools.com/php/php_form_url_email.asp

Cahier des charges :

- Optimisé pour l'UX
- Respectant les normes du W3C
- Vérifications instantanées
- Validation des données (le nom doit contenir en 3 et 15 lettres, n'être composé que de lettres / l'email doit être valide)

1) Réalisation de la page HTML du formulaire

Source de départ :

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Inscription à la newsletter</title>
</head>

<body>
    <form action="validation.php" method="post">
        <!-- ... -->
    </form>
</body>

<script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
<script>
    $(document).ready(function() {
      // ...
    });
</script>
</html>
```

Réalisez un formulaire d'inscription à une newsletter**, deux champs** sont demandés :

- Le prénom de la personne (afin de personnaliser les emails envoyés)
- Son adresse email

Un **fieldset** sera utilisé pour « entourer » le formulaire avec comme **légende** « Inscrivez-vous à notre newsletter ».

Un **label** accompagnera toujours le champ correspondant.

Les deux champs **seront obligatoires,** indiquez-le par un astérisque ou un « obligatoire » ou « requis ».

2) Création de la page validation.php

Cette page devra (pour le moment) vérifier vos données et afficher les messages d'erreur (avec echo) s'il y en a.
Il ne vous est pas demandé d'enregistrer les données en BDD.

```php
<?php
// Si il y a des données à traiter
if(isset($_POST)){
//      var_dump($_POST); // debug

}
?>
```

Trois tests seront réalisés :

- Y a-t-il un prénom ?
- Y a-t-il un email ?
- L'adresse email est-elle correcte ?

L'adresse email sera vérifiée grâce à la fonction *filter_var()* de php.

3) Validation en AJAX

a) Modification de notre page validation.php. Nous allons dès à présent stocker les erreurs dans un tableau **$erreurs[ ]** au lieu de les afficher.

Utilisation : $erreurs['NomDuChamp'] = 'Description de l'erreur' ;

```
if (le champ prénom est vide) {
        $erreurs['prenom'] = 'Prénom requis';
}
```

b) Envoi des erreurs encodées en JSON (validation.php)

```
if (il y a des erreurs) {
        echo json_encode($erreurs);
} else {
        echo json_encode(true); // Tout va bien
}
```

c) Pour lancer les vérifications à l'envoi du formulaire

```
$('form').submit(function(){

        //console.info('Envoi du formulaire') ;
        event.preventDefault(); // On empêche l'envoi du formulaire, donc le
        changement de page

        …
}
```

d) Envoi et vérification avec AJAX

```
function envoiFormulaire() {
    var data = {};

    $.ajax({
        dataType: 'json',
        type: 'POST',
        url: 'validation.php',
        data: {
            // Les infos en JSON ici
        },
        success: function(reponse) {
            //console.log(reponse)

            if (reponse === true) { // Si tout va bien
                // ...
            } else { // Si il y a des erreurs
                $.each(reponse, function(champ, erreur) {
                    // ...
                });
            }
        },
        error: function() {
            console.log('Problème de validation');
        }
    });

} // Fin envoiFormulaire
```

- Pour les data, nous utiliserons la fonction .val() pour récupérer les valeurs des champs.
- Si tout va bien on affiche un message **en vert**, sinon on affiche l'/les erreur(s) **en rouge** avec $(…).after(…) et si besoin $(…).addClass(…)
- Il faudra à la fin, supprimer les erreurs à chaque vérification avec .remove();

4) Optimisation du formulaire

Nous pouvons maintenant ajouter les attributs « required » et « pattern » à notre formulaire.

5) Aller plus loin…

Si vous avez terminé :

- Améliorez le design
- Rendre le tout responsive
- Ajouter des indicateurs d'aide au focus sur un champs (utilisez :focus)
- Utilisez la technique des « sliding label » (ressource)

Vous pouvez également transformer votre formulaire en formulaire de contact et utilisez un design original, une carte postale par exemple