TP PHP

Utiliser l'instruction d'affichage **echo** pour afficher :

1. une chaîne de caractères,
2. le contenu d'une variable,
3. une chaîne de caractères associée au contenu d'une variable .

TP N°2

1) Utiliser la fonction date pour afficher la date du jour en français (Lundi 23 Septembre par exemple).

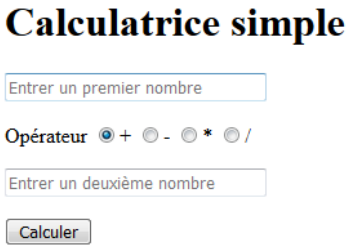
2) Écrire les fonctions PHP qui :
   a) affiche tous les éléments d'un tableau, (voir signification de sizeof)
   b) calcule la moyenne des nombres contenus dans un tableau donné,
   c) indique combien d'éléments sont supérieurs ou égaux à 10,
   d) teste si la note 20 est présente ou non,
   e) détermine la meilleure note obtenue.
   f) Ecrire une fonction qui fournit le jour en français à partir du numéro de jour de la semaine.
      i) Utiliser cette fonction pour afficher le jour d'aujourd'hui
      ii) Utiliser cette fonction Pour afficher tous les jours de la semaine (Utilisez une boucle et la fonction date)

3) Choisir un nombre de trois chiffres. Effectuer ensuite des tirages aléatoires et compter le nombre de tirages nécessaire pour obtenir le nombre initial. Arrêter les tirages et afficher le nombre de coups réalisés. Réaliser ce script d'abord avec l'instruction while puis avec l'instruction for. (on utilisera la fonction rand())

4) Déterminez quel jour de la semaine seront tous les premier Mai des années comprises entre 2017 et 2022. Si le jour est un samedi ou un dimanche, affichez le message « Désolé !». Si le jour est un vendredi ou un lundi affichez « Week end prolongé !».

TP N°3

1) 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.



2) Écrire un formulaire (formulaire.php) qui demande le nom, le prénom, la situation (célibataire, marié, divorcé), la fonction (technicien, développeur, chef de projet) de l'utilisateur et la photo (qui sera uploadée dans le dossier /uploads). Le bouton *ENVOYER* de ce formulaire lancera la page (traitement.php) qui provoquera l'affichage d'une du contenu suivant (avec les bonnes valeurs bien évidemment) :

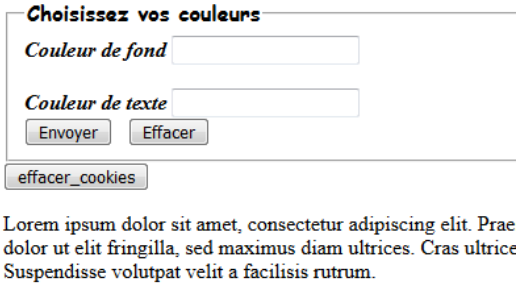
Nom : *le nom*
Prénom : *le prénom*
Fonction : *la fonction*
Situation familiale : *la situation*
Fichier : *lien vers la photo*

3) Page protégé par mot de passe
La page doit contenir à la fois le formulaire et le message secret.
Il faut construire le code de votre page en deux grandes parties :

- si aucun mot de passe n'a été envoyé (ou s'il est faux) : afficher le formulaire ;
- si le mot de passe a été envoyé et qu'il est bon : afficher les codes secrets.

TP N°4

1) Créer un formulaire de saisie des deux codes couleur préférés du visiteur du site pour la couleur de fond et le texte de la page. Les enregistrer dans deux cookies valables deux mois. À l'ouverture de la page d'accueil, récupérer ces valeurs et créer un style utilisant ces données. Offrez la possibilité d'effacer les cookies.



2) Même question mais en stockant les deux informations dans un même cookie. (voir fonction serialize et unserialize)

1) Comment faire de l'insertion de texte ?
Créez manuellement un fichier.txt avec le contenu suivant:
On effectue diverses opérations sur les fichiers en php.

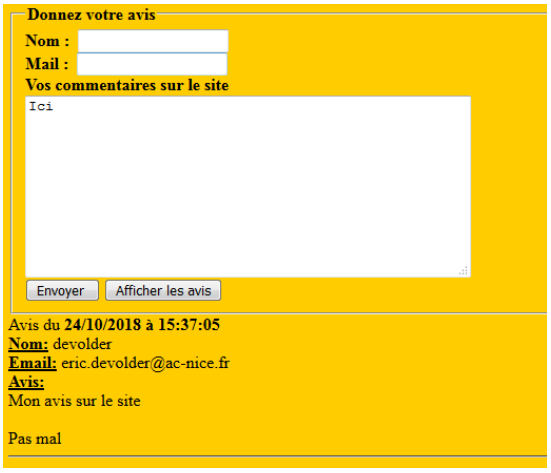
Pour rappel, le PHP est un langage qui s'execute côté serveur,
à la différence du HTML.

Exécuter le code ci-dessous:

```
<?php
$fichier = fopen('fichier.txt' , 'r+');
$texte = "Insertion de texte ! \n";
fseek($fichier,10);
fwrite($fichier,$texte);
fclose($fichier);
?>
```

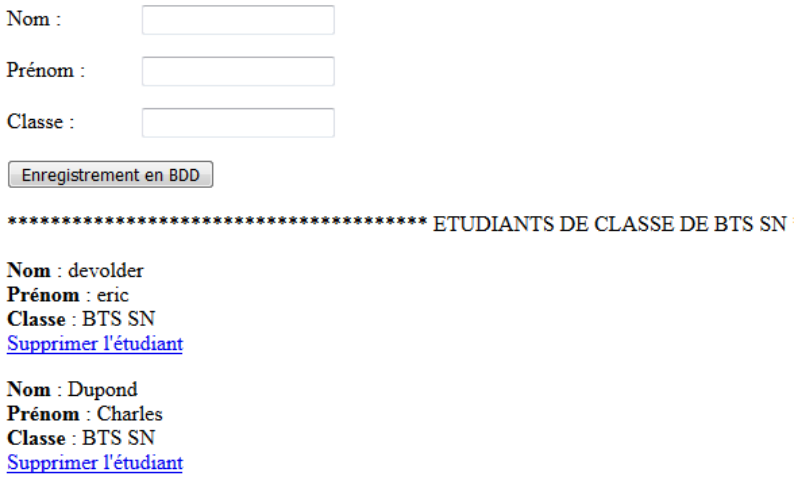
Que constatez-vous ? Comment modifier le code pour insérer du texte ?

2) Créer un livre d'or avec les champs : Nom , Email, Avis . Les données du formulaire s'enregistreront dans le fichier **livre.txt** avec en plus la date de création**.** Créé un bouton pour afficher les avis présents dans le fichier **livre.txt**
On pourra s'aider de la fonction: **fgetcsv .**
Attention également aux saut de lignes dans le champ Avis. (voir éventuellement str_replace).



TP N°6

1) Le but est de créer un formulaire (nom, prénom, classe) avec enregistrement en base de données MySQL avec possibilité de lecture du contenu et de suppression de certains éléments.



2) Un blog avec des commentaires:
Chaque billet du blog possèdera ses propres commentaires. Dans ce TP, nous nous concentrerons uniquement sur l'affichage des billets et des commentaires.
Il y aura deux pages à réaliser :

- index.php : liste des cinq derniers billets ;
- commentaires.php : affichage d'un billet et de ses commentaires.

Le visiteur arrive d'abord sur l'index où sont affichés les derniers billets. S'il choisit d'afficher les commentaires de l'un d'eux, il charge la page commentaires.php qui affichera le billet sélectionné ainsi que tous ses commentaires. Bien entendu, il faudra envoyer un paramètre à la page commentaires.php pour qu'elle sache quoi afficher…
Il sera possible de revenir à la liste des billets depuis les commentaires à l'aide d'un lien de retour.
Vous importerez le fichier **blog.sql** et **style.css .**

nl2br() permet de convertir les retours à la ligne en balises HTML
. C'est une fonction dont vous aurez sûrement besoin pour conserver facilement les retours à la ligne saisis dans les formulaires.
N'oubliez pas les éléments essentiels de sécurité, notamment la protection de tous les textes par **htmlspecialchars** Et ne faites jamais confiance à l'utilisateur !

3) Création d'une zone membre:

Le but est de créer une zone publique et une zone privée qui ne sera accessible que si on est connecté.

On va créer 7 fichiers:

- index.php : zone public

Accueil Zone prive Inscription Connexion

ZONE PUBLIC

Ceci est la zone public

- entete.php :

Accueil Zone prive Inscription Connexion

- connexionbdd.php : connexion à la base MySQL (site-web)

  Créez une table **membres** avec les champs:
  - id: type int en clé PRIMAIRE et AUTO-INCREMENT
  - pseudo: type varchar(255)
  - password: type varchar(255)
  - email: type varchar(255)
  - date_inscription: type datetime
- inscription.php : page d'inscription qui testera le bon format des différents champs avec les bons patterns et vérifiera les différentes validités (pseudo et email déjà présents en base de données ? les deux mots de passe saisis sont-ils identiques ? L'adresse e-mail a-t-elle une forme valide ? …) et sécurisera les données envoyées.

  Il ne faut pas stocker le mot de passe en dur dans la base de données . Il faut le hacher grâce à une fonction qui transforme n'importe quel texte en un nombre hexadécimal qui représente le mot de passe mais qui est illisible.

On utilise la fonction sha1

```
$password = sha1($_POST['password']); // Hachage du mot de passe
```

Accueil Zone prive Inscription Connexion

Inscription à l'espace membre

Votre pseudo

Votre adresse e-mail

Votre mot de passe — Au moins un chiffre, une lettre majuscule et minuscule et au moins 8 caractères.

Confirmez votre mot de passe — Au moins un chiffre, une lettre majuscule et minuscule et au moins 8 caractères.

ENVOYER

- connexion.php : page de connexion

  Nous utiliserons le système de sessions ainsi que les cookies pour une connexion automatique.

Accueil Zone prive Inscription Connexion

Connexion à l'espace membre

Votre pseudo

Votre mot de passe — Au moins un chiffre, une lettre majuscule et minuscule et au moins 8 caractères.

Connexion automatique

ENVOYER

- page-utilisateur.php (page privé qui sera accessible une fois connecté)

Accueil Zone prive Inscription Connexion

Bonjour eric

Vous etes dans votre espace prive

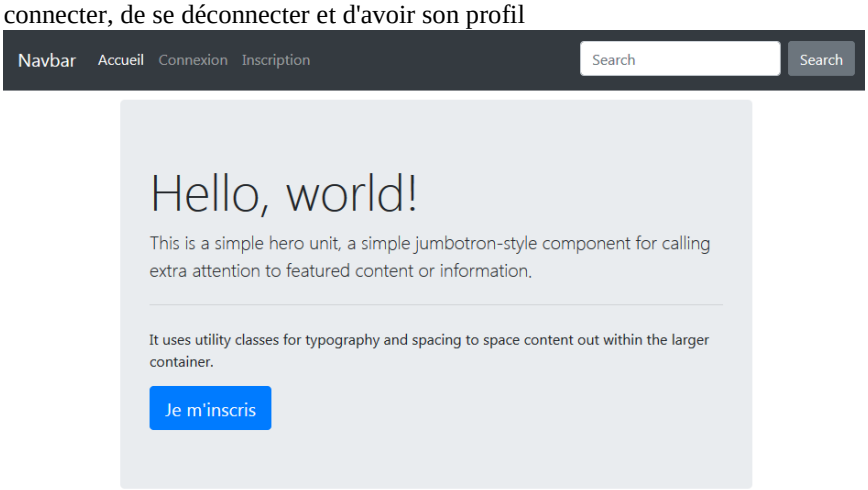
Deconnexion

- deconnexion.php (page qui détruit les sessions et cookies)

Création d'un espace membre en utilisant le framework Bootstrap:

(voir document bootstrap.pdf)

On aura la possibilité de s'inscrire, d'activer le compte par email, de réinitialiser le mot de passe en cas d'oublie, de se connecter, de se déconnecter et d'avoir son profil



Création d'une version MVC également

TP N°7

1) Écrivez une classe représentant une ville. Elle doit avoir les propriétés nom et département et une méthode affichant « la ville X est dans le département Y ». Créez des objets ville, affectez leurs propriétés, et utilisez la méthode d'affichage.

2) Modifiez la classe précédente en la dotant d'un constructeur.
Réalisez les mêmes opérations de création d'objets et d'affichage.

3) Créez une classe représentant une personne. Elle doit avoir les propriétés nom, prénom et adresse, ainsi qu'un constructeur et un destructeur. Une méthode getpersonne() doit retourner les coordonnées complètes de la personne. Une méthode setadresse() doit permettre de modifier l'adresse de la personne. Créez des objets personne, et utilisez l'ensemble des méthodes.

4) Créez une classe nommée form représentant un formulaire HTML. Le constructeur doit créer le code d'en-tête du formulaire en utilisant les éléments <form> et <fieldset>. Une méthode settext() doit permettre d'ajouter une zone de texte. Une méthode setsubmit() doit permettre d'ajouter un bouton d'envoi. Les paramètres de ces méthodes doivent correspondre aux attributs des éléments HTML correspondants. La méthode getform() doit retourner tout le code HTML de création du formulaire. Créez des objets form, et ajoutez-y deux zones de texte et un bouton d'envoi. Testez l'affichage obtenu.
Le fichier contenant la définition de la classe form (TP7-4-form.php) est indépendant ce qui permet son inclusion dans d'autres scripts en vue de l'utilisation de la classe ou de son extension.

TP N°8

Créez une sous-classe nommée form2 en dérivant la classe form de **TP7-4-form.php** Cette nouvelle classe doit permettre de créer des formulaires ayant en plus des boutons radio et des cases à cocher. Elle doit donc avoir les méthodes supplémentaires qui correspondent à ces créations. Créez des objets, et testez le résultat.

TP N°9

1) Comptes bancaires:

On veut créer une classe CompteBancaire avec les attributs suivants:

private $numero_compte; // numéro du compte
private $devise; // la devise utilisé (EUR, USD ….)
private $solde=0; // le solde du compte
private $coeff=1; // coeff euros (Pour la conversion des euros dans une autre monnaie)
private $titulaire; // titulaire du compte
private static $nbr_comptebancaire=0; // nombre de compte bancaire créé

Cette classe doit permettre de créer un compte bancaire avec son numéro, son titulaire, son solde et sa devise et d'avoir le nombre de comptes créés.
On doit pouvoir pour le compte créé:

- obtenir le nom du titulaire
- obtenir sa devise
- obtenir son solde
- créditer le compte (en euros)
- débiter le compte (en euros) si le solde est suffisant
- effectuer un virement (en euros) sur un autre compte si le solde est suffisant.
- changer la devise et de faire la conversion dans une autre devise (voir complément ci-dessous)

Créer deux comptes bancaires et tester les différentes méthodes

Créer maintenant une classe CompteEpargne hérité de CompteBancaire et avec l'attribut **private** $tauxInteret
Cette classe doit permettre de créer un compte épargne avec son numéro, son titulaire, son solde, sa devise et son taux d'intêret.
On doit pouvoir pour le compte créé:

- obtenir le taux d'intérêt
- calculer les intérêts
- ajouter les intérêts au solde

Compléments sur le parsage de fichier xml:

Parser un fichier xml (langage de balisage avec des attributs)
Soit un fichier test.xml avec le contenu suivant:

```
<A attr="val_a">
   <B attr="val_b1">
   </B>
   <B attr="val_b2">
      <C>
         <D>texte</D>
      </C>
   </B>
</A>
```

On veut accéder aux différentes valeurs (balises et attributs)
On utilise la fonction **simplexml_load_file** http://php.net/manual/fr/function.simplexml-load-file.php

```php
<?php

$xml = simplexml_load_file('test.xml');

echo '<pre>';print_r($xml);echo'</pre>';

echo $xml['attr'].'<br>'; // Résultat : "val_a"

echo $xml->B[1]['attr'].'<br>'; // Résultat : "val_b2"

foreach ($xml->B as $b) {
   echo $b['attr'].'<br>'; // Résultat : "val_b1" puis "val_b2"
```

```
}

echo $xml->B[1]->C->D.'<br>'; // Résultat : "texte"
```

Pour la conversion de devise , on va utiliser ce fichier xml.
http://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml

Si on reprend le même principe, on accédera à la devise RUB grâce au code suivant:

```
<?php

$tauxChange = simplexml_load_file('http://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml');

echo 'Pays: '.$tauxChange->Cube->Cube->Cube[14]['currency'].' - taux: '.$tauxChange->Cube->Cube->Cube[14]['rate'];
```

Modifier maintenant le code pour afficher tous les pays et taux de change sous la forme:

USD : 1.1346
JPY : 124.72
BGN : 1.9558
CZK : 25.697
……

2) Créer deux classes:

1) Une class qui permet de créer des personnages

Avec les **attributs privés** suivant:

- $degat (dégât d'un personnage)
- $nom (nom d'un personnage)
- $experience (expérience d'un personnage)
- $force (force d'un personnage)

Avec les méthodes suivantes:

- frapper (frapper un personnage)
- getDegat (obtenir les dégâts d'un personnage)
- getExperience (obtenir l'expérience d'un personnage)
- getNom (obtenir le nom d'un personnage)

2) Une class qui permet de créer des groupes de personnages (experts ou débutants)

Avec les **attributs privés** suivant:

- $nom (nom du groupe)
- $personnages (tableau de personnages: ensemble des personnages d'un groupe)

Avec les méthodes suivantes:

- addPersonnage (ajouter un personnage dans un groupe donné)
- getCountGroup (compter le nombre des membres d'un groupe donné)
- getPersonnageGroup (obtenir le nom des personnages d'un groupe donné)

Créez un groupe **expert** et **débutant**
Créez deux personnages.

- Si un des personnages en frappe un autre avec une certaine force alors son expérience augmente de 1.
- Le personnage frappé a ses dégâts qui augmente de la valeur de la force: $degat =$degat + $force
- Si l'expérience d'un personnage **est inférieure à 10** alors ce personnage fait parti des **débutants** sinon c'est un **expert.**

Testez les différentes méthodes.
Voilà ce que l'on pourrait obtenir.

```
Nbre personnages débutants 1
Il y a Fabien dans le groupe débutant

Nbre personnages experts 1
Il y a Eric dans le groupe expert

Eric a 11 d'expérience
Fabien a 10 d'expérience

Eric a 0 de dégât
Fabien a 20 de dégât
```

TP N°10

Créez deux namespaces nommés Firme\Client et Firme\Commercial possédant chacun des classes Personne. Chacune d'elles doit avoir des propriétés pour enregistrer les coordonnées de la personne et son code, un constructeur, des méthodes set()et get()pour pouvoir modifier et afficher les propriétés.
Créez ensuite des objets représentant deux clients et un commercial.

1) Création du namespace Firme\Client fichier TP10-1.php
2) Création du namespace Firme\Commercial TP10-2.php
3) Utilisation des namespaces. fichier TP10-3.php