

## TP 1 : Utilisation du Framework Symfony : PHP, TWIG, ...

### Consignes :



- Le compte-rendu doit être rendu **sous forme numérique** de préférence en format pdf.
- Le compte-rendu doit être soumis sur Moodle dans la section TP 1 puis dans évaluation.
- Le projet zippé contenant vos fichiers (twig.php) doit être rendu également.

### Exercice 1 : PHP, TWIG et Symfony

- 1-1** Nous vous demandons de créer un projet **Symfony** nommé **SymfonyTP1ex1**.  
Vous devez créer également le contrôleur **TP1Ex1Q1**  
Proposer deux méthodes permettant d'afficher à l'aide de ce contrôleur la date et l'heure du jour (l'une d'entre elle sera de n'utiliser qu'une page PHP).  
Vous expliquerez vos deux méthodes.
- 1-2** Maintenant, sur la page d'accueil de votre site, vous afficherez un formulaire permettant de saisir un nom et un prénom.  
Vous serez dirigé vers une autre page qui affichera le texte "Bonjour" suivi de vos prénom et nom.  
Vous afficherez à la suite également la date et l'heure du jour.

### Exercice 2 : Manipulation de JavaScript

- 2-1** Toujours sur Symfony, écrire une fonction JavaScript appelée lorsque l'utilisateur va cliquer sur le bouton.  
Au clic, afficher une boîte de dialogue qui demande l'année de naissance à l'utilisateur, puis vérifier qu'il s'agit bien d'une valeur numérique, et enfin, afficher l'âge calculé dans le document, à l'emplacement du paragraphe dont l'identifiant est **id\_Age**.
- 2-2** Créer un script JavaScript qui, au survol de la souris sur le texte d'un paragraphe, modifie la police pour qu'elle ait une taille de 30 pixels, que le texte s'affiche en italique et en rouge.  
Si la souris ne survole pas le paragraphe, le texte est remis tel quel.
- 2-3** Afficher sur une page html-twig une liste non numérotée affichant tous les BUT de l'IUT de Roanne.  
Vous commencerez par BUT RT et BUT GIM  
Créer une fonction JavaScript permettant d'ajouter un élément à la liste déjà existante.  
Créer une autre fonction JavaScript qui va permettre d'effacer le dernier élément de la liste.

### Exercice 3 : PDO et Faille de sécurité

- 3-1** Toujours à l'aide du framework **Symfony**, mais en créant une entité, créer une table **InformationsConnexions** ayant pour champs deux chaînes de caractères **Login** et **MotdePasse**.  
Pour l'instant, cette table ne contiendra qu'un utilisateur ayant pour login root et comme mot de passe root.
- 3-2** Créer une page de connexion qui contiendra un formulaire permettant de saisir comme informations le login et le mot de passe.  
Nous exécuterons directement les requêtes SQL à partir des champs du formulaire.  
Pour cela, dans vos fonctions écrites en PHP, vous utiliserez l'extension **PDO**, sans préparation de la requête.

Il faudra écrire dans vos fichiers PHP `use \PDO`

Vous pourrez vous aider du document en annexe nommé `PHP et SQL - Annexe.pdf`

Si l'authentification se déroule bien, vous serez redirigé sur une page qui saluera l'utilisateur.

**3-3** La méthode que nous venons de faire présente-t-elle une faille de sécurité.

Si la réponse est positive, montrez le avec votre formulaire et proposez une méthode permettant de résoudre ce problème.

**3-4** Au lieu de passer par une méthode `POST`, tester votre connexion avec la méthode `GET`.

Que pourriez-vous faire pour afficher un message compromettant ou pour récupérer les cookies d'une session ?

#### **Exercice 4 : PDO et Requête SQL**

**4-1** Toujours sur **Symfony**, créer une table **EURO** qui représente le taux de changes de plusieurs devises. Vous ne créez pas d'entité cette fois-ci, mais vous utiliserez l'extension **PDO**. Les trois champs seront l'**ID** (clé primaire), le nom de la monnaie **devise** et sa **valeur** en euros.

**4-2** Récupérer les informations du fichier **devise.txt** en le téléversant, puis en le lisant pour mettre des valeurs dans les différents champs de votre table.

**4-3** Vous afficherez du coup dans un tableau sur une autre page de votre site les différents taux de conversion présents dans votre table.