

## Exercices SIO

### Exercice 8 : Contenu protégé par mot de passe

Le but de cet exercice est de créer un contenu protégé par mot de passe. Pour cela, choisir un mot de passe (par exemple : kangourou). Créer un formulaire permettant de saisir un mot de passe. Le contenu protégé doit alors s'afficher uniquement si le mot de passe est correct. Dans le cas contraire, le formulaire doit de nouveau s'afficher.

### Exercice 9 : Calculatrice

Le but de cet exercice est de créer une calculatrice simple. L'utilisateur doit saisir deux nombres et choisir une opération parmi l'addition, la soustraction, la multiplication ou la division. Le résultat de l'opération doit alors s'afficher. Attention aux opérations invalides !!

Afin de faciliter la saisie par l'utilisateur, créer un formulaire contenant deux balise input de type text où l'utilisateur saisit directement les chiffres (par exemple : 17.5 ). Et des boutons radios pour l'opérateur (le traitement de la valeur sélectionnée devra impérativement être réalisé par un switch() ).

### Exercice 10 : Formulaire de contact

Il est intéressant qu'un utilisateur puisse laisser un message au webmaster ou à l'administrateur d'un site Web. L'idée consiste alors à créer un formulaire où l'utilisateur saisit ses coordonnées ainsi qu'un message. Lors de la soumission du formulaire, un message est alors automatiquement envoyé à la personne concernée. Le but de cet exercice est de créer un tel formulaire et le traitement PHP associé.

Question 10.1 : Créer un formulaire permettant à un utilisateur de saisir ses pseudo, numéro de téléphone et adresse e-mail. Vérifier, à l'aide des expressions régulières, que la saisie vérifie les caractéristiques suivantes :

- le pseudo est constitué de 4 à 8 lettres,
- le numéro de téléphone est constitué de 10 chiffres, le premier étant un 0, le suivant un nombre entre 1 et 6,
- l'adresse mail contient une seule arobase, se termine par ".fr", ".com" ou ".org", les autres caractères étant des lettres, des chiffres ou le symbole underscore. De plus, l'adresse doit contenir au moins 3 caractères avant l'arobase et entre 4 et 10 caractères entre l'arobase et la fin de la chaîne.

NB :

faites des recherches sur les expressions régulières  
`if (!preg_match('#^[a-zA-Z]{4,8}$#',$_POST['pseudo']))`  
`preg_match('#^0[1-6][0-9]{8}$#',$_POST['tel'])`

Le script PHP doit alors indiquer si la saisie est correcte et dans ce cas, afficher

## Exercices SIO

les informations saisies.

Question 10.2 : Rajouter dans le formulaire une zone de texte permettant à l'utilisateur de saisir le message qu'il souhaite laisser. Afin d'éviter l'envoi automatique de mails par des robots, poser également une question très simple du style : "Quel animal aboie ?" et ajouter un champ de texte pour que l'utilisateur puisse répondre à la question. Dans le traitement des données, vérifier que la réponse donnée par l'utilisateur est correcte. (Attention, pour un confort d'utilisation, les réponses "chien(s)", "le(s) chien(s)", "un chien" et "des chiens" doivent être valides.)

Si les données saisies par l'utilisateur sont valides, enregistrez le message dans un fichier (voir fopen).

Dans le cas où les informations ne sont pas correctes, afficher dans le formulaire les informations saisies par l'utilisateur.

Question 10.3 : Améliorer le script de manière à ce qu'un numéro de téléphone puisse aussi être saisi avec un espace ou un tiret tous les deux chiffres afin de permettre les numéros suivants :  
"0123456789", "01-23-45-67-89", "01 23 45 67 89".

Question 10.4 : Améliorer la présentation de la page Web à l'aide du CSS.

Exercice 12 : Nombre de secondes écoulées depuis le premier janvier 2014

Écrire un script PHP permettant à un utilisateur de saisir une date de l'année 2014 ainsi qu'une heure et affichant alors le nombre de secondes écoulées depuis le premier janvier de cette année.

Séparer le code en deux fichiers afin de regrouper au maximum le code XHTML d'une part et le code PHP d'autre part.

NB : vérifiez bien les valeurs saisies ( exemple les heures entre 0 et 23, mois entre 0 et 12, attention au mois de février !!)

### Exercice 13 : Jeu Devine Chiffre

Le jeu Devine chiffre se joue à un seul joueur. L'ordinateur choisit un nombre entier aléatoire compris entre 1 et 100 inclus. Le joueur a 7 tentatives pour trouver ce nombre.

À chaque tentative, si le nombre entré par l'utilisateur est différent du nombre aléatoire, le programme indique si le nombre saisi est plus petit ou plus grand que le nombre aléatoire.

Remarque : Le tirage d'un nombre aléatoire entre 1 et 100 se fait en PHP grâce à l'instruction `$nb = mt_rand(1,100);`.

Question 13.1 : Programmer ce jeu en PHP. Pour cela, créer un formulaire permettant à l'utilisateur de choisir un nombre. Le nombre de tentatives restantes sera donné à l'aide d'une balise input de type hidden. Le nombre aléatoire sera stocké à l'aide d'une balise input de type hidden.

## Exercices SIO

Question 13.2 : Le PHP est-il adapté pour la programmation de ce jeu ?