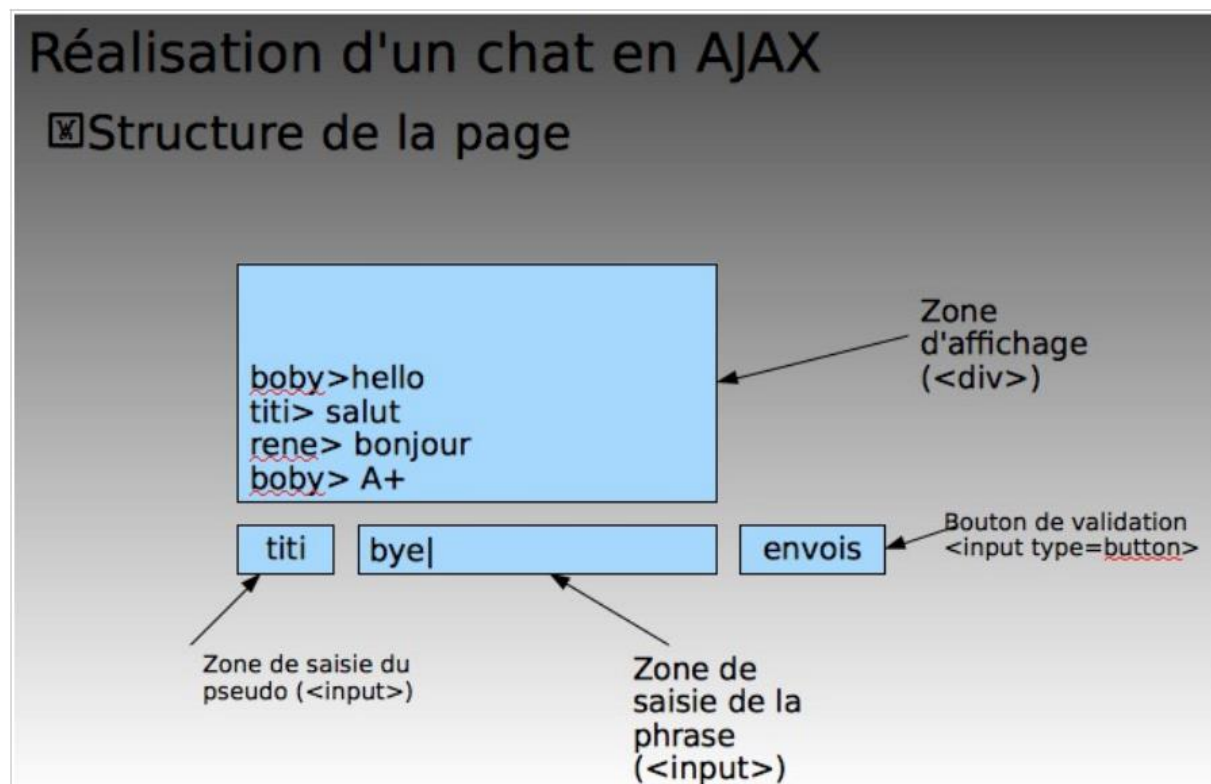


# Réalisation d'un chat interactif

Nous allons ici réaliser une application de chat interactif, avec sauvegarde des messages postés en base de données.



## Déroulement des séances :

- Séance 1 : Mise en place de l'API (BDD, dire.php et affiche.php)
- Séance 2 : Création du front (index.html avec JQuery et ajax)
- Séance 3 : Amélioration (design, appuie sur la touche entrée, protection...)

### 1) Préparation : Création de la table de données

- Créer une table **chat** (id, pseudo, message) dans votre base de données
- Remplir cette table avec une quinzaine d'enregistrement

### 2) Réalisation de la page d'enregistrement d'un nouveau message par un chatteur (dire.php?pseudo=toto&message=salut...)

Cette page devra enregistrer dans la base le message (et le pseudo) passée dans l'url ; aucun affichage de retour n'est nécessaire (ou alors juste un message pour le débogage)

Faites des essais pour plusieurs messages et vérifier leurs présences dans la BDD

### 3) Réalisation de la page d'affichage des 10 derniers messages (affiche.php)

Cette page devra afficher les 10 derniers messages tapés sur le chat sous la forme :

**pseudo > phrase** (+ <br/>)

bob > hello !

Toto > tu vas bi1 ?

Bob > oui merci

...

Faites des essais en saisissant des messages grâce à la page dire.php?pseudo=... et en les relisant avec la page affiche.php

## 4) Réalisation de la page html affichant l'interface du chat

Source de départ :

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Chat jQuery</title>
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
  <script>
    $(document).ready(function(){
      // Notre script pour afficher et envoyer les messages
    });
  </script>
</head>
<body>

  <div id="affiche">Chargement des messages...</div>

  <input type="text" id="pseudo" placeholder="Votre pseudo">
  <input type="text" id="message" placeholder="Votre message">
  <input type="submit" id="envoyer" value="Envoyer">

</body>
</html>
```

Lors du **clique** sur le bouton «Envoyer», générer l'appel à la ressource dire.php en utilisant JQuery : `$(...).click()` , `$(...).val()` et `$.ajax()`;

Faire une routine qui se lance régulièrement (toute les secondes par exemple), et qui vient récupérer l'affichage de la page affiche pour l'afficher dans la zone spécifique.  
Utiliser «setInterval» pour charger régulièrement les données de la page ;  
et `$.get(...).done(...)` pour écrire dans la zone d'affichage les 10 derniers messages (en «loadant» la page affiche.php)

Il faut rafraichir régulièrement la zone d'affichage des messages (toutes les secondes par exemple) utiliser simplement la fonction javascript setinterval :

SetInterval ( function() { // Do something every 5 seconds} , 5000 ) ;

## 5) Fonctionnalités avancées

- Afficher les messages dans le bon ordre (à faire dans le fichier affiche.php)
- Modifier l'ensemble pour que le message s'envoie lorsque l'on appuie sur «Entrée» plutôt que lors du clic sur le bouton. (pour cela utiliser `$(...).keydown(function(event) { ... })`)
- Vider la zone de saisie du message à chaque envoi (utiliser `$(...).val()` par exemple)
- Enregistrer et afficher l'heure des messages (dire et affiche.php)
- Protéger le système contre les injections SQL (à faire dans le fichier dire.php)
- Protéger le système contre l'envoi de message vide et le flood (à faire dans le fichier dire.php)
- Réaliser un parseur de tag (ou smiley) permettant de faire vibrer (voire d'émettre un son) la zone des messages affichées lorsque l'on tape «wizz» comme message
- Améliorer la mise en forme grâce à CSS3 en adoptant ce design par exemple (mettre le pseudo dans des bulles de couleurs à la place de la photo)

