

---

## TP1 : Gestions des évènements en JS

Ayoub KARINE (ayoub.karine@isen-ouest.yncrea.fr)

---

### Objectifs du TP

- Définition de la structure de l'application web en utilisant HTML et Bootstrap (BS)
- Affichage et masquage des éléments HTML en utilisant JavaScript

### Création de la partie statique du site web via HTML et BS

1. Téléchargez le dossier [tp1Input](#)
2. Supprimez la classe "d-none" de la section qui a l'id "authentication" et afficher le résultat sur le navigateur. Ce dernier doit être comme suit :

#### Chat

##### Authentication

Login :	Entrez votre login
Mot de passe :	Entrez votre mot de passe

☐ Rester connecté

Figure 1 : Exemple de visuel pour la partie authentication de l'application web

Nous allons maintenir la classe "d-none" dans la suite de TP.

3. Modifiez le fichier "index.html" afin d'avoir l'affichage ci-dessous tout en suivant les consignes suivantes :

## Chat

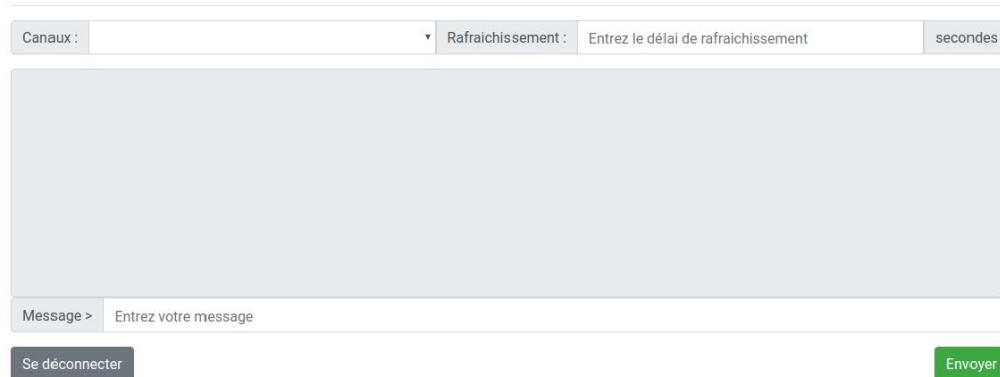


Figure 2 : Exemple de visuel pour la partie chat de l'application web

### Consignes :

- L'en-tête de la page devra contenir un titre de niveau 1 ainsi qu'un séparateur horizontal.
- Dans la balise section d'id "chat", ajoutez une zone de chat comportant un formulaire avec :
  - une liste déroulante pour le choix de salon :  
<https://getbootstrap.com/docs/4.5/components/forms/#form-controls>
  - un input pour le délai de rafraîchissement automatique  
<https://getbootstrap.com/docs/4.0/components/input-group/#basic-example>
  - une textarea non modifiable pour l'affichage des messages d'un salon donné. Pour ce faire, utilisez l'attribut readonly qui est booléen.
  - un input pour l'envoi d'un message dans le salon sélectionné
  - un bouton pour se déconnecter.
  - un bouton de type submit pour envoyer le message. Ce bouton doit être positionné tout à droite. Pour ce faire, utilisez la classe float-right :  
<https://getbootstrap.com/docs/4.5/utilities/float/#responsive>

## Ajout du dynamisme au site web via JS

Le but de cette partie de TP est, via le langage JS, d' :

- afficher la zone d'authentification (Figure 1) et masquer la zone de chat (Figure 2) en cliquant sur le bouton "se déconnecter"
- afficher la zone de chat (Figure 2) et masquer la zone d'authentification (Figure 1) en cliquant sur le bouton "se connecter"

1. Dans le dossier JS, créez un fichier "utils.js" dans lequel vous programmez une fonction permettant d'afficher ou de masquer un élément HTML dont l'id est passé en paramètre. Pour ce faire utiliser la fonction toggle de JS :

<https://developer.mozilla.org/fr/docs/Web/API/DOMTokenList/toggle>

2. Dans le dossier JS, créez un fichier "chat.js" dans lequel vous programmez :
  - a. deux gestionnaires d'événements pour gérer les clics sur les boutons "se connecter" et "se déconnecter".
  - b. une fonction "login(event)" qui va être déclenchée si on clique sur le bouton "se connecter". Elle permet d'afficher la zone de chat (Figure 2) et de masquer la zone de d'authentification (Figure 1). Par défaut, la soumission d'un formulaire lance le rechargement de la page. Ainsi, lors de l'ajout d'un gestionnaire pour l'événement submit, il est souvent nécessaire de désactiver ce rechargement. Pour cela, on utilise :

```
1 element.addEventListener('submit', myCallback);
2
3 function myCallback(event)
4 {
5     event.preventDefault();
6     ...
7 }
```

- c. une fonction "logout()" qui va être déclenchée si on clique sur le bouton "se déconnecter". Elle permet d'afficher la zone d'authentification (Figure 2) de de cacher la zone de chat (Figure 2). Afin d'éviter des déconnexions involontaires, vous afficherez une boîte de dialogue de confirmation avant de déconnecter l'utilisateur en utilisant la fonction "confirm" de JS :

<https://developer.mozilla.org/fr/docs/Web/API/Window/confirm>