Exercices AJAX : Série 02

1. Objectifs

- Apprendre à créer un système d'inscription

2. Exercice 01

1. Énoncé

> - *Avant de commencer gardez à l'esprit qu'une **vérification client** n'apporte aucune sécurité. C'est pour ça qu'elle doit impérativement être couplée à une vérification côté serveur.*
> - *Elle apporte une dimension visuelle et guide l'utilisateur, mais rien de plus.*

- Dans ce tp vous êtes appelés à réaliser les pages suivantes :
  - Une page **inscriptionUtilisateur.php** contenant le formulaire initial vide.
  - Une page **fonctionsAjax.js** contenant le code **AJAX** (du JavaScript hein ?)
  - Une page **verifier.php** interrogée par AJAX et qui renvoie les options disponibles pour un select donné.
- Ils doivent tous être placés dans le même répertoire, accessibles par votre serveur web.

- **Pour la page inscriptionUtilisateur.php.**

  - Au niveau de l'interface, vous êtes appelés de créer un formulaire bootstrap4 qui ressemble au schémas suivant:
  - Le formulaire doit avoir les attributs suivants : `id`="form_inscription" `onsubmit`="return false;">

apcpedagogie.com



- **input id="nomPrenom".**
  - L'utilisateur est appelé de saisir son nom et prénom, dès qu'il tape le premier caractère une zone de vérification sera affichée en bas comme le montre le schémas suivant:



  - Une fois l'utilisateur à saisie le sixième caractère la zone de texte se change comme le montre le schémas suivant:



  - Notant aussi que la zone de vérification sera masquée dans cet étape

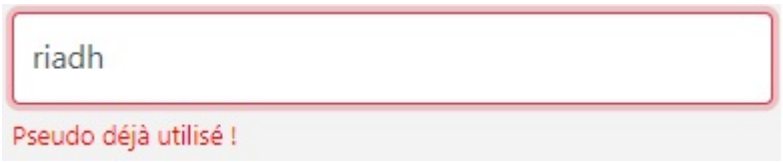
apcpedagogie.com

- Aussi dés que l'utilisateur saisie le vingtième caractère la zone de texte prend le style de départ et la zone de vérification s'affiche une autre fois avec couleur rouge.
- Enfin si l'utilisateur quitte la zone de texte une signe de validation sera affichée à droite de la zone comme le schémas suivant:

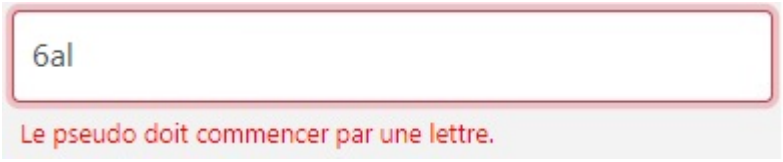


- **Contraintes:**
  - La zone nomPrénom doive être composée de deux mots au minimum.
  - Doit avoir un nombre de caractère minimum de 6 caractères et un maximum de 20 caractères.
  - L'utilisateur peut saisir des caractères spéciaux mais ils seront filtrés et ne seront pas comptabilisés comme des caractères dans la longueur du champ.

- **input id="pseudo".**
  - Le comportement général de cette zone de texte sera le même pour le "nomPrenom" précédemment décrit
  - L'utilisateur est appelé de saisir un nouveau pseudo, dès qu'il tape le dernier caractère d'un mot qui se trouve dans la base une zone de vérification sera affichée en bas comme le montre le schémas suivant:



  - Dans le cas ou l'utilisateur à commencer son pseudo par un chiffre,dès qu'il tape le troisième caractère un message sera affiché dans la zone de vérification, comme le montre le schémas suivant:



  - **Contraintes:**

        - La zone pseudo doive être composée d'un seul mot (espace non autorisé).
        - Doit avoir un nombre de caractère minimum de 3 caractères et un maximum de 12 caractères.
        - Le pseudo ne doit pas commencer avec un chiffre
        - L'utilisateur est appelé de saisir un pseudo qui n'est pas présent dans la base .
        - L'utilisateur peut saisir des caractères spéciaux mais ils seront filtrés et ne seront pas comptabilisés comme des caractères dans la longueur du champ.

    - **input id="email".**

        - Le comportement général de cette zone de texte email sera le même que les autres **"inputs"** précédemment décrits
        - **Contraintes:**
            - La validité d'émail est vérifiée par HTML5.
            - L'utilisateur est appelé de saisir un email qui n'est pas présent dans la base .

    - **input id="user-passwd".**

        - Le comportement général de cette zone de texte password sera le même que les autres **"inputs"** précédemment décrits
        - **Contraintes:**
            - Le mot de passe doit contenir entre 6 et 15 caractères, dont au moins 1 chiffre, un majuscule, un minuscule et un caractère spécial.
            - Le mot de passe ne sera pas visible au base de donnée.

    - **input id="user-repeatpass".**

        - Le comportement général de cette zone de texte repeatpassword sera le même que les autres **"inputs"** précédemment décrits

- **Pour la page fonctionsAjax.js.**

    - Ce fichier Javascript est chargé avec le premier fichier sur le client. Il effectue des requêtes asynchrones au script PHP `verifier.php` pour réaliser la

apcpedagogie.com

validation **AJAX**

- Dans ce fichier vous devez créer les fonctions en langage JavaScript qui vont vérifier en temps réel la disponibilité du pseudo et du mot de passe.

  Exemple de fonctions

- La fonction **verifier_pseudo**: c'est cette fonction qui est appelée à chaque frappe dans le champ "pseudo" de notre formulaire. Elle vérifie la longueur du pseudo tapé (si le pseudo fait moins de 3 ou plus de 12 caractères, un message d'erreur s'affiche) puis, si tout est bon, lance une requête Ajax qui se connectera a notre script PHP (vérifier.php) pour vérifier si le pseudo tapé est libre.
- La requête Ajax doit ressemble à l'exemple suivant:

```
function verifier_pseudo(){
                    $.ajax({
                        type: "post",
                        url:  "verifier.php",
                        data: {
                            'pseudo_check' : $("#pseudo").val()
                        },
                        success: function(data){
                                if(data == "success"){
                                    ..............
                                    return true;
                                } else {
                                    ..............
                                }
                                }
                    });
```

- Tous les contraintes serons vérifiées en temps réel en langage JavaScript

2. **Pour la page verifier.php.**

- Ce script PHP définit les fonctions de vérification, qui contient la logique métier et prend en charge les opérations de base de données nécessaires à la mise en oeuvre du rôle de celui-ci
- Cette page doit avoir les vérifications nécessaires coté serveur à savoir:
  - Tous les champs sont obligatoires
  - La vérification du nom prénom, le champs ne peut pas être vide
  - La vérification du pseudo, il ne doit pas déjà exister dans la base de données

apcpedagogie.com

- La vérification de l'email, l'adresse e-mail doit être donnée dans un format valide
- La vérification des mots de passe
- Le chiffrement de mot de passe
- Traitement de l'inscription
- Envoi d'un mail d'activation à l-utilisateur

3. **Créer la table utilisateurs comme le montre le schémas suivant.**

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Extra
1	cdeUser	int(11)			Non	*Aucun(e)*	AUTO_INCREMENT
2	prenomUtilisateur	varchar(200)	latin1_swedish_ci		Oui	*NULL*	
3	pseudo	varchar(25)	latin1_swedish_ci		Non	*Aucun(e)*	
4	emailUtilisateur	varchar(45)	latin1_swedish_ci		Oui	*NULL*	
5	motDePasse	varchar(255)	latin1_swedish_ci		Oui	*NULL*	
6	dateCreatUser	datetime			Oui	*NULL*	
7	ipUtilisateur	varchar(25)	latin1_swedish_ci		Non	*Aucun(e)*	
8	confirmkey	varchar(32)	latin1_swedish_ci		Non	*Aucun(e)*	
9	actif	tinyint(4)			Non	*Aucun(e)*	

4. **Solution**

- Essayez de faire l'exercice de votre côté avant de regarder la Solution !

Les méthodes get et post jQuery AJAX
Sommaire du cours AJAX

Sommaire du cours AJAX

Source : https://apcpedagogie.com/exercices-ajax-serie-02 · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com