

TP2 : PHP (Corrigé)

Exercice 1 : Premiers pas en PHP

Voici un exemple de code PHP.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
4 <head>
5     <title > <?php echo 'Premiers pas en PHP'; ?> </title>
6     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 </head>
8 <body>
9 <h1> Mes premiers pas en PHP </h1>
10 <?php $temps=2; echo '<p> Je débute depuis ' ; echo $temps; echo ' heures... </p>'; ?>
11 <p> Mais cela a l'air intéressant <?php echo '!' ?> </p>
12 <?php echo '
13 <h2> Vive le PHP </h2>
14 <p> Les pages vont pouvoir être dynamiques! </p>
15 ' ; ?>
16 <p> Encore quelques paragraphes </p>
17 echo '<p> Avant dernier paragraphe </p>';
18 <p> Voilà, c'est terminé! </p>
19 </body>
20 </html>

```

Répondre aux questions suivantes :

1. Donner dans ce fichier les parties correspondant à du code PHP et à du code XHTML.
2. Si ce fichier s'appelle `exo1.php`, et s'il est stocké sur le site Web `www.exemple.org` dans le répertoire `PremierTP`, comment "exécuter" ce fichier ? Que donne son exécution ? Ceci est-il valide ? Pourquoi ? Corriger alors le problème.
3. Remplacer les trois instructions `echo` de la ligne 10 par une seule.

Correction :

1. XHTML : lignes 1-4, balises ouvrantes et fermantes `title` de la ligne 5, lignes 6-9, début et fin de la ligne 11, lignes 16-20. Le reste est du PHP.
2. Avec un navigateur, taper l'url `http://www.exemple.org/PremierTP/exo1.php`. L'interprétation du code PHP produit du code XHTML. Le résultat de l'interprétation de ce code donne

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
4 <head>
5     <title > Premiers pas en PHP </title>
6     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 </head>
8 <body>
9 <h1> Mes premiers pas en PHP </h1>
10 <p> Je débute depuis 2 heures... </p><p> Mais cela a l'air intéressant ! </p>
11 <h2> Vive le PHP </h2>
12 <p> Les pages vont pouvoir être dynamiques! </p>
13 <p> Encore quelques paragraphes </p>
14 echo '<p> Avant dernier paragraphe </p>';
15 <p> Voilà, c'est terminé! </p>

```

```
16 </body>
17 </html>
```

Le code XHTML généré n'est pas valide car il contient du texte (`echo ' et ';`) qui n'est contenue dans aucune balise de type `block`. Pour corriger cela, il faut ajouter `<?php` au début de la ligne 17 et `?>` à la fin de cette même ligne.

3. `echo '<p> Je débute depuis '. $temps . ' heures... </p>';` ◇

Exercice 2 : Inclusion d'en-tête et pied de page

Le langage PHP permet d'inclure des fichiers dans d'autres. Ceci permet alors de décomposer un code XHTML ou PHP en plusieurs parties logiques et d'insérer ensuite ces différentes parties dans un fichier PHP. Utiliser l'inclusion de fichiers pour séparer, dans le fichier précédent, le corps du document du reste. Quel est l'intérêt d'une telle décomposition ?

Correction :

Fichier `debut.php` :

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
4 <head>
5     <title > Premiers pas en PHP </title>
6     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 </head>
8 <body>
```

Fichier `fin.php` :

```
1 </body>
2 </html>
```

Fichier `exo2.php` :

```
1 <?php require('debut.php'); ?>
2 <h1> Mes premiers pas en PHP </h1>
3 <p> Je débute depuis 2 heures... </p><p> Mais cela a l'air intéressant ! </p>
4 <h2> Vive le PHP </h2>
5 <p> Les pages vont pouvoir être dynamiques! </p>
6 <p> Encore quelques paragraphes </p>
7 echo '<p> Avant dernier paragraphe </p>';
8 <p> Voilà, c'est terminé! </p>
9 <?php require('fin.php'); ?>
```

Intérêts : clarification du code, décomposition en parties logiques, les parties d'un site Web qui ne bougent pas et sont présentes dans toutes les pages Web du site peuvent être écrites une seule fois. ◇

Exercice 3 : Affichage des carrés de tous les nombres entre 1 et 30*

Créer une page PHP permettant d'afficher, sous forme de liste non ordonnée, les carrés des nombres de 1 à 30 selon le format :

- $1^2 = 1$
- $2^2 = 4$

...

- $30^2 = 900$

Utiliser d'abord l'instruction `echo` avec des apostrophes puis avec des guillemets.

Remarque : Pour mettre du texte en exposant en XHTML, il est possible d'utiliser la balise `sup`.

Créer ensuite une fonction prenant en paramètre un nombre n et retournant la chaîne de caractères $n^2 = m$. Utiliser cette fonction pour produire la même page Web.

Correction :

```

1 <?php require('../Exo2/debut.php'); ?>
2 <h1> Suite de carrés </h1>
3 <ul>
4 <?php
5 for ($i = 1; $i < 31 ; $i++)
6     echo '<li>' . $i . '<sup> 2 </sup> = ' . $i * $i . '</li>'. "\n";
7 ?>
8 </ul>
9 <?php require('../Exo2/fin.php'); ?>

```

Instruction echo avec des guillemets : echo " \$i ² =" . \$i*\$i . "\n";.

```

1 <?php
2     function chCarre($nb)
3     {
4         return $nb . '<sup>2 </sup> = ' . $nb * $nb ;
5     }
6
7 for ($i = 1; $i < 31 ; $i++)
8     echo '<li>' . chCarre($i) . "</li>\n";
9 ?>

```

Exercice 4 : Table de multiplication*

Créer un script PHP permettant d'afficher la table de multiplication donnée par la figure 3.3.

	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

FIGURE 3.3 – Table de multiplication

L'opérateur modulo (symbole %) permet de donner le reste de la division entière. Pour déterminer si une ligne est paire ou impaire, il suffit alors de regarder la valeur du numéro de ligne modulo 2.

Correction :

```
1 <table>
2 <?php
3     $n=10;
4     for($ligne=1;$ligne<$n;$ligne++)
5     {
6         if($ligne%2)
7             $class='impair';
8         else
9             $class='pair';
10        echo '<tr class="'. $class. '>';
11        for($col=1;$col<$n;$col++)
12        {
13            if($ligne==1 || $col==1)
14                $cellule ='th';
15            else
16                $cellule ='td';
17            echo "<$cellule>";
18            if(($ligne!=1) and ($ligne==$col))
19                echo '<strong>';
20            if($ligne!=1 || $col!=1)
21                echo $ligne*$col;
22            if(($ligne!=1) and ($ligne==$col))
23                echo '</strong>';
24            echo "</$cellule>";
25        }
26        echo "</tr>\n";
27    }
28    ?>
29 </table>
```

CSS :

```
1 table
2 {
3     border-collapse : collapse;
4 }
5
6 th,td
7 {
8     border : 1px solid black;
9     width : 40px;
10    height : 40px;
11    text-align : center;
12 }
13
14 th
15 {
16     background-color : #D48030;
17 }
18
19 .impair
20 {
21     background-color : #BEA58E;
22 }
23
24 .pair
25 {
```

```

26     background-color : #EBD9C8;
27 }

```

Exercice 5 : Initiation aux tableaux

Créer un tableau ayant pour valeur les noms des douze mois de l'année. Quelles sont alors les clés? Parcourir ensuite le tableau pour afficher les mois de l'année sous forme de liste ordonnée.

Correction :

```

1 <?php
2 $mois = array('janvier','février','mars','avril','mai','juin',
3 'juillet','août','septembre','octobre','novembre','décembre');
4
5 echo '<ol>';
6 foreach($mois as $m)
7     echo '<li>' . $m . "</li>\n";
8 echo '</ol>';
9 ?>

```

Les clés sont les entiers 0,1,...,11. (Attention, les indices attribués automatiquement commencent à 0!) ◇

Créer un tableau associant à chaque mois de l'année le nombre de jours du mois. (On supposera que l'année n'est pas bissextile.) Quelles sont alors les clés? Les valeurs? Afficher, sous forme de tableau, le nombre de jours de chaque mois.

Correction :

```

1 $jourMois = array('janvier'=>31, 'février' => 28, 'mars' =>31, 'avril' => 30, 'mai' => 31,
2 'juin' => 30, 'juillet' => 31, 'août' => 31, 'septembre' => 30, 'octobre' => 31,
3 'novembre' => 30, 'décembre' => 31);
4
5 echo '<table> <tr> <th> Mois </th> <th> Nombre de jours </th> </tr>'. "\n";
6 foreach($jourMois as $m => $nbJ)
7     echo '<tr> <td>' . $m . ' </td><td> ' . $nbJ . " </td> </tr>\n";
8 echo '</table>';

```

Les clés sont les noms des mois de l'année et les valeurs les nombres de jours par mois. ◇

Exercice 6 : Tableau à deux dimensions

On définit le tableau suivant :

```

1 $personnes = array(
2 'mdupond'=> array('prenom' => 'Martin', 'nom' => 'Dupond', 'age' => 25, 'ville' => 'Paris'),
3 'jm'=> array('prenom' => 'Jean', 'nom' => 'Martin', 'age' => 20, 'ville' => 'Villetaneuse'),
4 'toto'=> array('prenom' => 'Tom', 'nom' => 'Tonge', 'age' => 18, 'ville' => 'Epinay'),
5 'arn'=> array('prenom' => 'Arnaud', 'nom' => 'Dupond', 'age' => 33, 'ville' => 'Paris'),
6 'email'=> array('prenom'=>'Emilie', 'nom'=>'Ailta', 'age'=>46, 'ville'=>'Villetaneuse'),
7 'dask' => array('prenom'=>'Damien', 'nom'=>'Askier', 'age'=>7, 'ville'=>'Villetaneuse')
8 );

```

Question 6.1 : Quelles sont les clés du tableau `$personnes` et leur type? De quel type sont les valeurs de ce tableau? Quelle est la valeur associée à 'toto'?

Question 6.2 : Comment accéder à la valeur 33 dans le tableau? À la valeur 'Epinay'? Au tableau contenant les valeurs 'Damien', 'Askier', 7, 'Villetaneuse'?

Question 6.3 : Écrire une fonction permettant d'afficher le tableau dans son ensemble. Ajouter une première ligne contenant les clés 'prenom', 'nom', 'age' et 'ville'. Ajouter ensuite un fichier CSS afin d'obtenir le tableau donné par la figure 3.4.

prenom	nom	age	ville
Martin	Dupond	25	Paris
Jean	Martin	20	Villetaneuse
Tom	Tonge	18	Epinay
Arnaud	Dupond	33	Paris
Emilie	Alta	46	Villetaneuse
Damien	Askier	7	Villetaneuse

FIGURE 3.4 – Tableau d'informations de personnes

Question 6.4 : Écrire une fonction permettant d'afficher sous forme de tableau (en utilisant toujours le CSS), les informations des personnes habitant dans une ville donnée en paramètre. Par exemple, si la fonction est appelée avec le tableau `$personnes` défini précédemment et la ville 'Epinay', le tableau affiché doit alors contenir uniquement la ligne relative à `toto`.

Correction :

Les clés sont de type chaînes de caractères et les différentes valeurs de clés sont : 'mdupond', 'jm', 'toto', 'arn', 'email' et 'dask'. Les valeurs du tableau sont de type tableau. La valeur associée à 'toto' est le tableau : `array('prenom' => 'Tom', 'nom' => 'Tonge', 'age' => 18, 'ville' => 'Epinay')`.

Pour accéder à la valeur 33, écrire : `$personnes['arn']['age']`.

Pour accéder à Epinay, écrire : `$personnes['toto']['ville']`.

Pour accéder au tableau `$personnes['dask']`.

```

1  fonction  affichageTableau ($personnes)
2  {
3      echo '<table>'. "\n";
4      $ligne = 0;
5      foreach ($personnes as $pseudo => $info)
6      {
7          if($ligne==0)
8          {
9              echo '<tr>'. "\n";
10             foreach ($info as $c => $v)
11                 echo '<th>' . $c . '</th>' ;
12             echo '</tr>'. "\n";
13             $ligne++;
14         }
15         if($ligne%2==1)
16             $class = 'impair';
17         else
18             $class = 'pair';
19         echo '<tr class="' . $class . '>'. "\n";
20         foreach ($info as $v)
21             echo '<td>' . $v . '</td>' ;
22         echo '</tr>'. "\n";
23         $ligne++;
24     }
25     echo '</table>'. "\n";
26 }

```

```

1  fonction  affichageHabitantsVille ($personnes, $ville )

```

```

2 {
3     echo '<table>'. "\n";
4     $entete = true;
5     foreach ($personnes as $pseudo => $info)
6     {
7         if($entete)
8         {
9             echo '<tr>'. "\n";
10            foreach ($info as $c => $v)
11                echo '<th>' . $c . '</th>' ;
12            echo '</tr>'. "\n";
13            $entete = false;
14        }
15        if($info['ville']==$ville)
16        {
17            echo '<tr>'. "\n";
18            foreach ($info as $v)
19                echo '<td>' . $v . '</td>' ;
20            echo '</tr>'. "\n";
21        }
22    }
23    echo '</table>'. "\n";
24 }

```

La deuxième fonction peut être modifiée pour ajouter les class `pair` et `impair`. Le code CSS associé au tableau est équivalent à celui donné pour la table de multiplication.

Pour les étudiants en avance, on peut également leur demander de faire une fonction affichant les informations de la personne la plus âgée (ou la plus jeune). ◇

Exercice 7 : Paramètres dans l'url

Question 7.1 : Appeler une page PHP en passant dans l'url un paramètre de nom `pseudo` et ayant pour valeur un des pseudonymes du tableau `$personnes`. Faire en sorte que la page affiche le pseudo et les informations associées contenues dans le tableau `$personnes`. Améliorer le script pour que ce dernier affiche `Désolé, votre pseudonyme n'apparaît pas dans la liste` si le pseudonyme n'est pas une clé du tableau `$personnes`.

Correction :

Exemple d'appel : <http://aquanux/~001/TP2/exo7.php?pseudo=toto>

```

1 <?php
2     $personnes = array(...);
3
4     $trouve= false ;
5     if( ( isset($_GET['pseudo']) ) and ( trim($_GET['pseudo']) !='' ) )
6     {
7         $pseudo = $_GET['pseudo'];
8         foreach ($personnes as $p => $info)
9             if ($p==$pseudo)
10            {
11                echo '<p> Bonjour ' . $info['prenom'] . ' ' . $info['nom'] .
12                ' . Vous avez ' . $info['age'] . ' et vous habitez ' .
13                $info['ville'] . ' . </p>'. "\n";
14                $trouve=true;
15            }
16    }
17    if($trouve==false)
18    {
19        echo '<p> Désolé, votre pseudo n'apparaît pas dans la liste! </p>'. "\n";

```

```
20     }
21  ?>
```

Question 7.2 : Créer un formulaire permettant à l'utilisateur de saisir le pseudonyme à rechercher afin de faciliter la saisie pour l'utilisateur. Mettre ensuite directement le formulaire dans le script PHP créé précédemment afin de pouvoir effectuer facilement plusieurs recherches. Faire en sorte que le champ de saisie du pseudonyme contienne la dernière valeur saisie.

Correction :

```
1 <h1> Veuillez saisir un pseudonyme : </h1>
2 <form method="get" action="exo7.php">
3 <p> Pseudonyme : <input type="text" name="pseudo" value="<?php if(isset($_GET['pseudo']))
4                                     echo $_GET['pseudo'];?>"/>
5 <input type="submit" value="Recherche" /> </p>
6 </form>
```

Question 7.3 : Créer un deuxième formulaire demandant un pseudonyme, un prénom, un nom, un âge et une ville, et ajoutant dans le tableau `$personnes` une nouvelle personne dont les valeurs sont celles données par le formulaire. Ajouter plusieurs personnes et expliquer alors le problème. (Afficher le tableau en entier pour mieux voir le problème.) Comment remédier à ceci (réponse sans code car les connaissances nécessaires à la réponse dépassent le cadre de ce cours) ?

Correction :

Comme on crée deux formulaires sur une même page, il vaut mieux ajouter à chaque formulaire un champ caché permettant de savoir de quel formulaire viennent les paramètres. La correction vérifie, à l'aide d'une fonction, si les différents paramètres sont corrects pour l'ajout (s'ils existent et s'ils ne sont pas vides).

```
1 <?php
2 function testParametreFormulaireAjout ()
3 {
4     $ok = true;
5     $param = array('pseudo', 'nom', 'prenom', 'age', 'ville');
6     foreach ($param as $v)
7         if( !(isset($_GET[$v])) or (trim($_GET[$v])=='') )
8             $ok=false;
9     return $ok;
10 }
11
12 if( ( isset($_GET['formulaire']) ) and ( $_GET['formulaire']=='ajouter' ) )
13 {
14     if(testParametreFormulaireAjout ())
15     {
16         $personnes[$_GET['pseudo']] = array('prenom' => $_GET['prenom'],
17         'nom' => $_GET['nom'], 'age' => $_GET['age'], 'ville' => $_GET['ville']);
18         echo '<p> La personne a bien été ajoutée </p>';
19     }
20 }
21
22 ?>
23 <h1> Ajout d'une personne : </h1>
24 <form method="get" action="exo7.php">
25 <p> Pseudo : <input type="text" name="pseudo"/>
26 <input type="hidden" name="formulaire" value="ajouter"/> </p>
27 <p> Prénom : <input type="text" name="prenom"/> </p>
28 <p> Nom : <input type="text" name="nom"/> </p>
29 <p> Age : <input type="text" name="age"/> </p>
30 <p> Ville : <input type="text" name="ville"/> </p>
```



```

31 <p> <input type="submit" value="Ajouter une personne"/> </p>
32 </form>

```

Remarque : Pour les exercices suivants, des exemples sont fournis à l'adresse <http://www-lipn.univ-paris13.fr/~lacroix/Documents/TP/IntroWeb/> pour montrer précisément à quoi doit ressembler chaque script.

Exercice 8 : Contenu XHTML protégé par mot de passe²

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.

Correction :

```

1 ... <body>
2 <?php
3     //On teste si l'utilisateur a saisi un mot de passe
4     if( isset($_POST['mdp']) and $_POST['mdp']=='kangourou')
5     {
6     ?>
7     <!-- Si l'utilisateur a saisi le bon mot de passe -->
8     <p> Voici le contenu secret obtenu uniquement si vous avez le mot de passe...!!! </p>
9     <p> Code : VIVE LE PHP!!! </p>
10
11 <?php
12     } //Accolade fermante du if
13     else
14     {
15     ?>
16     <!-- Si l'utilisateur n'a pas saisi le bon mot de passe -->
17     <h1> Veuillez saisir le mot de passe pour accéder au contenu privé </h1>
18     <form action="secret.php" method="post" >
19     <p> <input type="password" name="mdp" /> <input type="submit" value="Connexion"/> </p>
20     </form>
21 <?php }// Accolade fermante du else ?>
22 </body> ...

```

Exercice 9 : Calculatrice²

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.

Correction :

```

1 ...<body>
2 <?php
3 function calcul ($nb1, $op, $nb2)
4 {
5     if ($op=="ad")
6         return $nb1 . ' + ' . $nb2 . ' = ' . ($nb1 + $nb2);
7     elseif ($op=="so")
8         return $nb1 . ' - ' . $nb2 . ' = ' . ($nb1 - $nb2);

```

2. Exercice tiré du site du zéro

```

9     elseif ($op=="mu")
10         return $nb1 . ' * ' . $nb2 . ' = ' . ($nb1 * $nb2);
11     elseif ($op=="di")
12         if($nb2!=0)
13             return $nb1 . ' / ' . $nb2 . ' = ' . ($nb1 / $nb2);
14         else
15             return NULL;
16     else
17         return NULL;
18 }
19
20 if (isset($_GET['nb1']) and isset($_GET['nb2']) and isset($_GET['op']))
21 {
22     if (is_numeric($_GET['nb1']) and is_numeric($_GET['nb2']))
23     {
24         $a = (int)$_GET['nb1'];
25         $b = (int)$_GET['nb2'];
26         $v = calcul($a,$_GET['op'],$b);
27         if($v!=NULL)
28             echo '<p>' . $v . '</p>';
29     }
30 }
31 ?>
32 <h1> Calculatrice simple </h1>
33 <form action="calculatrice.php" method="get">
34
35 <p> Premier nombre : <input type="text" name="nb1" /> </p>
36 <p> Opérateur <select name="op">
37     <option value="ad"> addition </option>
38     <option value="so"> soustraction </option>
39     <option value="mu"> multiplication </option>
40     <option value="di"> division </option>
41 </select> </p>
42 <p> Deuxième nombre : <input type="text" name="nb2" /> </p>
43 <p> <input type="submit" value="Calculer" /> </p>
44 </form>
45
46 </body> </html>

```

◇

Afin de faciliter la saisie par l'utilisateur, créer un formulaire contenant une seule balise `input` de type `text` où l'utilisateur saisit directement l'opération (par exemple : 17.5 * 13.46). Vérifier, à l'aide des expressions régulières, que le texte saisi par l'utilisateur correspond à une opération. Dans ce cas, afficher le résultat.

Correction :

Dans le code précédent, mettre une seule balise `input` de type `text` dans le formulaire dont le `name` est `'operation'`. Remplacer ensuite le test `if (is_numeric...)` par le code suivant :

```

1     if(preg_match('#^ *([0-9]+(?:\.[0-9]+)?) *([*+/-]) *([0-9]+(?:\.[0-9]+)?) *$#',
2         $_GET['operation'],$res) )
3     {
4         $a = (double)$res[1];
5         $b = (double) $res[3];
6         $v = calcul($a,$res[2],$b);
7         if($v!=NULL)
8             echo '<p>' . $v . '</p>';
9     }

```

◇

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.

Le script PHP doit alors indiquer si la saisie est correcte et dans ce cas, afficher 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, envoyer alors un mail sur son adresse de l'iut contenant les informations relatives à l'utilisateur et le message saisi, grâce à la fonction PHP : `mail('email_destinataire', 'sujet', 'message')`; 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.

Correction :

Je ne donne pas la réponse aux deux dernières questions. Pour la troisième question, il suffit de rajouter que l'on peut avoir, tous les deux chiffres, la partie [-]?. La dernière question est du code CSS pour améliorer la présentation. Les étudiants peuvent reprendre le CSS du site d'énigmes.

```

1  ...
2  <body>
3  <h1> Contact </h1>
4  <p> Veuillez saisir vos infos... </p>
5
6  <?php
7  function testParametres($tab)
8  {
9      foreach($tab as $v)
10         if( !isset($_POST[$v]) or trim($_POST[$v])=='')
11             return false;
12     return true;
13 }
14
15 function testP_er()
16 {

```

```

17 $ok=true;
18 if (!preg_match('#^[a-zA-Z]{4,8}$#',$_POST['pseudo']))
19     $ok=false;
20 if (!preg_match('#^0[1-6][0-9]{8}$#',$_POST['tel']))
21     $ok=false;
22 if (!preg_match('#^[a-zA-Z]{3,}@[a-zA-Z]{4,10}\.(fr|com|org)$#',$_POST['mail']))
23     $ok=false;
24 if (!preg_match('#^ *([Uu]n|[Dd]es)? *[Cc]hiens? *$#',$_POST['rep']))
25     $ok=false;
26 return $ok;
27 }
28 $envoi= false;
29 if(testParametres(array('pseudo','tel','mail','message','rep')) )
30 {
31     if(testP_er())
32     {
33         $mes = 'Pseudo : ' . $_POST['pseudo'] . "\nTel : " . $_POST['tel'] .
34         "\nE-mail : " . $_POST['mail']. "\nMessage : " . $_POST['message'];
35         $test = mail('lacroix@iutv.univ-paris13.fr', '[test] Contact',$mes);
36         if($test)
37             echo '<p> Un e-mail a été envoyé à l\'administrateur du site </p>';
38         $envoi=true;
39     }
40     echo '<p> probleme dans la saisie </p>';
41 }
42 if(!$envoi)
43 {
44     ?>
45     <form method="post" action="contact.php">
46     <p> Pseudo : <input type="text" name="pseudo" value="<?php echo $_POST['pseudo'];?>"/> </p>
47     <p> Téléphone : <input type="text" name="tel" value="<?php echo $_POST['tel'];?>"/> </p>
48     <p> E-mail : <input type="text" name="mail" value="<?php echo $_POST['mail'];?>"/> </p>
49     <p> Message : <textarea rows="5" cols="50" name="message"><?php echo $_POST['message'];?>
50     </textarea > </p>
51     <p>Question: Quel animal aboie? <input type="text" name="rep"
52     value="<?php echo $_POST['rep'];?>"/>
53     <input type="submit" value="Envoyer un message" /></p>
54     </form>
55     <?php
56     }
57     ?>
58 </body>
59 </html>

```

Exercice 11 : Questionnaire à choix multiples

Question 11.1 : Le but de cet exercice est de faire une première page Web contenant des énigmes ou questions avec plusieurs réponses possibles (une seule d'entre elles étant correcte). L'utilisateur doit alors répondre aux différentes questions et le site affiche alors le score obtenu par l'utilisateur.

Question 11.2 : Ajouter sur la page d'accueil un formulaire pour que l'utilisateur puisse saisir son nom. Cette donnée sera utilisée par la suite pour personnaliser l'affichage du site.

Question 11.3 : Stocker, à l'aide d'un cookie le meilleur score obtenu par l'utilisateur. Dans la page du score, afficher ce meilleur score en plus du score obtenu à l'instant par l'utilisateur.

Correction :

Fichier `entete.html` :

```

1 <?php session_start(); ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
5 <head>
6 <title> Site d'énigmes</title>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8 <link href="style1.css" rel="stylesheet" type="text/css" />
9 </head>
10 <body>
11     <div id="entete">
12         <h1> <span> Site d'énigmes </span> </h1>
13     </div>

```

Fichier menu.html :

```

1 <div id="menu">
2     <ul>
3         <li> <a href="index.php"> Accueil </a> </li>
4         <li> <a href="questionnaire.php"> Questionnaire </a> </li>
5         <li> <a href="scores.php"> Score </a> </li>
6     </ul>
7 </div>
8 <div id="principal">

```

Fichier piedDePage.html :

```

1 </div>
2 <div id="piedDePage">
3 </div>
4 </body>
5 </html>

```

Fichier index.html :

```

1 <?php
2     include('entete.html');
3     include('menu.html');
4 ?>
5 <!-- On est dans l'id principal. L'entete et le menu ont déjà été ajoutés -->
6 <h1> <span> Bienvenue sur le site d'énigmes </span> </h1>
7
8 <p class = "accueil">
9 Ce site permet de tester son aptitude à la résolution d'énigmes.
10 Il a été élaboré pour le cours d'Introduction Web et correspond
11 au projet de fin de semestre à mettre en place. Son élaboration
12 reprend toutes les connaissances XHTML, CSS et PHP vues lors de ce semestre.
13 </p>
14
15 <?php
16 //Si l'utilisateur a soumis son nom
17 if( isset($_POST['nom']) and trim($_POST['nom'])!='' )
18     $_SESSION['nom'] = $_POST['nom'];
19
20 if( !isset($_SESSION['nom']) or trim($_SESSION['nom'])=='' )
21
22     echo '<h1> Veuillez entrer votre prénom </h1>
23         <form action="index.php" method="post">
24             <p> <input name="nom" type="text" /> <input type="submit"/></p>
25         </form>';

```

```

26     else
27         echo '<p> Bienvenue '. $_SESSION['nom'] . '</p>';
28     ?>
29 <?php include('piedDePage.html'); ?>

```

Fichier questionnaire.php :

```

1 <?php
2     include('entete.html');
3     include('menu.html');
4 ?>
5 <!-- On est dans l'id principal. L'entete et le menu ont déjà été ajoutés -->
6 <p> <?php echo $_SESSION['nom']; ?>, Veuillez remplir le questionnaire </p>
7
8 <form method="post" action="scores.php">
9 <p> Question 1 : que vaut 1+2?</p>
10 <ul>
11 <li> <input type="radio" name="enigme1" value="vrai" /> 3 </li>
12 <li> <input type="radio" name="enigme1" value="faux" /> 4 </li>
13 </ul>
14
15 <p> Question 2 : que vaut 4+5?</p>
16
17 <ul>
18 <li> <input type="radio" name="enigme2" value="vrai" /> 9 </li>
19 <li> <input type="radio" name="enigme2" value="faux" /> -1 </li>
20
21 </ul>
22
23 <p> Question 3 : que vaut 11 <sup>2 </sup> + 12<sup> 2 </sup>?</p>
24 <ul>
25 <li> <input type="radio" name="enigme3" value="faux" /> 23 </li>
26 <li> <input type="radio" name="enigme3" value="vrai" /> 265 </li>
27 <li> <input type="radio" name="enigme3" value="faux" /> 255 </li>
28 </ul>
29
30 <p> <input type="submit" name="valider" value="Calcul du résultat" /> </p>
31 </form>
32 <?php include('piedDePage.html'); ?>

```

Fichier fonctions.php :

```

1 <?php
2 //Tableau associant à chaque name d'enigme un nom
3 //qui sera utilisé par dans le tableau de score
4 $nomEnigme = array(
5     'enigme1' => 'Somme de deux nombres',
6     'enigme2' => 'Somme de deux nombres (suite)',
7     'enigme3' => 'Somme de carrés'
8 );
9
10 //Fonction calculant le score obtenu
11 function calculScore($t)
12 {
13     $nb = 0;
14     foreach($t as $v)
15     {
16         if ($v=='vrai')
17             $nb = $nb + 1;
18     }

```

```

19     return $nb;
20 }
21
22 //Fonction affichant le tableau
23 function afficheTableau($nomEnigme)
24 {
25
26     echo '<table>
27         <tr class="impair">
28             <th> Question numéro </th>
29             <th> Résultat </th>
30         </tr>';
31     $i=0;
32     foreach($nomEnigme as $c=>$v)
33     {
34         if($i%2==0)
35             echo '<tr><td>'. $v .'</td><td>';
36         else
37             echo '<tr class="impair"><td>'. $v .'</td><td>';
38         if(isset($_POST[$c]))
39             if($_POST[$c]=='vrai')
40                 echo ' </td></tr>';
41             else
42                 echo ' </td></tr>';
43         else
44             echo ' </td></tr>';
45         $i++;
46     }
47     echo '</table>';
48 }
49 ?>

```

Fichier scores.php :

```

1 <?php
2 include_once('fonctions.php');
3 if(isset($_POST['valider']))
4     $newScore = calculScore($_POST);
5 else
6     $newScore = -1;
7 if( isset($_COOKIE['meilleurScore']) )
8     $meilleurScore=$_COOKIE['meilleurScore'];
9 else
10    $meilleurScore = -1;
11 if($newScore > $meilleurScore)
12 {
13     setcookie('meilleurScore',$newScore, time() + 7 * 24 * 60 * 60, null, null, false, true);
14     $_COOKIE['meilleurScore'] = $newScore;
15 }
16 ?>
17 <?php
18     include('entete.html');
19     include('menu.html');
20 ?>
21 <!-- On est dans l'id principal. L'entete et le menu ont déjà été ajoutés -->
22 <h1> <span> Score obtenu </span> </h1>
23 <p> <?php echo $_SESSION['nom']; ?>, voici le score que vous avez obtenu
24     pour les différentes questions : </p>
25 <div class="tableauEtRes">

```

```

26 <?php
27     if (isset($_POST['valider']))
28     {
29         afficheTableau ($nomEnigme);
30         echo '<p class="valeurScore"> Score : ' . calculScore($_POST) . ' / '.
31         count($nomEnigme). '</p>';
32     }
33     else
34         echo '<p> Il faut répondre aux questions pour avoir un score! </p>';
35 ?>
36 <p> Meilleur score obtenu à ce jour <?php echo $_COOKIE['meilleurScore']; ?> </p>
37 </div>
38 <?php include('piedDePage.html'); ?>

```

Pour les étudiants en avance, rajouter le code permettant de dire combien de fois la personne a effectué le questionnaire (cookie contenant le nombre de fois que le joueur a répondu au questionnaire). ◇

Exercice 12 : Nombre de secondes écoulées depuis le premier janvier 2012

Écrire un script PHP permettant à un utilisateur de saisir une date de l'année 2012 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.

Correction :

Fichier nbS :

```

1 <?php require_once('codePHP.php'); ?>
2
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
4 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
6 <head>
7     <title>Ceci est une page de test avec des balises PHP</title>
8     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
9 </head>
10 <body>
11     <h1> Nombre de secondes passées depuis le premier janvier </h1>
12
13     <!-- Paragraphe donnant le nombre de secondes écoulées ou un message
14     d'erreur si les paramètres n'ont pas été saisis correctement -->
15     <p> <?php echo $message; ?> </p>
16
17     <h2> Veuillez saisir une date : </h2>
18 <form method="get" action="nbS.php">
19 <p> Date : <input type="text" name="jour" size="2" value="<?php echo $_GET['jour']; ?>" /> /
20         <input type="text" name="mois" size="2" value="<?php echo $_GET['mois']; ?>" /> /
21         2011
22 </p>
23 <p> Heure : <input type="text" name="heure" size="2" value="<?php echo $_GET['heure']; ?>" /> h
24         <input type="text" name="min" size="2" value="<?php echo $_GET['min']; ?>" /> min
25         <input type="text" name="sec" size="2" value="<?php echo $_GET['sec']; ?>" /> sec
26 </p>
27 <p> <input type="submit" value="Nombre de secondes écoulées depuis le 01/01/11 à 00h00:00" />
28 </p>
29 </form>
30 </body>
31 </html>

```


Fichier codePHP :

```

1 <?php
2 //teste que le tableau $_GET contient les clés contenues
3 //dans le tableau $tab passé en paramètre et que les
4 //valeurs associées à ces clés ne sont pas composées
5 //uniquement d'espaces.
6 fonction testParametres($tab)
7 {
8     foreach($tab as $v)
9         if( !isset($_GET[$v]) or trim($_GET[$v])==' ' )
10             return false;
11     return true;
12 }
13
14 //Vérifie que :
15 // - le mois est compris entre 1 et 12,
16 // - le jour entre 1 et le nombre de jours du mois,
17 // - l'heure comprise entre 1 et 23,
18 // - les minutes entre 1 et 59,
19 // - les secondes entre 1 et 59.
20 fonction verificationParametre ()
21 {
22     $nbJoursMois = array(31,28,31,30,31,30,31,31,30,31,30,31);
23     // Code retour :
24     // - 0 : ok
25     // - 1 : erreur date
26     // - 2 : erreur heure
27     if (($_GET['mois'] <= 0) or ($_GET['mois'] > 12))
28         return 1;
29     if (($_GET['jour'] <= 0) or ($_GET['jour'] > $nbJoursMois[$_GET['mois']-1]))
30         return 1;
31     if (($_GET['heure'] < 0) or ($_GET['heure'] > 23) or ($_GET['min'] < 0) or
32         ($_GET['min'] > 59) or ($_GET['sec'] < 0) or ($_GET['sec'] > 59))
33         return 2;
34     return 0;
35 }
36
37 // $message contient le message d'erreur s'il y en a une ou la phrase :
38 //Le nombre de secondes écoulées depuis le ... est égal à ... secondes
39 //sinon
40 $message = '';
41 // $nbSecondes contient le nombre de secondes écoulées
42 $nbSecondes = -1;
43
44 if(testParametres(array('jour','mois','heure','min','sec')) )
45 {
46     $ok = verificationParametre ();
47     if ($ok==0)
48     {
49         //Calcul du nombre de secondes écoulées
50         $nbJoursMois = array(31,28,31,30,31,30,31,31,30,31,30,31);
51         $nbJ = 0;
52         for ($i = 0; $i < $_GET['mois'] - 1 ; $i++)
53             $nbJ += $nbJoursMois[$i];
54
55         $nbSecondes = $_GET['sec'] + 60 * $_GET['min'] +
56         3600 * $_GET['heure'] + 3600 * 24 * ($_GET['jour'] + $nbJ-1);
57         //Variable $date la date (forme jj/mm/2011 à HHhMM:ss)

```

```

58         $message = 'Le nombre de secondes écoulées depuis le ' .
59         $_GET['jour']. '/' . $_GET['mois'] . '/2011 à ' . $_GET['heure'] .
60         'h' . $_GET['min'] . ':' . $_GET['sec'] . ' est égal à ' .
61         $nbSecondes . ' secondes.';
62     }
63     elseif ($ok==1)
64         $message = 'Problème dans la date! Recommencez la saisie.';
65     else
66         $message = 'Problème dans l\'heure! Recommencez la saisie.';
67     }
68     else
69         $message = 'Paramètres non saisis! Recommencez la saisie.';
70     ?>

```

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 dans la variable `$_SESSION`.

Correction :

```

1  <?php include('entete.html') ?>
2  <h1> Bienvenue sur le site du jeu devineChiffre </h1>
3  <?php
4  if( !isset($_POST['jeu']) )
5  {
6      $_SESSION['nbM'] = mt_rand(0,100);
7      echo '<form method="post" action="devineChiffre.php">
8      <p> <input type="submit" value="Commencer le jeu" />
9      <input type="hidden" name="jeu" value="7" />
10     </p>
11     </form>';
12     echo '<p> Nombre magique : ' . $_SESSION['nbM'] . '</p>';
13 }
14 else
15 {
16     $gagne = false;
17     //L'utilisateur a saisi un nombre
18     if( isset($_POST['nb']) )
19     {
20         $nb = (int) $_POST['nb'];
21         //var_dump($nb);
22         if($nb==$_SESSION['nbM'])
23         {
24             echo '<p> Vous avez gagné! <a href="devineChiffre"> Rejouer </a></p>';
25             $gagne = true;
26         }
27         elseif ($nb <$_SESSION['nbM'])

```

```

28         echo '<p> Le nombre à trouver est plus grand </p>';
29     else
30         echo '<p> Le nombre à trouver est plus petit </p>';
31 }
32 if(!$gagne)
33 {
34     if( $_POST['jeu']!=0)
35     {
36         echo '<p> Il vous reste ' . $_POST['jeu'] . ' tentatives </p>';
37         echo '<form method="post" action="devineChiffre.php">
38         <p> Nombre : <input type="text" name="nb" />
39         <input type="submit" value="Tenter" />
40         <input type="hidden" name="jeu" value="' . ($_POST['jeu'] - 1) . '" />
41         </p>
42         </form>';
43     }
44     else
45         echo '<p> Vous avez perdu! <a href="devineChiffre"> Rejouer </a> </p>';
46 }
47 }
48 ?>
49 </body>
50 </html>

```

Le fichier `entete.html` contient l'instruction `<?php session_start(); ?>` ainsi que l'en-tête d'un document XHTML.



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

Correction :

Non, car chaque coup joué correspond à une requête. On a donc une surcharge inutile du serveur et un encombrement du réseau. De plus, le jeu est plus lent. Ce jeu devrait être programmé avec un langage orienté client plutôt que serveur (par exemple : javascript).

